

# FileMaker® Server 16

Guida alla Pubblicazione Web personalizzata



**FileMaker®**  
An Apple Subsidiary

© 2004-2017 FileMaker, Inc. Tutti i diritti riservati.

FileMaker, Inc.

5201 Patrick Henry Drive

Santa Clara, California 95054 Stati Uniti

FileMaker, FileMaker Go e il logo della cartella sono marchi di FileMaker, Inc. registrati negli Stati Uniti e in altri Paesi. FileMaker WebDirect e FileMaker Cloud sono marchi di FileMaker, Inc. Tutti gli altri marchi sono di proprietà dei rispettivi proprietari.

La documentazione di FileMaker è protetta da copyright. Non è permesso fare copie o distribuire questa documentazione senza previa autorizzazione scritta di FileMaker. È possibile utilizzare la presente documentazione soltanto unitamente a una copia del software FileMaker concessa in licenza.

Tutte le persone, le società, gli indirizzi e-mail e gli URL elencati negli esempi sono fittizi e ogni riferimento a persone, società, indirizzi e-mail o URL esistenti è puramente casuale. Gli autori sono elencati nel documento Riconoscimenti fornito insieme a questo software. I prodotti di terze parti e gli URL sono citati unicamente a scopo informativo e non costituiscono obbligo o raccomandazione. FileMaker, Inc. non si assume alcuna responsabilità nei confronti delle prestazioni di questi prodotti.

Per ulteriori informazioni, visitare il nostro sito Web all'indirizzo <http://www.filemaker.com/it>.

Edizione: 01

# Sommario

<b><i>Prefazione</i></b>	8
Informazioni su questa guida	8
Dove reperire la documentazione di FileMaker	8
<b>Capitolo 1</b>	
<b><i>Introduzione alla Pubblicazione Web personalizzata</i></b>	9
Motore per la Pubblicazione Web	10
Modalità di elaborazione di una richiesta del Motore per la Pubblicazione Web	10
Pubblicazione Web personalizzata con XML	11
Pubblicazione Web personalizzata con PHP	11
Confronto tra XML e PHP	11
Motivi per scegliere XML	11
Motivi per scegliere PHP	12
<b>Capitolo 2</b>	
<b><i>Preparazione dei database per la Pubblicazione Web personalizzata</i></b>	13
Attivazione della Pubblicazione Web personalizzata in un database	13
Accesso ad un database protetto	13
Protezione dei database pubblicati	14
Supporto del server Web per i tipi di supporto Internet (MIME)	15
Pubblicazione del contenuto dei campi Contenitore sul Web	15
Oggetti dei campi Contenitore incorporati in un database	15
Campi Contenitore con riferimenti ai file memorizzati	16
Campi Contenitore con dati memorizzati esternamente	16
Campi Contenitore e download progressivo	17
Modalità di visualizzazione dei dati dei campi Contenitore	17
Script FileMaker e Pubblicazione Web personalizzata	18
Suggerimenti e considerazioni sugli script	18
Comportamento degli script nelle soluzioni di Pubblicazione Web personalizzata	19
Trigger di script e soluzioni di Pubblicazione Web personalizzata	19
<b>Capitolo 3</b>	
<b><i>Pubblicazione Web personalizzata con XML</i></b>	20
Creazione di siti Web dinamici con il Motore per la Pubblicazione Web	20
Caratteristiche principali della Pubblicazione Web personalizzata con XML	21
Requisiti per la pubblicazione sul Web	21
Requisiti per la pubblicazione di un database mediante Pubblicazione Web personalizzata	21
Cosa si richiede agli utenti Web per accedere alla soluzione di Pubblicazione Web personalizzata	22
Connessione a Internet o ad una rete intranet	22
Operazioni successive	22

## Capitolo 4

<b>Accesso a dati XML con il Motore per la Pubblicazione Web</b>	23
Uso di Pubblicazione Web personalizzata con il linguaggio XML	23
Differenze di importazione/esportazione XML tra Motore per la Pubblicazione Web e FileMaker Pro	23
Modalità di generazione di dati XML da una richiesta da parte del Motore per la Pubblicazione Web	24
Procedura generale per accedere a dati XML dal Motore per la Pubblicazione Web	25
Sintassi URL per dati XML e oggetti Contenitore	25
Sintassi URL per dati XML	25
Sintassi URL per oggetti Contenitore FileMaker in soluzioni XML	26
Codifica testo URL	27
Accesso a dati XML attraverso il Motore per la Pubblicazione Web	28
Nome spazio per XML FileMaker	28
Codici di errore dei database FileMaker	29
Recupero di DTD per le grammatiche FileMaker	29
Utilizzo della grammatica fmresultset	29
Descrizione degli elementi della grammatica fmresultset	30
Dati XML con la grammatica fmresultset	32
Utilizzo di altre grammatiche XML FileMaker	33
Descrizione degli elementi della grammatica FMPXMLRESULT	33
Dati XML con la grammatica FMPXMLRESULT	34
Descrizione degli elementi della grammatica FMPXMLLAYOUT	35
Dati XML con FMPXMLLAYOUT grammatica	37
Dati codificati nel formato UTF-8	38
Utilizzo di stringhe di query FileMaker per richiedere dati XML	38
Commutazione dei formati per una risposta XML	40
Comprendere come una richiesta XML viene elaborata	41
Risoluzione dei problemi di accesso a documenti XML	41

## Capitolo 5

<b>Nomi validi utilizzati nelle stringhe di query XML</b>	42
Comandi e parametri query	42
Linee guida per l'utilizzo dei comandi e dei parametri query	43
Analisi dei comandi query	44
Sintassi di un nome di campo completamente qualificato	45
Utilizzo dei comandi query con campi portale	45
Sintassi per specificare un campo globale	47

Riferimento comando query	47
Comando query –dbnames (Nomi database)	47
Comando query –delete (Elimina record)	47
Comando query –dup (Duplica record)	48
Comando query –edit (Modifica record)	48
Comandi query –find, –findall o –findany (Trova record)	49
Comando query –findquery (Ricerca composta)	49
Comando query –layoutnames (Nomi formati)	50
Comando query –new (Nuovo record)	50
Comando query –scriptnames (Nomi script)	51
Comando query –view (Visualizza le informazioni sul formato)	51
Riferimento parametro query	51
Parametro query –db (Nome database)	51
Parametro query –delete.related (Elimina record portale)	52
Parametro query –field (Nome campo Contenitore)	52
Parametro query fieldname (Nome campo non Contenitore)	52
Parametro query fieldname.op (Operatore di confronto)	53
Parametro query –lay (Formato)	54
Parametro query –lay.response (Applicazione del formato per la risposta)	54
Parametro query –lop (Operatore logico)	55
Parametro query –max (Record max)	55
Parametro query –modid (ID modifica)	56
Parametro query –query (Richiesta di ricerca composta)	56
Parametro query –recid (ID record)	57
Parametro query –relatedsets.filter (Filtra record portale)	58
Parametro query –relatedsets.max (Limita record portale)	59
Parametro query –script (Script)	59
Parametro query –script.param (Trasferisci parametro a script)	60
Parametro query –script.pfind (Script prima di trovare)	60
Parametro query –script.pfind.param (Trasferisci parametro a script prima di trovare)	60
Parametro query –script.presort (Script prima di ordinare)	61
Parametro query –script.presort.param (Trasferisci parametro a script prima di ordinare)	61
Parametro query –skip (Salta record)	62
Parametro query –sortfield (Campo ordinamento)	62
Parametro query –sortorder (Ordinamento)	63

## Capitolo 6

### *Pubblicazione Web personalizzata con PHP*

<i>Pubblicazione Web personalizzata con PHP</i>	64
Caratteristiche principali della Pubblicazione Web personalizzata con PHP	64
Requisiti della Pubblicazione Web personalizzata	64
Requisiti per la pubblicazione di un database mediante Pubblicazione Web personalizzata	64
Cosa è richiesto agli utenti Web per accedere a una soluzione di Pubblicazione Web personalizzata	65
Connessione a Internet o a una rete intranet	65
Installazione manuale di FileMaker API per PHP	66
Operazioni successive	67

## Capitolo 7

<b><i>Panoramica della Pubblicazione Web personalizzata con PHP</i></b>	68
Utilizzo del Motore per la Pubblicazione Web con soluzioni PHP	68
Operazioni generali per la Pubblicazione Web personalizzata con PHP	68

## Capitolo 8

<b><i>Utilizzo di FileMaker API per PHP</i></b>	70
Dove ottenere informazioni aggiuntive	70
Informazioni di riferimento a FileMaker API per PHP	70
Supporto FileMaker API per PHP	70
Utilizzo della classe FileMaker	71
Oggetti classe FileMaker	71
Oggetti comando FileMaker	71
Decodifica dei dati per l'uso in FileMaker API	72
Collegamento a un database FileMaker	72
Utilizzo dei record	73
Creazione di un record	73
Duplicazione di un record	73
Modifica di un record	74
Eliminazione di un record	74
Esecuzione di script FileMaker	75
Recupero dell'elenco degli script disponibili	75
Esecuzione di uno script FileMaker	75
Esecuzione di uno script prima di eseguire un comando	75
Esecuzione di uno script prima di ordinare un gruppo di risultati	76
Esecuzione di uno script dopo la creazione di un gruppo di risultati	76
Ordine di esecuzione degli script	76
Utilizzo dei formati FileMaker	77
Utilizzo dei portali	78
Elenco dei portali definiti in un formato specifico	78
Recupero dei nomi dei portali per un oggetto risultato specifico	78
Recupero delle informazioni sui portali per un formato specifico	78
Recupero delle informazioni per un portale specifico	78
Recupero del nome della tabella per un portale	79
Recupero dei record del portale per un record specifico	79
Creazione di un nuovo record in un portale	79
Eliminazione di un record da un portale	79
Utilizzo delle liste valori	80
Recupero dei nomi di tutte le liste valori per un formato specifico	80
Recupero di una matrice di tutte le liste valori per un formato specifico	80
Recupero dei valori per una lista valori indicata	80
Esecuzione di richieste di ricerca	81
Utilizzo del comando Find All	82
Utilizzo del comando Find Any	82
Utilizzo del comando Find	82
Utilizzo di un comando Compound Find	83
Elaborazione dei record in un gruppo di risultati	85
Limitazione delle righe del portale restituite dalle richieste di ricerca	86

Pre-convalida di comandi, record e campi	86
Pre-convalida dei record in un comando	88
Pre-convalida dei record	88
Pre-convalida dei campi	88
Elaborazione degli errori di convalida	88
Gestione degli errori	90
<b>Capitolo 9</b>	
<b><i>Preparazione, test e monitoraggio di un sito</i></b>	91
Preparazione di un sito Pubblicazione Web personalizzata	91
Test di un sito Pubblicazione Web personalizzata	92
Fogli di stile per testare l'output XML	93
Monitoraggio di un sito	94
Us	94
Utilizzo del registro del Motore per la Pubblicazione Web	94
Utilizzo del registro errori del Modulo Server Web	96
Utilizzo dei registri Tomcat	97
<b>Appendice A</b>	
<b><i>Codici di errore per la Pubblicazione Web personalizzata</i></b>	98
Codici di errore in formato XML	98
Codici di errore per database FileMaker	99
<b>Indice</b>	100

# Prefazione

## Informazioni su questa guida

Questa guida suppone che il lettore abbia esperienza nell'utilizzo di FileMaker® Pro per la creazione di database. È necessario comprendere le nozioni di base sulla progettazione dei database di FileMaker Pro e i concetti di campo, relazione, formato, portale e contenitore. Per informazioni su FileMaker Pro, consultare la [Guida di FileMaker Pro](#).

Questa guida suppone inoltre che il lettore abbia esperienza nello sviluppo di siti Web, in particolare nell'utilizzo di tecnologie come XML o PHP per integrare i dati FileMaker con siti Web e applicazioni Web.

Questa guida fornisce le seguenti informazioni sulla Pubblicazione Web personalizzata con FileMaker Server:

- cosa è richiesto per sviluppare una soluzione di Pubblicazione Web personalizzata
- come pubblicare i database con XML
- come ottenere dati XML da database ospitati da FileMaker Server
- come pubblicare i database utilizzando PHP
- come utilizzare FileMaker API per PHP per ottenere dati da database ospitati da FileMaker Server
- cosa è richiesto agli utenti Web per accedere a Pubblicazione Web personalizzata

## Dove reperire la documentazione di FileMaker

- Nella FileMaker Server Admin Console, selezionare il menu **Guida > Documentazione sul prodotto FileMaker Server**.
- Fare clic sui link nella pagina iniziale della FileMaker Server Admin Console.
- Per avere informazioni, visualizzare o scaricare altra documentazione di FileMaker, visitare il sito <http://www.filemaker.com/documentation/it>.

La Guida in linea è accessibile dalla FileMaker Server Admin Console. Selezionare il menu **Guida > Guida di FileMaker Server**.



# Capitolo 1

## Introduzione alla Pubblicazione Web personalizzata

Con FileMaker Server, è possibile pubblicare i database FileMaker su Internet o su una rete intranet nei seguenti modi.

**FileMaker WebDirect:** Con FileMaker WebDirect, è possibile pubblicare rapidamente e facilmente i formati da un database sul Web. Non è necessario installare un software aggiuntivo; chiunque disponga di un browser Web compatibile e abbia accesso a Internet o a una rete intranet può connettersi alla soluzione FileMaker WebDirect per visualizzare, modificare, ordinare o eseguire ricerche all'interno dei record, a condizione che disponga dei privilegi di accesso necessari.

Con FileMaker WebDirect, è necessario che sul computer host sia in esecuzione FileMaker Server. L'interfaccia utente è simile all'applicazione desktop FileMaker Pro. Le pagine Web e i moduli con cui l'utente Web interagisce dipendono dai formati e dalle visualizzazioni definiti nel database FileMaker Pro. Vedere la [Guida di FileMaker WebDirect](#).

**Pubblicazione statica:** Se i dati vengono modificati di rado oppure se non si desidera che gli utenti si connettano dinamicamente al database, è possibile ricorrere alla pubblicazione statica. La pubblicazione statica consente di esportare i dati di un database FileMaker Pro per creare una pagina Web personalizzabile con HTML. La pagina Web non riflette le modifiche apportate alle informazioni nel database e gli utenti non si connettono al database. (con FileMaker WebDirect, i dati vengono aggiornati nel browser Web ogni volta che vengono aggiornati sul database). Vedere la [Guida di FileMaker Pro](#).

**FileMaker Data API:** Se si ha esperienza con l'utilizzo dell'architettura REST (REpresentational State Transfer), FileMaker fornisce un'implementazione API REST che consente ai servizi Web di accedere ai dati nelle soluzioni ospitate. Il servizio Web chiama FileMaker Data API per ottenere un token di autenticazione per accedere a una soluzione ospitata, quindi utilizza questo token nelle chiamate successive per creare, aggiornare ed eliminare record e per eseguire richieste di ricerca. FileMaker Data API restituisce i dati in formato JSON (JavaScript Object Notation). Vedere la [Guida di FileMaker Data API](#).

**Pubblicazione Web personalizzata:** Per integrare il database FileMaker con un sito Web personalizzato, utilizzare le tecnologie di Pubblicazione Web personalizzata disponibili con FileMaker Server. FileMaker Server, che ospita i database pubblicati, non richiede che FileMaker Pro sia installato o sia in esecuzione per rendere disponibile la Pubblicazione Web personalizzata.

Con la Pubblicazione Web personalizzata è possibile:

- Integrare un database in un sito Web
- Determinare il modo in cui gli utenti interagiscono con i dati
- Controllare il modo in cui i dati vengono visualizzati nel browser Web

FileMaker Server provvede due tecnologie di Pubblicazione Web personalizzata:

- Pubblicazione Web personalizzata con XML: Utilizzare la pubblicazione di dati XML per scambiare dati FileMaker con altri siti Web e applicazioni. Utilizzando richieste HTTP URL con comandi e parametri query specifici FileMaker, è possibile eseguire query su un database ospitato da FileMaker Server, scaricare i dati risultanti in formato XML e utilizzare i dati XML risultanti nel modo desiderato.

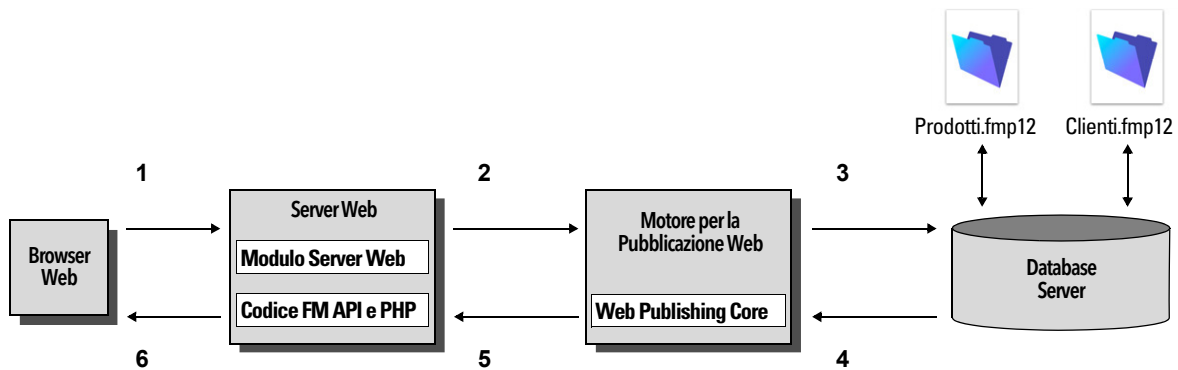
- Pubblicazione Web personalizzata con PHP: Utilizzare FileMaker API per PHP, che fornisce un'interfaccia PHP con il database FileMaker orientata agli oggetti, per integrare i dati FileMaker in un'applicazione Web PHP. Eseguendo direttamente la codifica delle pagine Web PHP, si ha il controllo completo sull'interfaccia utente e sul modo in cui l'utente interagisce con i dati.

## Motore per la Pubblicazione Web

Per supportare FileMaker WebDirect e Pubblicazione Web personalizzata, FileMaker Server utilizza un insieme di componenti software definiti *Motore per la Pubblicazione Web di FileMaker Server*. Il Motore per la Pubblicazione Web gestisce le interazioni tra il browser di un utente Web, il server Web e FileMaker Server.

**Pubblicazione Web personalizzata con XML:** Gli utenti Web accedono alla soluzione di Pubblicazione Web personalizzata facendo clic su un collegamento HREF o inserendo un URL (Uniform Resource Locator) che indica l'indirizzo del server Web e una richiesta di stringa di query FileMaker. Il Motore per la Pubblicazione Web restituisce i dati XML specificati nella richiesta di stringa di query.

**Pubblicazione Web personalizzata con PHP:** Quando un utente Web accede alla soluzione Pubblicazione Web personalizzata, PHP su FileMaker Server si collega con il Motore per la Pubblicazione Web e risponde attraverso FileMaker API per PHP.



Uso del Motore per la Pubblicazione Web di FileMaker Server per la Pubblicazione Web personalizzata

### Modalità di elaborazione di una richiesta del Motore per la Pubblicazione Web

1. Una richiesta viene inviata da un'applicazione o da un browser Web al server Web.
2. Il server Web invia la richiesta attraverso il Modulo Server Web di FileMaker al Motore per la Pubblicazione Web.
3. Il Motore per la Pubblicazione Web richiede i dati al database ospitato dal server Database.
4. FileMaker Server invia i dati FileMaker richiesti al Motore per la Pubblicazione Web.
5. Il Motore per la Pubblicazione Web converte i dati FileMaker per rispondere alla richiesta.
  - Per le richieste PHP, FileMaker API per PHP converte la richiesta PHP in una richiesta XML. Il Motore per la Pubblicazione Web elabora la richiesta XML e reinvia i dati XML a FileMaker API per PHP. Quindi FileMaker API per PHP converte i dati XML in oggetti PHP che possono essere utilizzati dall'applicazione PHP.
  - Per le richieste XML, il Motore per la Pubblicazione Web invia i dati XML direttamente al server Web.

6. Il server Web invia l'output al browser Web o al programma che lo richiedono.

**Importante** Quando si pubblicano dati sul Web si deve prestare particolare attenzione alla sicurezza. Consultare le linee guida per la sicurezza contenute nella [Guida per la sicurezza di FileMaker](#).

Per informazioni sulla preparazione di un database per la Pubblicazione Web personalizzata, vedere capitolo 2, "Preparazione dei database per la Pubblicazione Web personalizzata."

## Pubblicazione Web personalizzata con XML

La Pubblicazione Web personalizzata con XML di FileMaker consente di inviare richieste query a un database FileMaker Pro ospitato da FileMaker Server e di visualizzare, modificare o gestire i dati risultanti. Utilizzando una richiesta HTTP con i parametri e i comandi query appropriati, è possibile recuperare i dati FileMaker come documento XML. È possibile esportare i dati XML in altre applicazioni.

## Pubblicazione Web personalizzata con PHP

FileMaker API per PHP fornisce un'interfaccia PHP con il database FileMaker orientata agli oggetti. FileMaker API per PHP consente di accedere ai dati e alla logica memorizzati in un database FileMaker Pro e di pubblicarli sul web o di esportarli in altre applicazioni. API supporta anche comandi di ricerca complessi e composti per estrarre e filtrare i dati memorizzati nei database FileMaker Pro.

Progettato inizialmente come linguaggio di programmazione procedurale, PHP è stato migliorato come linguaggio di sviluppo web orientato agli oggetti. PHP fornisce le funzioni del linguaggio di programmazione per costruire virtualmente ogni tipo di logica in una pagina web. Ad esempio è possibile utilizzare costrutti di logica condizionale per controllare la creazione di pagine, il percorso dei dati o il flusso di lavoro. PHP fornisce anche l'amministrazione e la sicurezza del sito.

## Confronto tra XML e PHP

Le seguenti sezioni forniscono alcune direttive per stabilire la soluzione più adeguata per un determinato sito.

### Motivi per scegliere XML

- La sintassi dei parametri delle richieste XML di FileMaker è progettata per interagire con i database, semplificando lo sviluppo della soluzione.
- XML è uno standard W3C.
- XML è un formato leggibile dall'uomo e dalla macchina che supporta Unicode, consentendo la comunicazione dei dati con ogni tipo di linguaggio scritto.
- XML è particolarmente adatto per la presentazione di record, elenchi e dati con struttura ad albero.
- È possibile utilizzare FMPXMLRESULT per l'accesso ai dati XML utilizzando la Pubblicazione Web personalizzata e per l'esportazione XML dai database FileMaker Pro.

**Nota** Per informazioni sulla Pubblicazione Web personalizzata con XML, vedere capitolo 3, "Pubblicazione Web personalizzata con XML."

## Motivi per scegliere PHP

- PHP è un linguaggio di scripting procedurale più efficace e orientato agli oggetti e, nello stesso tempo, relativamente facile da apprendere. Offre molte risorse per l'apprendimento, lo sviluppo e il supporto.
- FileMaker API per PHP consente di accedere ai dati e alla logica memorizzati in un database FileMaker Pro e di pubblicarli sul web o di esportarli in altre applicazioni.
- PHP consente di utilizzare una logica condizionale per controllare la creazione o il flusso delle pagine.
- PHP fornisce le funzioni del linguaggio di programmazione per costruire vari tipi di logica in una pagina web.
- PHP è uno dei linguaggi di scripting Web più conosciuti.
- PHP è un linguaggio open source, disponibile sul sito [php.net](http://php.net).
- PHP consente di accedere ad una grande quantità di componenti di terze parti che possono essere integrati nelle proprie soluzioni.

**Nota** Per informazioni sulla Pubblicazione Web personalizzata con PHP, vedere capitolo 6, “Pubblicazione Web personalizzata con PHP.”

# Capitolo 2

## Preparazione dei database per la Pubblicazione Web personalizzata

Prima di utilizzare la Pubblicazione Web personalizzata con un database, è necessario preparare il database e proteggerlo da accessi non autorizzati.

### Attivazione della Pubblicazione Web personalizzata in un database

È necessario attivare un privilegio esteso per la Pubblicazione Web personalizzata in ogni database da pubblicare. Se non si attiva un privilegio esteso per la Pubblicazione Web personalizzata nel database, gli utenti Web non saranno in grado di utilizzare la Pubblicazione Web personalizzata per accedere al database, anche se questo è ospitato da FileMaker Server che è configurato per supportare un Motore per la Pubblicazione Web.

**Per attivare la Pubblicazione Web personalizzata per un database:**

1. In FileMaker Pro, aprire il database che si desidera pubblicare utilizzando un account che dispone del set di privilegi Accesso completo. In alternativa, si può aprire il database utilizzando un account che dispone dei privilegi di accesso Gestisci privilegi estesi.
2. Assegnare il privilegio esteso per la Pubblicazione Web personalizzata da utilizzare:
  - Per la Pubblicazione Web personalizzata con XML, utilizzare fmxml
  - Per la Pubblicazione Web personalizzata con PHP, utilizzare fmphp
3. Assegnare il(i) set di privilegi che comprendono il privilegio esteso di Pubblicazione Web personalizzata a uno o più account, o all'account Admin o Ospite.

**Nota** Quando si definiscono i nomi utente e le password per la Pubblicazione Web personalizzata, si possono usare solo caratteri ASCII stampabili, ad esempio **a-z**, **A-Z** e **0-9**. Per conferire maggiore sicurezza ai nomi account e password, inserire caratteri di punteggiatura come “!” e “%,” ma non i due punti. Per ulteriori informazioni sulla configurazione degli account, consultare la [Guida di FileMaker Pro](#).

### Accesso ad un database protetto

La Pubblicazione Web personalizzata consente di limitare l'accesso ai database pubblicati proteggendoli mediante password, crittografia e connessioni sicure. Quando si usa una soluzione Pubblicazione Web personalizzata per accedere ad un database, agli utenti Web possono essere richiesti i dati sull'account (nome utente e password). Se l'account Ospite per il database è disattivato o non ha un set di privilegi attivato che includa il privilegio esteso di Pubblicazione Web personalizzata, il Motore per la Pubblicazione Web usa l'autenticazione di base HTTP per richiedere l'autenticazione agli utenti Web. Il browser dell'utente Web visualizza la finestra di dialogo di autenticazione di base HTTP in cui l'utente può digitare il nome utente e la password per un account che dispone del privilegio esteso di Pubblicazione Web personalizzata.

I seguenti punti riassumono il processo che si verifica quando un utente Web utilizza una soluzione di Pubblicazione Web personalizzata per accedere ad un database:

- Se per un account non è stata assegnata nessuna password, gli utenti Web specificano soltanto il nome utente.
- Se l'account Ospite è disattivato, agli utenti verrà richiesto il nome utente e la password quando accedono al database. per l'account deve essere attivato il privilegio esteso di Pubblicazione Web personalizzata.
- Se l'account Ospite è attivato con il set di privilegi che comprende il privilegio esteso di Pubblicazione Web personalizzata, tutti gli utenti Web aprono automaticamente il database con i privilegi di accesso assegnati all'account Ospite. Se all'account Ospite è assegnato il privilegio esteso Pubblicazione Web personalizzata:
  - Agli utenti Web non viene richiesto il nome utente e la password quando aprono un file.
  - Tutti gli utenti Web accedono automaticamente con l'account Ospite e assumono i privilegi dell'account Ospite. È possibile permettere agli utenti di cambiare i propri account di accesso da un browser Web con l'istruzione di script Nuovo accesso (ad esempio, per passare da un account Ospite ad un account con più privilegi).
  - Il set di privilegi predefinito per gli account Ospite fornisce accesso di sola lettura. È possibile cambiare i privilegi predefiniti, compresi i privilegi estesi, per questo account. Vedere la [Guida di FileMaker Pro](#).
- Se un utente Web ha inserito informazioni valide sull'account, queste vengono riutilizzate fino allo scadere della sessione del browser. Quando la sessione del browser scade, all'utente Web viene nuovamente richiesto di inserire un account valido.

**Nota** Per impostazione predefinita, gli utenti Web non possono modificare la loro password da un browser Web. Questa funzione può essere inserita in un database con l'istruzione di script password Modifica password che permette agli utenti Web di modificare le password dal browser. Vedere la [Guida di FileMaker Pro](#).

## Protezione dei database pubblicati

Quando si utilizza la Pubblicazione Web personalizzata, è possibile definire chi può accedere ai database pubblicati.

- Assegnare delle password agli account del database che vengono usati per la Pubblicazione Web personalizzata.
- Attivare un privilegio esteso per la Pubblicazione Web personalizzata solo nei set di privilegi degli account a cui si desidera consentire l'accesso ai database pubblicati.
- Disattivare il privilegio esteso per la Pubblicazione Web personalizzata per un determinato database deselezionando il privilegio esteso fmxml o fmphp per tutti i set di privilegi in quel database. Vedere la [Guida di FileMaker Pro](#).
- Attivare o disattivare la Pubblicazione Web personalizzata per tutte le soluzioni di Pubblicazione Web personalizzata nel Motore per la Pubblicazione Web che utilizzano Admin Console di FileMaker Server. Vedere la [Guida all'installazione e alla configurazione di FileMaker Server](#) e la [Guida di FileMaker Server](#).
- Configurare il server Web per limitare gli indirizzi IP che possono accedere ai database attraverso il Motore per la Pubblicazione Web. Ad esempio, è possibile specificare che l'accesso ai database sia consentito esclusivamente agli utenti Web con l'indirizzo IP 192.168.100.101. Per informazioni sulla limitazione degli indirizzi IP, vedere la documentazione del server Web.

FileMaker Server supporta la crittografia per i dati scritti su disco e per i dati trasmessi ai client.

- Crittografare il database utilizzando la funzione Crittografia database di FileMaker Pro Advanced. La crittografia protegge il file di database FileMaker e i file temporanei scritti sul disco. Vedere la [Guida all'installazione e alla configurazione di FileMaker Server](#) e la [Guida di FileMaker Pro](#).
  - Un database crittografato ospitato su FileMaker Server viene aperto utilizzando l'Admin Console o l'interfaccia a riga di comando (CLI). L'amministratore di FileMaker Server apre il file utilizzando la sua password di crittografia del database, in modo da permettere ai client di FileMaker di utilizzare il database crittografato.
  - Una volta che il database di FileMaker crittografato è stato aperto con la password di crittografia dall'amministratore di FileMaker Server, i client di FileMaker non hanno bisogno di una password di crittografia per accedere al database crittografato. Per informazioni sull'apertura di un database crittografato, vedere la [Guida di FileMaker Server](#).
- Utilizzare la crittografia Secure Sockets Layer (SSL) per la comunicazione tra il server Web e i browser Web. Le connessioni SSL sono realizzate mediante una connessione HTTPS. FileMaker Server fornisce un certificato SSL standard firmato da FileMaker, Inc. che non verifica il nome del server. Il certificato predefinito di FileMaker è destinato solo ai test. Per l'uso in un ambiente di produzione è richiesto un certificato SSL personalizzato. Vedere la [Guida all'installazione e alla configurazione di FileMaker Server](#).

Se si seleziona **Usare HSTS per client Web** nell'Admin Console, utilizzare la directory HTTPS per ospitare i file del sito PHP. Vedere capitolo 7, "Operazioni generali per la Pubblicazione Web personalizzata con PHP".

Per informazioni sulla protezione del database, vedere la [Guida per la sicurezza di FileMaker](#).

**Nota** Per motivi di sicurezza, le pagine Web ospitate da altri server Web non possono utilizzare il tag <iframe> per incorporare i contenuti di Pubblicazione Web personalizzata. Per incorporare i contenuti di Pubblicazione Web personalizzata nei tag <iframe> di pagine Web separate, anche quelle pagine Web devono essere ospitate dal server Web FileMaker Server.

## Supporto del server Web per i tipi di supporto Internet (MIME)

Il server Web determina il supporto per i tipi MIME (Multipurpose Internet Mail Extensions) correnti registrati per Internet. Il Motore per la Pubblicazione Web non cambia il supporto del server Web per MIME. Vedere la documentazione del server Web.

## Pubblicazione del contenuto dei campi Contenitore sul Web

I contenuti di un campo Contenitore possono essere incorporati nel database, collegati come riferimento attraverso il relativo percorso o memorizzati esternamente.

### Oggetti dei campi Contenitore incorporati in un database

Se nei campi Contenitore vengono memorizzati i file effettivi del database FileMaker, non è necessario eseguire nessun'operazione con i campi Contenitore se il file di database è ospitato nella posizione appropriata ed è accessibile su FileMaker Server. Vedere "Sintassi URL per oggetti Contenitore FileMaker in soluzioni XML" a pagina 26.

## Campi Contenitore con riferimenti ai file memorizzati

Se un campo Contenitore memorizza un riferimento ai file, seguire questi passi per pubblicare i file di riferimento utilizzando il Motore per la Pubblicazione Web.

1. Memorizzare i file oggetto Contenitore nella cartella Web all'interno della cartella FileMaker Pro.
2. In FileMaker Pro, inserire gli oggetti nel campo Contenitore e selezionare l'opzione **Memorizza solo un riferimento al file**.
3. Copiare o spostare i file degli oggetti di riferimento nella cartella Web nella stessa posizione relativa nella cartella principale del software del server Web.
  - Per IIS (Windows):  
[unità]: \Programmi\FileMaker\FileMaker Server\HTTPServer\conf  
dove [unità] è l'unità in cui risiede il componente Motore per la Pubblicazione Web della distribuzione di FileMaker Server.
  - Per Apache (macOS): /Libreria/FileMaker Server/HTTPServer/htdocs

**Nota** Per gli oggetti Contenitore memorizzati come riferimenti ai file, il server Web deve essere configurato per supportare i tipi MIME per i file da gestire, come i filmati. Il server Web determina il supporto per i tipi MIME (Multipurpose Internet Mail Extensions) correnti registrati per Internet. Il Motore per la Pubblicazione Web non cambia il supporto del server Web per MIME. Vedere la documentazione del server Web.

## Campi Contenitore con dati memorizzati esternamente

Se un campo Contenitore memorizza oggetti esternamente, ovvero se si è selezionato **Memorizza dati Contenitore esternamente** nella finestra di dialogo Opzioni per il campo di FileMaker Pro, utilizzare FileMaker Pro per trasferire i file di database dal file system del client a FileMaker Server. Se si utilizza FileMaker Pro per caricare un database, i dati del campo Contenitore memorizzati esternamente vengono memorizzati in FileMaker Server come parte del processo. Per informazioni sul trasferimento dei file di database in FileMaker Server, consultare la [Guida di FileMaker Pro](#).

Se si carica manualmente un database che utilizza un campo Contenitore con oggetti memorizzati esternamente, è necessario seguire questi passi per pubblicare gli oggetti Contenitore memorizzati esternamente utilizzando il Motore per la Pubblicazione Web.

### Per caricare un database manualmente:

1. Mettere il file di database nella sede appropriata sul server. Posizionare i file di database FileMaker Pro, (i tasti di scelta rapida (Windows) o alias (macOS) riferiti a quei file) che si desidera aprire con FileMaker Server, nelle seguenti cartelle:
  - Windows:  
[unità]: \Programmi\FileMaker\FileMaker Server\Data\Databases\  
dove [unità] è l'unità principale da cui viene avviato il sistema.
  - macOS: /Libreria/FileMaker Server/Data/Database/  
O è possibile inserire i file in una cartella aggiuntiva per i database (opzionale).
2. Nella cartella in cui è stato posizionato il database, creare una cartella RC\_Data\_FMS, se non esiste già.



3. Nella cartella RC\_Data\_FMS, creare una cartella con un nome che corrisponda a quello del database. Ad esempio, se il database si chiama Clienti, creare una cartella Clienti. Inserire gli oggetti memorizzati esternamente nella nuova cartella creata.

**Nota** Quando i database sono ospitati su FileMaker Server, non è possibile per più database condividere una cartella di oggetti Contenitore comune. Gli oggetti Contenitore di ciascun database devono essere in una cartella con lo stesso nome del database.

4. Per i file che verranno condivisi da macOS, modificarli in modo che appartengano al gruppo **fmsadmin**.

Per informazioni sul caricamento manuale dei database, vedere la [Guida di FileMaker Server](#).

## Campi Contenitore e download progressivo

Il Motore per la Pubblicazione Web supporta il download progressivo di file audio (.mp3), file video (consigliati .mov, .mp4 e .avi) e file PDF per i contenitori interattivi. Ad esempio, un utente Web può iniziare a visualizzare un filmato anche se il relativo file non è ancora stato scaricato completamente. Per consentire il download progressivo, può essere necessario creare i file utilizzando opzioni che supportano lo streaming o che ottimizzano la visualizzazione sul Web. Ad esempio, creare file PDF utilizzando l'opzione per ottimizzare la visualizzazione sul Web.

Se l'impostazione di FileMaker Server **Usare SSL per connessioni database** è selezionata, FileMaker Server utilizza connessioni sicure per trasmettere i dati con HTTPS.

- I dati Contenitore interattivi vengono scaricati con HTTPS.
- La sicurezza dei dati è la stessa che avrebbero se la soluzione ospitata fosse un database locale, poiché non vengono creati file cache temporanei e i dati vengono crittografati durante la trasmissione.

Se l'impostazione di FileMaker Server **Usare SSL per connessioni database** non è selezionata, le connessioni utilizzate da FileMaker Server per trasmettere i dati non sono crittografate durante la trasmissione e i dati vengono trasmessi con HTTP.

- I client FileMaker visualizzano i dati Contenitore interattivi con un ritardo minimo.
- FileMaker Server memorizza i dati dei campi Contenitore in una cartella cache sul server quando un client FileMaker Pro, FileMaker Go o Web richiede tali dati. I dati possono rimanere nella cartella cache sul server per due ore, limite entro il quale FileMaker Server svuota periodicamente la cartella cache. I dati non vengono memorizzati nella cache locale del client.

Se viene modificata l'impostazione **Usare SSL per connessioni database**, affinché la modifica abbia effetto, riavviare il servizio FileMaker Server (Windows) o i processi in background di FileMaker Server (macOS).

## Modalità di visualizzazione dei dati dei campi Contenitore

Se un database viene pubblicato utilizzando il Motore per la Pubblicazione Web, agli oggetti del campoContenitore vengono applicate le seguenti limitazioni:

- Gli utenti Web non possono modificare o fare aggiunte al contenuto di un campo Contenitore. Gli utenti Web non possono utilizzare campi Contenitore per caricare oggetti nel database.
- Per i database che utilizzano un campo Contenitore con le miniature abilitate, il Motore per la Pubblicazione Web scarica l'intero file e non una miniatura.

## Script FileMaker e Pubblicazione Web personalizzata

La funzione Gestisci script in FileMaker Pro consente di combinare varie attività e di automatizzare quelle eseguite più di frequente. Quando sono utilizzati con Pubblicazione Web personalizzata, gli script FileMaker permettono agli utenti Web di eseguire altre operazioni o una serie di operazioni.

FileMaker supporta molte istruzioni di script nella Pubblicazione Web personalizzata. Gli utenti Web possono eseguire diverse operazioni automatizzate quando si usano gli script in una stringa di query per un URL. Per visualizzare le istruzioni di script supportate dalla Pubblicazione Web personalizzata, nella finestra Area di lavoro script di FileMaker Pro, fare clic sul pulsante **Compatibilità** e selezionare **Pubblicazione Web personalizzata**. Le istruzioni di script non disattivate sono supportate per la Pubblicazione Web personalizzata. Per informazioni sulla creazione degli script, consultare la [Guida di FileMaker Pro](#).

### Suggerimenti e considerazioni sugli script

Anche se molte istruzioni di script funzionano allo stesso modo sul Web, alcune hanno un funzionamento diverso. Vedere “Comportamento degli script nelle soluzioni di Pubblicazione Web personalizzata” a pagina 19. Prima di condividere il database, testare tutti gli script che saranno eseguiti da un browser Web. Effettuare l’accesso con account diversi per assicurarsi che questi funzionino nel modo previsto per tutti i client. Controllare nel file di registro del Motore per la Pubblicazione Web (wpe.log) gli eventuali errori relativi agli script. Vedere “Utilizzo del registro del Motore per la Pubblicazione Web” a pagina 94.

Tenere presente quanto segue:

- Considerare i valori che uno script può restituire. Prepararsi a gestire tutti i dati restituiti. In FileMaker Pro, uno script può restituire tutti i record di una tabella o del gruppo trovato corrente. Tuttavia, se uno script restituisce tutti i record di una tabella, un’applicazione Web può esaurire la memoria nel tentativo di elaborarli. Considerare la possibilità di utilizzare il parametro query `-max` con le query XML o il metodo `setRange()` con le query PHP per limitare il numero di record restituiti.
- Utilizzare gli account e i privilegi per limitare il gruppo di script eseguibili da un utente Web. Verificare che gli script contengano soltanto istruzioni di script compatibili con il Web e offrire accesso soltanto agli script che possono essere usati da un browser Web.
- Considerare inoltre gli effetti collaterali degli script che eseguono una combinazione di istruzioni controllate dai privilegi di accesso. Ad esempio, se uno script contiene un’istruzione per eliminare record e l’utente Web non accede con un account che autorizza tale operazione, essa non verrà eseguita. Tuttavia, l’esecuzione dello script potrebbe continuare, provocando risultati imprevisti.
- Nella finestra Area di lavoro script, garantire privilegi di accesso completo a uno script per permettere allo script di eseguire operazioni a cui non si desidera consentire l’accesso ai singoli utenti. Ad esempio, è possibile impedire agli utenti di eliminare record con i rispettivi account e privilegi, ma nello stesso tempo è possibile consentire loro di eseguire uno script che elimini determinati tipi di record alle condizioni predefinite all’interno di uno script.
- Per consentire agli script di installare i plug-in per le soluzioni di Pubblicazione Web personalizzata e FileMaker WebDirect, utilizzare la FileMaker Server Admin Console per abilitare l’impostazione **Consenti all’istruzione di script Installa file plug-in di aggiornare i plug-in per la Pubblicazione Web**. Per impedire allo script di installare i plug-in per le soluzioni di pubblicazione Web, deselezionare questa impostazione.

- Alcuni script che utilizzano un'istruzione di un client FileMaker Pro potrebbero richiedere un'ulteriore istruzione di script Salva record/richieste per salvare i dati sull'host. Poiché gli utenti Web non hanno una connessione diretta con l'host, non ricevono notifica delle variazioni dei dati. Ad esempio, le funzioni come le liste valori condizionali non sono così immediate per gli utenti del Web perché i dati devono essere salvati sull'host prima che i loro effetti vengano visualizzati nel campo lista valori.
- Gli script che modificano i dati devono comprendere l'istruzione di script Salva record/richieste perché le modifiche apportate ai dati non sono visibili nel browser finché i dati non vengono salvati o "inviati" al server. Tra queste vi sono diverse istruzioni di script come Taglia, Copia e Incolla. Molte azioni ad istruzione singola devono essere convertite in script per includere l'istruzione Salva record/richieste. Per creare script da eseguire da un browser Web includere l'istruzione Salva record/richieste alla fine dello script per assicurarsi che tutte le modifiche vengano salvate.
- Per creare script condizionali in base al tipo di client, utilizzare la funzione Get (VersioneApplicazione). Se il valore restituito comprende una stringa "Motore per la Pubblicazione Web", significa che l'utente sta accedendo al database con la Pubblicazione Web personalizzata. Per informazioni sulle funzioni, consultare la [Guida di FileMaker Pro](#).
- Aprire tutti gli script che gli utenti Web potrebbero eseguire e assicurarsi che lo script venga eseguito correttamente quando il database è ospitato come una soluzione di Pubblicazione Web personalizzata. Verificare che lo script utilizzi solo istruzioni di script supportate per la Pubblicazione Web personalizzata, come descritto in precedenza.

### Comportamento degli script nelle soluzioni di Pubblicazione Web personalizzata

Alcune istruzioni di script funzionano in modo diverso nelle soluzioni di Pubblicazione Web personalizzata e in FileMaker Pro. Per informazioni sulla compatibilità, consultare la [Guida di FileMaker Pro](#).

Gli script nelle soluzioni di Pubblicazione Web personalizzata non possono eseguire script in altri file FileMaker, a meno che i file siano ospitati sulla stessa installazione di FileMaker Server e che lo stesso privilegio esteso per la Pubblicazione Web personalizzata sia attivato negli altri file.

### Trigger di script e soluzioni di Pubblicazione Web personalizzata

In FileMaker Pro, i trigger di script possono essere attivati sia dagli script che dalle operazioni dell'utente (come ad esempio facendo clic su un campo). Nella Pubblicazione Web personalizzata, invece, solo gli script possono attivare i trigger di script. Per informazioni sui trigger di script, consultare la [Guida di FileMaker Pro](#).

**Nota** Nelle soluzioni di Pubblicazione Web personalizzata, i trigger di script SuAperturaPrimaFinestra e SuChiusuraUltimaFinestra non sono attivati. Eseguire manualmente gli script utilizzando il parametro query –script per XML o il metodo `newPerformScriptCommand()` per PHP.

# Capitolo 3

## Pubblicazione Web personalizzata con XML

### Creazione di siti Web dinamici con il Motore per la Pubblicazione Web

Il Motore per la Pubblicazione Web offre la Pubblicazione Web personalizzata per FileMaker Server con la pubblicazione dati XML. La Pubblicazione Web personalizzata offre diversi vantaggi:

- **Personalizzazione:** Si può stabilire in che modo gli utenti Web interagiscono con i dati FileMaker, e in che modo i dati vengono visualizzati nei browser Web.
- **Scambio dati:** Utilizzando XML FileMaker, è possibile scambiare i dati FileMaker con altri siti Web e applicazioni.
- **Integrazione dati:** È possibile integrare i dati FileMaker in altri siti Web e con altre applicazioni personalizzate e middleware. Si può fare in modo che i dati sembrino appartenere ad un altro sito Web invece di visualizzare un intero formato FileMaker nel browser Web.
- **Sicurezza:** L'amministratore di FileMaker Server può attivare o disattivare le singole applicazioni Pubblicazione Web XML per tutti i database ospitati dal server. Il proprietario del database FileMaker può controllare l'accesso degli utenti a Pubblicazione Web XML per ciascun database.
- **Controllo e filtraggio dei dati pubblicati:** È possibile controllare e filtrare i dati e le informazioni sul tipo di database che si desidera pubblicare, evitando in tal modo un uso non autorizzato del database. Si possono anche nascondere i metadati, come i nomi dei campi e dei database.
- **Uso di uno standard aperto:** Questo permette di avere maggiore accesso a strumenti, a risorse e a personale esperto per la realizzazione di soluzioni di Pubblicazione Web personalizzata. Se si conosce l'XML standard, è possibile partire immediatamente con lo sviluppo di soluzioni dopo avere appreso poche nozioni di base sulla Pubblicazione Web personalizzata con XML, come la sintassi URL e i parametri query da utilizzare.

La Pubblicazione Web personalizzata con XML permette di recuperare i dati dai database FileMaker e di utilizzare facilmente i dati in altri formati di uscita. Utilizzando una richiesta HTTP con i parametri e i comandi query appropriati, è possibile recuperare i dati FileMaker come documento XML.

In seguito i dati XML possono essere utilizzati in altre applicazioni. Vedere "Accesso a dati XML attraverso il Motore per la Pubblicazione Web" a pagina 28.

## Caratteristiche principali della Pubblicazione Web personalizzata con XML

La Pubblicazione Web personalizzata con XML di FileMaker Server offre molte importanti funzioni:

- I database sono ospitati su FileMaker Server e non è necessario che FileMaker Pro sia in esecuzione.
- È possibile utilizzare l'elaborazione lato server di XML utilizzando Javascript.
- Come in FileMaker Pro, l'accesso ai dati, ai formati e ai campi dipende dalle impostazioni degli account utente definite nei privilegi di accesso del database. Per il Motore per la Pubblicazione Web sono stati inoltre introdotti altri miglioramenti per la sicurezza. Vedere "Protezione dei database pubblicati" a pagina 14.
- Gli utenti Web possono eseguire script complessi, a più istruzioni. FileMaker supporta molte istruzioni di script nella Pubblicazione Web personalizzata. Vedere "Script FileMaker e Pubblicazione Web personalizzata" a pagina 18.
- È possibile trasferire un valore di parametro ad uno script FileMaker. Vedere "Parametro query –script.param (Trasferisci parametro a script)" a pagina 60, l'"Parametro query –script.prefind.param (Trasferisci parametro a script prima di trovare)" a pagina 60 e l'"Parametro query –script.presort.param (Trasferisci parametro a script prima di ordinare)" a pagina 61.
- La nuova grammatica XML `fmresultset` permette di accedere ai campi per nome e di gestire i dati (portale) `relatedset`.
- Per accedere ai dati in un database, è necessario specificare il formato. Vedere capitolo 5, "Nomi validi utilizzati nelle stringhe di query XML".

## Requisiti per la pubblicazione sul Web

### Requisiti per la pubblicazione di un database mediante Pubblicazione Web personalizzata

Per pubblicare i database che utilizzano la Pubblicazione Web personalizzata con XML è necessario:

- Una distribuzione di FileMaker Server che comprende:
  - Un server Web, Microsoft IIS (Windows) o Apache (macOS);
  - Il server database FileMaker abilitato per la Pubblicazione Web personalizzata;
  - Il Motore per la Pubblicazione Web, installato e configurato;
- Uno o più database FileMaker Pro ospitati da FileMaker Server;
- Indirizzo IP o nome di dominio dell'host del server Web;
- Un browser Web e la possibilità di accedere al server Web per sviluppare e controllare la soluzione di Pubblicazione Web personalizzata.

Vedere la [Guida all'installazione e alla configurazione di FileMaker Server](#).

## Cosa si richiede agli utenti Web per accedere alla soluzione di Pubblicazione Web personalizzata

Per accedere alla soluzione di Pubblicazione Web personalizzata che fa uso di XML, gli utenti Web devono disporre di:

- Un browser Web;
- Accesso a Internet o a una rete intranet e al server Web;
- Indirizzo IP o nome di dominio dell'host del server Web.

Se il database è protetto da password, gli utenti Web devono anche introdurre un nome utente e una password per l'account database.

## Connessione a Internet o ad una rete intranet

Quando si pubblicano database su Internet o su una rete intranet, è necessario che sul computer host sia in esecuzione FileMaker Server e i database da condividere devono essere ospitati e disponibili. Inoltre:

- Pubblicare il database su un computer con una connessione continua a Internet o a un'intranet. È possibile pubblicare database anche in assenza di una connessione continua, ma in tal caso i database risulteranno disponibili agli utenti solo quando il computer che li ospita è connesso a Internet o alla rete intranet.
- Il computer host del server Web, che fa parte della distribuzione di FileMaker Server, deve avere un nome di dominio o un indirizzo IP *statico* (permanente) dedicato. Se ci si collega a Internet con un provider di servizi Internet (ISP), l'indirizzo IP potrebbe essere *assegnato in modo dinamico* (ossia essere diverso ad ogni connessione). L'utilizzo di un indirizzo IP dinamico rende più difficile l'individuazione dei database da parte degli utenti. Se non si è certi del tipo di accesso di cui si dispone, contattare il provider di servizi Internet o l'amministratore della rete.

## Operazioni successive

Di seguito sono riportati alcuni suggerimenti per iniziare a sviluppare soluzioni di Pubblicazione Web personalizzata:

- Se già non è stato fatto, utilizzare la Admin Console di FileMaker Server per attivare la Pubblicazione Web personalizzata. Vedere la [Guida di FileMaker Server](#) e la [Guida all'installazione e alla configurazione di FileMaker Server](#).
- In FileMaker Pro, aprire il database FileMaker che si desidera pubblicare e assicurarsi che il database abbia i privilegi estesi appropriati attivati per Pubblicazione Web personalizzata. Vedere "Attivazione della Pubblicazione Web personalizzata in un database" a pagina 13.
- Per sapere come accedere ai dati nei database FileMaker con XML, vedere "Accesso a dati XML attraverso il Motore per la Pubblicazione Web" a pagina 28.

# Capitolo 4

## Accesso a dati XML con il Motore per la Pubblicazione Web

È possibile ottenere e aggiornare dati FileMaker in formato XML (Extensible Markup Language) usando il Motore per la Pubblicazione Web. Numerosi utenti, organizzazioni e aziende utilizzano XML per trasferire informazioni sui prodotti, dati di transazioni, inventari e altri dati di carattere professionale.

### Uso di Pubblicazione Web personalizzata con il linguaggio XML

Se conoscete XML standard, è possibile cominciare subito a usare il Motore per la Pubblicazione Web dopo aver appreso poche nozioni di base sulla Pubblicazione Web personalizzata con XML, come ad esempio la sintassi URL e i parametri query da usare.

Utilizzando richieste HTTP URL con comandi e parametri query specifici FileMaker, è possibile eseguire query su un database ospitato da FileMaker Server e scaricare i dati risultanti in formato XML. Ad esempio è possibile eseguire query in un database per tutti i record compresi in un certo codice postale, e usare i dati XML risultanti nel modo desiderato.

Vedere la [Knowledge Base di FileMaker](#).

**Nota** Il Motore per la Pubblicazione Web genera dati XML corretti e conformi alla specifica XML 1.0. Per informazioni dettagliate sulla corretta composizione dei dati XML, vedere le specifiche XML all'indirizzo [www.w3.org](http://www.w3.org).

### Differenze di importazione/esportazione XML tra Motore per la Pubblicazione Web e FileMaker Pro

Il Motore per la Pubblicazione Web e FileMaker Pro consentono entrambi di utilizzare dati XML con database FileMaker. Tuttavia, esistono alcune differenze significative tra i due metodi:

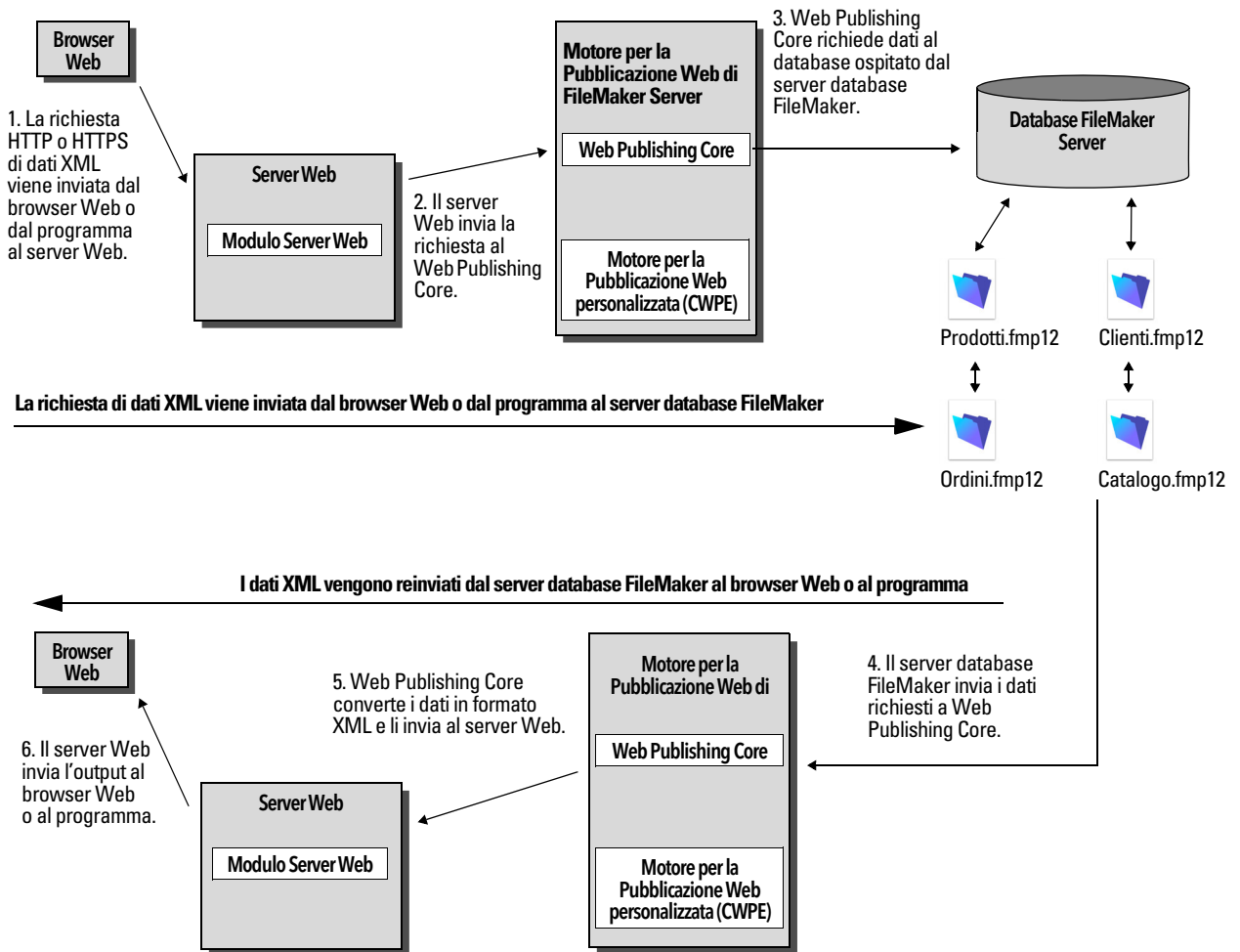
- Per accedere ai dati XML, il Motore per la Pubblicazione Web supporta le grammatiche `fmresultset`, `FMPXMLRESULT` e `FMPXMLLAYOUT`. Per l'importazione XML, FileMaker Pro usa la grammatica `FMPXMLRESULT`, e per l'esportazione utilizza la grammatica `FMPXMLRESULT`. Vedere "Accesso a dati XML attraverso il Motore per la Pubblicazione Web" a pagina 28.
- Per accedere a dati XML con il Motore per la Pubblicazione Web, usare una stringa di query Motore per la Pubblicazione Web in un URL. Per importare ed esportare dati XML con FileMaker Pro, si utilizzano i comandi di menu o gli script FileMaker Pro.
- Il Motore per la Pubblicazione Web è basato sul server e può essere installato sullo stesso host o su un host diverso da FileMaker Server. L'importazione e l'esportazione di dati XML di FileMaker Pro sono basate sul desktop.
- È possibile accedere in modo dinamico ai dati XML dai database FileMaker utilizzando le richieste URL con il Motore per la Pubblicazione Web. La funzione di esportazione XML di FileMaker Pro genera un file di dati XML predefinito.
- Lavorare con dati XML attraverso il Motore per la Pubblicazione Web è un'operazione interattiva. L'importazione e l'esportazione di dati XML di FileMaker Pro sono operazioni batch.

- Il Motore per la Pubblicazione Web può accedere ai dati XML da un portale FileMaker; FileMaker Pro no.
- Il Motore per la Pubblicazione Web può accedere ai dati dei campi Contenitore; FileMaker Pro no.
- Il Motore per la Pubblicazione Web fornisce un accesso in tempo reale ai dati FileMaker via HTTP o HTTPS; FileMaker Pro no.

**Nota** Per informazioni sull'utilizzo di FileMaker Pro per importare ed esportare i dati in formato XML, consultare la [Guida di FileMaker Pro](#).

### Modalità di generazione di dati XML da una richiesta da parte del Motore per la Pubblicazione Web

Quando una richiesta di dati XML viene inviata al server Web, il Motore per la Pubblicazione Web esegue una query al database FileMaker e restituisce i dati sotto forma di documento XML.





## Procedura generale per accedere a dati XML dal Motore per la Pubblicazione Web

Di seguito vi è una descrizione della procedura da seguire per usare il Motore per la Pubblicazione Web per accedere a dati XML da un database FileMaker:

1. In Admin Console di FileMaker Server, assicurarsi che la Pubblicazione XML sia attivata. Vedere la [Guida di FileMaker Server](#).
2. In FileMaker Pro, aprire ogni database FileMaker da pubblicare e assicurarsi che abbia attivato il privilegio esteso fmxml per la Pubblicazione Web personalizzata XML. Vedere “Attivazione della Pubblicazione Web personalizzata in un database” a pagina 13.

Per accedere ai dati XML di un portale, impostare la visualizzazione per il formato del database su **Visualizza come modulo** oppure **Visualizza come lista**. Se un utente o uno script cambia la visualizzazione del formato del database in **Visualizza come tabella**, soltanto il primo record correlato (prima riga del portale) è accessibile come dati XML.

I dati XML vengono restituiti in un ordine corrispondente a quello in cui gli oggetti dei campi erano stati aggiunti al formato. Se si vuole che l'ordine dei dati XML corrisponda a quello di visualizzazione dei campi sulla schermata (dall'alto verso il basso, da sinistra verso destra), selezionare tutti i campi, raggrupparli, quindi separarli. Questa procedura permette di ripristinare l'ordine del formato in modo che corrisponda a quello dello schermo.

3. Inviare una richiesta HTTP o HTTPS, sotto forma di un URL che specifichi la grammatica XML FileMaker, un comando query e uno o più parametri query FileMaker, al Motore per la Pubblicazione Web attraverso un modulo HTML, un collegamento HREF o uno script nel proprio programma o nella propria pagina Web. È anche possibile digitare l'URL in un browser Web. Per informazioni su come specificare l'URL, vedere la sezione seguente, “Sintassi URL per dati XML e oggetti Contenitore”. Per informazioni sui comandi e i parametri query, vedere “Utilizzo di stringhe di query FileMaker per richiedere dati XML” a pagina 38, e capitolo 5, “Nomi validi utilizzati nelle stringhe di query XML”.
4. Il Motore per la Pubblicazione Web usa la grammatica specificata nell'URL per generare dati XML che contengono i risultati della richiesta, come ad esempio un set di record dal database, e li restituisce al programma o al browser Web.
5. Il browser Web, se ha un parser XML, visualizza i dati, oppure il programma usa i dati nel modo specificato.

## Sintassi URL per dati XML e oggetti Contenitore

Questa sezione descrive la sintassi URL per utilizzare il Motore per la Pubblicazione Web per accedere ai dati XML e agli oggetti Contenitore dai database FileMaker.

### Sintassi URL per dati XML

La sintassi URL per utilizzare il Motore per la Pubblicazione Web per accedere ai dati XML dai database FileMaker è:

```
<schema>://<host>[:<porta>]/fmi/xml/<grammatica_xml>.xml [?<stringa di query>]
```

dove:

- <schema> può essere il protocollo HTTP o HTTPS.
- <host> è l'indirizzo IP o il nome di dominio dell'host su cui è installato il server Web.

- `<porta>` è opzionale e indica la porta utilizzata dal server Web. Se non è specificata nessuna porta, viene utilizzata la porta predefinita del protocollo (porta 80 per HTTP o porta 443 per HTTPS).
- `<grammatica_xml>` è il nome della grammatica XML FileMaker. I valori possibili sono `fmresultset`, `FMPXMLRESULT` oppure `FMPXMLLAYOUT`. Vedere “Utilizzo della grammatica `fmresultset`” a pagina 29 e “Utilizzo di altre grammatiche XML FileMaker” a pagina 33.
- `<stringa di query>` è la combinazione di un comando query e di uno o più parametri query per la Pubblicazione XML FileMaker (il comando `-dbnames` non richiede alcun parametro). Vedere il “Utilizzo di stringhe di query FileMaker per richiedere dati XML” a pagina 38, e il capitolo 5, “Nomi validi utilizzati nelle stringhe di query XML”.

**Nota** La sintassi URL, compresi i nomi dei comandi e dei parametri query, fa distinzione tra maiuscole e minuscole, eccetto per parti della stringa di query. La maggior parte dell'URL è in caratteri minuscoli, con l'eccezione dei due nomi della grammatica che sono in maiuscolo: `FMPXMLRESULT` e `FMPXMLLAYOUT`. Per informazioni sulle regole per le maiuscole/minuscole della stringa di query, vedere “Linee guida per l'utilizzo dei comandi e dei parametri query” a pagina 43.

### Esempi

```
http://server.company.com/fmi/xml/fmresultset.xml?-db=products&-lay=sales
&-findall
http://192.168.123.101/fmi/xml/FMPXMLRESULT.xml?-db=products&-lay=sales
&-findall
```

### Sintassi URL per oggetti Contenitore FileMaker in soluzioni XML

In un documento XML generato per una soluzione XML, la sintassi utilizzata per fare riferimento a un oggetto Contenitore è diversa per i campi Contenitore che memorizzano l'oggetto effettivo nel database e i campi Contenitore che memorizzano un riferimento all'oggetto.

#### Se un campo Contenitore memorizza l'oggetto effettivo nel database

L'elemento `<data>` del campo Contenitore utilizza la seguente sintassi URL relativa per fare riferimento all'oggetto:

```
<data>/fmi/xml/cnt/data.<estensione>?<stringa di query></data>
```

dove `<estensione>` è l'estensione del nome del file che identifica il tipo di oggetto, come ad esempio `.jpg`. L'estensione del file imposta il tipo MIME per consentire al browser Web di identificare correttamente i dati Contenitore. Per informazioni su `<stringa di query>`, vedere la sezione precedente, “Sintassi URL per dati XML”.

#### Esempio

```
<data>/fmi/xml/cnt/data.jpg?-db=products&-lay=sales
&-field=product_image(1)&-recid=2</data>
```

**Nota** Nell'XML generato per un campo Contenitore, il valore del parametro query `-field` è un nome di campo completamente qualificato. Il numero tra parentesi indica il numero di ripetizioni per il campo Contenitore e viene generato sia per i campi multipli sia per quelli non multipli. Vedere “Sintassi di un nome di campo completamente qualificato” a pagina 45.

Per recuperare i dati Contenitore dal database, utilizzare la seguente sintassi:

```
<schema>://<host>[:<porta>]/fmi/xml/cnt/data.<estensione>?<stringa di query>
```

Per informazioni su <schema>, <host> o <porta>, vedere la sezione precedente, “Sintassi URL per dati XML”.

### Esempio

```
http://www.company.com/fmi/xml/cnt/data.jpg?-db=products&-lay=sales
&-field=product_image(1)&-recid=2
```

### Se un campo Contenitore memorizza un riferimento al file invece di un oggetto effettivo

L'elemento <data> del campo Contenitore contiene un percorso relativo che fa riferimento all'oggetto.

### Esempio

```
<data>/images/logo.jpg</data>
```

**Nota** L'oggetto Contenitore riferito deve essere memorizzato nella cartella Web di FileMaker Pro quando il record viene creato o modificato, e quindi copiato o spostato in una cartella con la stessa posizione relativa nella cartella principale del software del server Web. Vedere “Pubblicazione del contenuto dei campi Contenitore sul Web” a pagina 15.

### Se un campo Contenitore è vuoto

L'elemento <data> del campo Contenitore è vuoto.

### Codifica testo URL

L'URL per accedere ai dati XML e agli oggetti Contenitore deve essere codificato in formato UTF-8 (Unicode Transformation 8 Bit). Vedere “Dati codificati nel formato UTF-8” a pagina 38.

### Esempio

Per impostare il valore del campo “info” su fiancée, è possibile utilizzare il seguente URL:

```
http://server.company.com/fmi/xml/fmresultset.xml?-db=members
&-lay=relationships&-recid=2&info=fianc%C3%A9e&-edit
%C3%A9 è la rappresentazione UTF-8 con codifica URL del carattere é.
```

Vedere le specifiche URL all'indirizzo [www.w3.org](http://www.w3.org).

## Accesso a dati XML attraverso il Motore per la Pubblicazione Web

Per accedere a dati XML attraverso il Motore per la Pubblicazione Web si utilizza un URL che specifica il nome della grammatica FileMaker da usare, un comando query FileMaker e uno o più parametri query FileMaker. Il Motore per la Pubblicazione Web genera dati XML dal database formattato da uno dei seguenti tipi di grammatica XML:

- **fmresultset:** Questa è la grammatica consigliata per il Motore per la Pubblicazione Web per accedere a dati XML. È flessibile e ottimizzata per un accesso più semplice ai campi per nome e per una manipolazione più semplice dei dati (portale) *relatedset*. Questa grammatica è anche collegata più direttamente alla terminologia FileMaker e alle funzioni come le opzioni di memorizzazione globale e l'identificazione dei campi Riassunto e Calcolo. Per facilitare la pubblicazione sul Web, questa grammatica è stata progettata in modo da essere più dettagliata della grammatica *FMPXMLRESULT*. Vedere "Utilizzo della grammatica fmresultset" a pagina 29.
- **FMPXMLRESULT e FMPXMLLAYOUT:** È anche possibile usare le grammatiche *FMPXMLRESULT* e *FMPXMLLAYOUT* con il Motore per la Pubblicazione Web per accedere ai dati XML. Per utilizzare lo stesso foglio di stile sia per l'esportazione XML sia per la Pubblicazione Web personalizzata, è necessario usare la grammatica *FMPXMLRESULT*. Per accedere alle liste valori e alle informazioni sulla visualizzazione dei campi nei formati è necessario usare la grammatica *FMPXMLLAYOUT*. Vedere "Utilizzo di altre grammatiche XML FileMaker" a pagina 33.

A seconda della grammatica specificata nella richiesta URL, il Motore per la Pubblicazione Web genera un documento XML usando una di queste grammatiche. Ogni documento XML contiene una dichiarazione di namespace XML predefinita per la grammatica. Vedere la sezione successiva, "Nome spazio per XML FileMaker". Utilizzare una di queste grammatiche nel documento o nella pagina Web per visualizzare e utilizzare i dati FileMaker in formato XML.

**Nota** I dati XML generati dal Motore per la Pubblicazione Web sono codificati in formato UTF-8 (Unicode Transformation Format 8). Vedere "Dati codificati nel formato UTF-8" a pagina 38.

### Nome spazio per XML FileMaker

I namespace XML unici servono per distinguere i tag XML dall'applicazione per cui sono stati progettati. Ad esempio, se il documento XML contiene due elementi `<DATABASE>`, uno per i dati XML FileMaker e un altro per i dati XML Oracle, i namespace identificano l'elemento `<DATABASE>` di ognuno.

Il Motore per la Pubblicazione Web genera un namespace predefinito per ogni grammatica.

#### Per questa grammatica Viene generato questo namespace predefinito

fmresultset	xmlns="http://www.filemaker.com/xml/fmresultset"
FMPXMLRESULT	xmlns="http://www.filemaker.com/fmpxmlresult"
FMPXMLLAYOUT	xmlns="http://www.filemaker.com/fmpxmllayout"

## Codici di errore dei database FileMaker

Il Motore per la Pubblicazione Web restituisce un codice di errore negli elementi del codice di errore all'inizio di ogni documento XML che rappresenta l'errore, se presente, nell'esecuzione dell'ultimo comando query eseguito. Nel caso non vi siano errori, viene restituito il valore zero (0).

### Per questa grammatica Viene utilizzata questa sintassi

fmresultset	<error code="0"></error>
FMPXMLRESULT	<ERRORCODE>0</ERRORCODE>
FMPXMLLAYOUT	<ERRORCODE>0</ERRORCODE>

L'elemento del codice di errore nel documento XML indica gli errori relativi al database e alle stringhe della query. Vedere appendice A, "Codici di errore per la Pubblicazione Web personalizzata".

## Recupero di DTD per le grammatiche FileMaker

È possibile recuperare le DTD (Document Type Definition) delle grammatiche FileMaker utilizzando una richiesta HTTP.

### Per questa grammatica Utilizzare questa richiesta HTTP

fmresultset	http://<host>[:<porta>]/fmi/xml/fmresultset.dtd
FMPXMLRESULT	http://<host>[:<porta>]/fmi/xml/FMPXMLRESULT.dtd
FMPXMLLAYOUT	http://<host>[:<porta>]/fmi/xml/FMPXMLLAYOUT.dtd

## Utilizzo della grammatica fmresultset

I nomi degli elementi XML in questa grammatica usano la terminologia FileMaker, e la memorizzazione dei campi è separata dal tipo di campo. La grammatica comprende anche la capacità di identificare i campi Riassunto, Calcolo e globali.

Per usare la grammatica `fmresultset`, specificare il nome indicato di seguito della grammatica `fmresultset` nell'URL che richiede il documento XML dal Motore per la Pubblicazione Web:

```
fmresultset.xml
```

### Esempio

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family
&-findall
```

**Nota** Usare caratteri minuscoli per specificare la grammatica `fmresultset`.

Il Motore per la Pubblicazione Web genera un documento XML usando la grammatica `fmresultset`. Nel documento XML, il Motore per la Pubblicazione Web fa riferimento alla DTD della grammatica `fmresultset` nell'istruzione `<!DOCTYPE>` nella seconda riga del documento, subito dopo l'istruzione `<?xml...?>`. L'istruzione `<!DOCTYPE>` specifica l'URL per scaricare la DTD per la grammatica `fmresultset`.

## Descrizione degli elementi della grammatica fmresultset

La grammatica `fmresultset` è costituita fondamentalmente dagli elementi `<datasource>`, `<metadata>` e `<resultset>`.

### Elemento `<datasource>`

Nella grammatica `fmresultset`, l'elemento `<datasource>` contiene gli attributi `table`, `layout`, `date-format`, `time-format`, `timestamp-format`, `total-count` e `database`.

- L'attributo `date-format` dell'elemento `<datasource>` specifica il formato delle date nel documento XML:

`MM/dd/yyyy`

dove:

- `MM` sono le 2 cifre del mese (da 01 a 12 , dove 01 è gennaio e 12 è dicembre)
- `dd` sono le 2 cifre del giorno del mese (da 01 a 31)
- `yyyy` sono le 4 cifre dell'anno
- L'attributo `time-format` dell'elemento `<datasource>` specifica il formato delle ore nel documento XML:

`HH:mm:ss`

dove:

- `HH` sono le due cifre indicanti l'ora (da 00 a 23, nel formato a 24 ore)
- `mm` sono le due cifre indicanti i minuti (da 00 a 59)
- `ss` sono le 2 cifre indicanti i secondi (da 00 a 59)
- L'attributo `timestamp-format` dell'elemento `<datasource>` combina i formati di `date-format` e `time-format` in un indicatore data e ora:

`MM/dd/yyyy HH:mm:ss`

### Elemento <metadata>

L'elemento <metadata> della grammatica `fmresultset` contiene uno o più elementi <field-definition> e <relatedset-definition>, ognuno dei quali contiene attributi per uno dei campi del gruppo di risultati.

Gli attributi <field-definition> specificano:

- se il campo è di tipo `auto-enter` ("yes" o "no")
- se il campo è di tipo `four-digit-year` ("yes" o "no")
- se il campo è globale ("yes" o "no")
- il numero massimo di valori multipli (attributo `max-repeat`)
- il numero massimo consentito di caratteri (attributo `max-characters`)
- se il campo è di tipo `not-empty` ("yes" o "no")
- se è solo per dati numerici ("yes" o "no")
- il risultato ("text", "number", "date", "time", "timestamp" o "container")
- se il campo è di tipo `time-of-day` ("yes" o "no")
- il tipo ("normal", "calculation" o "summary")
- e il nome del campo (completamente qualificato, se necessario)

L'elemento <relatedset-definition> rappresenta un portale. Ogni campo correlato in un portale è rappresentato dall'elemento <field-definition> contenuto nell'elemento <relatedset-definition>. Se un portale contiene più campi correlati, le definizioni dei campi per i campi correlati vengono raggruppate in un unico elemento <relatedset-definition>.

### Elemento <resultset>

L'elemento <resultset> contiene gli elementi <record> restituiti come risultato di una query e un attributo per il numero totale di record trovati. Ogni elemento <record> contiene i dati dei campi di un record nel gruppo di risultati, compresi gli attributi `mod-id` e `record-id` del record e l'elemento <data> che contiene i dati di un campo del record.

Ogni record in un portale è rappresentato da un elemento <record> all'interno di un elemento <relatedset>. L'attributo `count` dell'elemento <relatedset> specifica il numero di record nel portale, mentre l'attributo `table` specifica la tabella associata al portale.

## Dati XML con la grammatica fmresultset

### Esempio

```
<fmresultset xmlns="http://www.filemaker.com/xml/fmresultset" version="1.0">
  <error code="0" />
  <product build="03/29/2017" name="FileMaker Web Publishing Engine"
    version="16.0.1.0"/>
  <datasource database="art" date-format="MM/dd/yyyy" layout="web3"
    table="art" time-format="HH:mm:ss" timestamp-format="MM/dd/yyyy HH:mm:ss"
    total-count="12"/>
  <metadata>
    <field-definition auto-enter="no" four-digit-year="no" global="no" max-
      repeat="1" name="Title" not-empty="no" numeric-only="no" result="text"
      time-of-day="no" type="normal"/>
    <field-definition auto-enter="no" four-digit-year="no" global="no" max-
      repeat="1" name="Artist" not-empty="no" numeric-only="no" result="text"
      time-of-day="no" type="normal"/>
    <field-definition auto-enter="no" four-digit-year="no" global="no" max-
      repeat="1" name="Style" not-empty="no" numeric-only="no" result="text"
      time-of-day="no" type="normal"/>
    <field-definition auto-enter="no" four-digit-year="no" global="no" max-
      repeat="1" name="length" not-empty="no" numeric-only="no"
      result="number" time-of-day="no" type="calculation"/>
    <relatedset-definition table="artlocations">
      <field-definition auto-enter="no" four-digit-year="no" global="no" max-
        repeat="1" name="artlocations::Location" not-empty="no" numeric-
        only="no" result="text" time-of-day="no" type="normal"/>
      <field-definition auto-enter="no" four-digit-year="no" global="no" max-
        repeat="1" name="artlocations::Date" not-empty="no" numeric-only="no"
        result="date" time-of-day="no" type="normal"/>
    </relatedset-definition>
  </metadata>
  <resultset count="1" fetch-size="1">
    <record mod-id="6" record-id="14">
      <field name="Title">
        <data>Spring in Giverny 3</data>
      </field>
      <field name="Artist">
        <data>Claude Monet</data>
      </field>
      <field name="Style">
        <data/>
      </field>
      <field name="length">
        <data>19</data>
      </field>
      <relatedset count="0" table="artlocations"/>
    </record>
  </resultset>
</fmresultset>
```



## Utilizzo di altre grammatiche XML FileMaker

Le altre grammatiche XML FileMaker contengono informazioni sui tipi di campi, liste valori e formati. Dal punto di vista funzionale `FMPXMLRESULT` è equivalente a `fmresultset`. Per accedere alle liste valori e alle informazioni sulla visualizzazione dei campi nei formati è necessario usare la grammatica `FMPXMLLAYOUT`. Le grammatiche `FMPXMLRESULT` e `FMPXMLLAYOUT` sono più compatte per lo scambio dei dati.

Per utilizzare la grammatica `FMPXMLRESULT`, specificare il seguente nome di grammatica nell'URL che richiede il documento XML dal Motore per la Pubblicazione Web:

```
FMPXMLRESULT.xml
```

### Esempio

```
http://192.168.123.101/fmi/xml/FMPXMLRESULT.xml?-db=employees&-lay=family
&-findall
```

Per utilizzare la grammatica `FMPXMLLAYOUT`, specificare il seguente nome di grammatica con il comando query `-view` nell'URL che richiede il documento XML dal Motore per la Pubblicazione Web:

```
FMPXMLLAYOUT.xml
```

### Esempio

```
http://192.168.123.101/fmi/xml/FMPXMLLAYOUT.xml?-db=employees&-lay=family
&-view
```

**Nota** I nomi delle grammatiche `FMPXMLRESULT` e `FMPXMLLAYOUT` devono essere in maiuscolo.

Nel documento XML generato, il Motore per la Pubblicazione Web fa riferimento alla DTD della grammatica nell'istruzione `<!DOCTYPE>` nella seconda riga del documento, subito dopo l'istruzione `<?xml . . . ?>`. L'istruzione `<!DOCTYPE>` specifica l'URL per scaricare la DTD per la grammatica.

## Descrizione degli elementi della grammatica `FMPXMLRESULT`

Nella grammatica `FMPXMLRESULT`, l'elemento `<DATABASE>` contiene gli attributi `NAME`, `RECORDS`, `DATEFORMAT`, `LAYOUT` e `TIMEFORMAT`.

L'attributo `DATEFORMAT` dell'elemento `<DATABASE>` specifica il formato delle date nel documento XML. L'attributo `TIMEFORMAT` dell'elemento `<DATABASE>` specifica il formato delle ore nel documento XML. I formati della data e dell'ora delle grammatiche `FMPXMLRESULT` e `fmresultset` sono gli stessi. Vedere "Descrizione degli elementi della grammatica `fmresultset`" a pagina 30.

L'elemento `<METADATA>` della grammatica `FMPXMLRESULT` contiene uno o più elementi `<FIELD>`, ognuno dei quali contiene informazioni per uno dei campi/una delle colonne del gruppo di risultati, compresi il nome del campo definito nel database, il tipo di campo, l'autorizzazione `Yes` o `No` per i campi vuoti (attributo `EMPTYOK`) e il numero massimo di valori ripetuti (attributo `MAXREPEAT`). I valori validi per i tipi di campo sono `TEXT`, `NUMBER`, `DATE`, `TIME`, `TIMESTAMP` e `CONTAINER`.

L'elemento `<RESULTSET>` contiene tutti gli elementi `<ROW>` restituiti come risultato di una query e un attributo per il numero totale di record trovati. Ogni elemento `<ROW>` contiene i dati dei campi/delle colonne di una riga nel gruppo di risultati. Questi dati comprendono `RECORDID` e `MODID` per la riga (vedere "Parametro query –modid (ID modifica)" a pagina 56) e l'elemento `<COL>`. L'elemento `<COL>` contiene i dati di un campo/una colonna nella riga in cui più elementi `<DATA>` rappresentano uno dei valori in un campo multiplo o portale.

## Dati XML con la grammatica FMPXMLRESULT

### Esempio

```
<FMPXMLRESULT xmlns="http://www.filemaker.com/fmpxmlresult">
  <ERRORCODE>0</ERRORCODE>
  <PRODUCT BUILD="03/29/2017" NAME="FileMaker Web Publishing Engine"
    VERSION="16.0.1.0"/>
  <DATABASE DATEFORMAT="MM/dd/yyyy" LAYOUT="web" NAME="art" RECORDS="12"
    TIMEFORMAT="HH:mm:ss"/>
  <METADATA>
    <FIELD EMPTYOK="YES" MAXREPEAT="1" NAME="Title" TYPE="TEXT"/>
    <FIELD EMPTYOK="YES" MAXREPEAT="1" NAME="Artist" TYPE="TEXT"/>
    <FIELD EMPTYOK="YES" MAXREPEAT="1" NAME="Image" TYPE="CONTAINER"/>
  </METADATA>
  <RESULTSET FOUND="1">
    <ROW MODID="7" RECORDID="4">
      <COL>
        <DATA>Village Market</DATA>
      </COL>
      <COL>
        <DATA>Camille Pissarro</DATA>
      </COL>
      <COL>
        <DATA>/fmi/xml/cnt/Untitled.pct?-db=art&-lay=web&-recid=4
          &-field=Image(1)
        </DATA>
      </COL>
    </ROW>
  </RESULTSET>
</FMPXMLRESULT>
```

L'ordine degli elementi `<COL>` corrisponde all'ordine degli elementi `<FIELD>` nell'elemento `<METADATA>`; ad esempio, se i campi "Title" e "Artist" sono elencati nell'elemento `<METADATA>`, "Village Market" e quindi "Camille Pissarro" sono elencati nello stesso ordine negli elementi `<RESULTSET>` e `<ROW>`.

## Descrizione degli elementi della grammatica FMPXMLLAYOUT

Nella grammatica FMPXMLLAYOUT, l'elemento <LAYOUT> contiene il nome del formato, il nome del database ed elementi <FIELD> per ogni campo trovato nel formato corrispondente nel database. Ogni elemento <FIELD> descrive il tipo di stile del campo e contiene l'attributo VALUELIST per qualsiasi lista valori associata del campo.

L'elemento <VALUELISTS> contiene uno o più elementi <VALUELIST> per ogni lista valori trovata nel formato; ognuno comprende il nome della lista valori e un elemento <VALUE> per ogni valore nella lista.

A seconda delle opzioni selezionate nella finestra di dialogo **Specifica i campi per la lista valori** nel database FileMaker, l'elemento <VALUE> contiene un attributo DISPLAY che contiene il valore solo nel primo campo, solo nel secondo campo o in entrambi i campi di una lista valori. Ad esempio, si supponga che il primo campo in una lista valori contenga il numero ID dello stile artistico (come ad esempio "100") e il secondo campo visualizzi il nome associato allo stile artistico (come ad esempio "Impressionism"). Di seguito è visualizzato un riepilogo dei contenuti dell'attributo DISPLAY quando le varie combinazioni delle opzioni sono selezionate nella finestra di dialogo **Specifica i campi per la lista valori**:

- Se **Visualizza anche i valori secondo campo** non è selezionato, l'attributo DISPLAY contiene solo il valore del primo campo della lista valori.

### Esempio

L'attributo DISPLAY contiene solo il numero ID dello stile artistico:

```
<VALUELISTS>
  <VALUELIST NAME="style">
    <VALUE DISPLAY="100">100</VALUE>
    <VALUE DISPLAY="101">101</VALUE>
    <VALUE DISPLAY="102">102</VALUE>
  </VALUELIST>
</VALUELISTS>
```

- Se **Visualizza anche i valori secondo campo** e **Mostra valori solo dal secondo campo** sono entrambi selezionati, l'attributo DISPLAY contiene solo il valore del secondo campo.

### Esempio

L'attributo DISPLAY contiene solo il nome dello stile artistico:

```
<VALUELISTS>
  <VALUELIST NAME="style">
    <VALUE DISPLAY="Impressionism">100</VALUE>
    <VALUE DISPLAY="Cubism">101</VALUE>
    <VALUE DISPLAY="Abstract">102</VALUE>
  </VALUELIST>
</VALUELISTS>
```

- Se **Visualizza anche i valori secondo campo** è selezionato e **Mostra valori solo dal secondo campo** non è selezionato, l'attributo DISPLAY contiene i valori di entrambi i campi di una lista valori.

### Esempio

L'attributo DISPLAY contiene sia il numero ID dello stile artistico che il nome dello stile artistico:

```
<VALUELISTS>
  <VALUELIST NAME="style">
    <VALUE DISPLAY="100 Impressionism">100</VALUE>
    <VALUE DISPLAY="101 Cubism">101</VALUE>
    <VALUE DISPLAY="102 Abstract">102</VALUE>
  </VALUELIST>
</VALUELISTS>
```

Per i campi Data, Ora e Indicatore data e ora, i dati per le liste valori sono formattati con il formato "fm" per quel tipo di campo. I formati "fm" sono MM/DD/YYYY per la data, hh:mm:ss per l'ora e MM/DD/YYYY hh:mm:ss per l'indicatore data e ora. Per esempio, se la lista valori "birthdays" è utilizzata per il menu a tendina di un campo "birthdate" di un formato, e il campo "birthdate" è di tipo Data, i valori risultanti da tale lista valori saranno tutti in formato data "fm".

**Nota** Se in un formato due campi di tipo diverso condividono la stessa lista valori, il tipo del primo campo determina il formato dei dati della lista valori.

## Dati XML con FMPXMLLAYOUT grammatica

### Esempio

```
<FMPXMLLAYOUT xmlns="http://www.filemaker.com/fmpxmllayout">
  <ERRORCODE>0</ERRORCODE>
  <PRODUCT BUILD="03/29/2017" NAME="FileMaker Web Publishing Engine"
    VERSION="16.0.1.0"/>
  <LAYOUT DATABASE="art" NAME="web2">
    <FIELD NAME="Title">
      <STYLE TYPE="EDITTEXT" VALUELIST=""/>
    </FIELD>
    <FIELD NAME="Artist">
      <STYLE TYPE="EDITTEXT" VALUELIST=""/>
    </FIELD>
    <FIELD NAME="Image">
      <STYLE TYPE="EDITTEXT" VALUELIST=""/>
    </FIELD>
    <FIELD NAME="artlocations::Location">
      <STYLE TYPE="EDITTEXT" VALUELIST=""/>
    </FIELD>
    <FIELD NAME="artlocations::Date">
      <STYLE TYPE="EDITTEXT" VALUELIST=""/>
    </FIELD>
    <FIELD NAME="Style">
      <STYLE TYPE="POPUPMENU" VALUELIST="style"/>
    </FIELD>
  </LAYOUT>
  <VALUELISTS>
    <VALUELIST NAME="style">
      <VALUE DISPLAY="Impressionist">Impressionist</VALUE>
      <VALUE DISPLAY="Modern">Modern</VALUE>
      <VALUE DISPLAY="Abstract">Abstract</VALUE>
    </VALUELIST>
  </VALUELISTS>
</FMPXMLLAYOUT>
```

## Dati codificati nel formato UTF-8

Tutti i dati XML generati dal Motore per la Pubblicazione Web sono codificati in formato UTF-8 (Unicode Transformation 8 Bit). Questo formato comprime i dati dal formato standard Unicode a 16 bit nel formato a 8 bit per i caratteri ASCII. Per supportare la codifica Unicode e UTF-8, sono necessari parser XML.

La codifica UTF-8 comprende una rappresentazione diretta dei caratteri usati nella lingua inglese che utilizzano i valori 0-127 del set di caratteri standard ASCII e fornisce la codifica a più byte per i caratteri Unicode con valori superiori.

**Nota** Usare un browser Web o un editor di testo che supporti file UTF-8.

Di seguito sono descritte le caratteristiche del formato di codifica UTF-8.

- Tutti i caratteri ASCII sono caratteri UTF-8 a 1 byte. Una stringa ASCII valida è una stringa UTF-8 valida.
- Qualsiasi carattere non ASCII (qualsiasi carattere con gruppo di bit di ordine elevato) fa parte di un carattere a più byte.
- Il primo byte di un carattere UTF-8 qualsiasi indica il numero di ulteriori byte nel carattere.
- Il primo byte di un carattere a più byte si riconosce facilmente dai byte successivi; pertanto è facile individuare l'inizio di un carattere da una posizione qualsiasi in un flusso di dati.
- La conversione tra UTF-8 e Unicode è un'operazione facile.
- La codifica UTF-8 effettua una discreta compressione. È più efficace dell'Unicode per i testi con un'alta percentuale di caratteri ASCII. Nella peggiore delle ipotesi, una stringa UTF-8 è più grande della stringa Unicode corrispondente solo del 50%.

## Utilizzo di stringhe di query FileMaker per richiedere dati XML

Per richiedere dati XML ad un database FileMaker usare i comandi e i parametri query FileMaker in una stringa di query.

### Esempio

È possibile utilizzare il comando query `-findall` nella seguente stringa di query in un URL per richiedere un elenco di tutti i prodotti in un database FileMaker "products":

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=products  
&-lay=sales&-findall
```

Una stringa di query deve contenere un solo comando query, come `-new`. La maggior parte dei comandi query richiede diversi parametri query corrispondenti nella stringa di query. Ad esempio, tutti i comandi query tranne `-dbnames` necessitano del parametro `-db`, che specifica il database in cui eseguire la query.

È anche possibile usare comandi e parametri query in un URL.

Questa sezione contiene un riepilogo dei comandi e dei parametri query FileMaker. Per informazioni sul loro utilizzo in una stringa di query, vedere “Nomi validi utilizzati nelle stringhe di query XML” a pagina 42.

#### Utilizzare questo nome di comando query Per eseguire questo comando

-dbnames	Recuperare i nomi di tutti i database ospitati e condivisi sul Web.
-delete	Eliminare il record.
-dup	Duplicare il record.
-edit	Modificare il record.
-find	Trovare il(i) record.
-findall	Trovare tutti i record.
-findany	Trovare un record a caso.
-findquery	Eseguire una richiesta di ricerca complessa o composta.
-layoutnames	Recuperare i nomi di tutti i formati disponibili per un database ospitato e condiviso sul Web.
-new	Aggiungere un nuovo record.
-scriptnames	Recuperare i nomi di tutti gli script disponibili per un database ospitato e condiviso sul Web.
-view	Recuperare le informazioni sul formato da un database se è specificata la grammatica FMPXMLLAYOUT. Recuperare la sezione <metadata> del documento XML e un gruppo di record vuoto se è specificata la grammatica fmresultset o FMPXMLRESULT.

#### Utilizzare questi nomi di parametri query Con questi comandi query

-db (nome database)	Necessario con tutti i comandi query, tranne -dbnames
-delete.related	Opzionale con -edit
-field	Necessario per specificare un campo in un URL per richieste contenitore. Vedere “Sintassi URL per oggetti Contenitore FileMaker in soluzioni XML” a pagina 26.
fieldname	Con -edit è necessario il nome di almeno un campo. Opzionale con -find. Vedere “Parametro query fieldname (Nome campo non Contenitore)” a pagina 52.
fieldname.op (operatore)	Opzionale con -find
-lay (nome formato)	Necessario con tutti i comandi query, tranne -dbnames, -layoutnames e -scriptnames
-lay.response (passaggio ad altro formato per la risposta XML)	Opzionale con tutti i comandi query, tranne -dbnames, -layoutnames e -scriptnames
-lop (operatore logico)	Opzionale con -find
-max (record max)	Opzionale con -find, -findall e -findquery
-modid (ID modifica)	Opzionale con -edit
-query	Necessario con le richieste di ricerca composta -findquery
-recid (ID record)	Necessario con -edit, -delete, -dup. Opzionale con -find
-relatedsets.filter	Opzionale con -find, -findall, -findany, -edit, -new, -dup e -findquery
-relatedsets.max	Opzionale con -find, -edit, -new, -dup e -findquery
-script (eseguire lo script)	Opzionale con -find, -findall, -findany, -new, -edit, -delete, -dup, -view e -findquery

**Utilizzare questi nomi di parametri query Con questi comandi query**

<code>-script.param</code> (trasferire il valore di un parametro allo script specificato da <code>-script</code> )	Opzionale con <code>-script</code> e <code>-findquery</code>
<code>-script.prefind</code> (eseguire lo script prima di <code>-find</code> , <code>-findany</code> e <code>-findall</code> )	Opzionale con <code>-find</code> , <code>-findany</code> , <code>-findall</code> e <code>-findquery</code>
<code>-script.prefind.param</code> (trasferire il valore di un parametro allo script specificato da <code>-script.prefind</code> )	Opzionale con <code>-script.prefind</code> e <code>-findquery</code>
<code>-script.presort</code> (eseguire lo script prima dell'ordinamento)	Opzionale con <code>-find</code> , <code>-findall</code> e <code>-findquery</code>
<code>-script.presort.param</code> (trasferire il valore di un parametro allo script specificato da <code>-script.presort</code> )	Opzionale con <code>-script.presort</code> e <code>-findquery</code>
<code>-skip</code> (saltare i record)	Opzionale con <code>-find</code> , <code>-findall</code> e <code>-findquery</code>
<code>-sortfield.[1-9]</code> (campo di ordinamento)	Opzionale con <code>-find</code> , <code>-findall</code> e <code>-findquery</code>
<code>-sortorder.[1-9]</code> (criterio di ordinamento)	Opzionale con <code>-find</code> , <code>-findall</code>

## Commutazione dei formati per una risposta XML

Il parametro query `-lay` specifica il formato da usare per richiedere dati XML. Spesso, lo stesso formato è adatto per elaborare i dati prodotti dalla richiesta. In alcuni casi, si potrebbe voler cercare i dati utilizzando un formato che contiene campi che, per motivi di sicurezza, non esistono in un altro formato che si desidera utilizzare per visualizzare i risultati (per ricercare dati in un campo, il campo deve trovarsi sul formato specificato nella richiesta XML).

Per specificare un formato diverso per visualizzare una risposta XML anziché il formato usato per elaborare la richiesta XML, è possibile usare il parametro query opzionale `-lay.response`.

### Esempio

La richiesta di seguito ricerca i valori superiori a 100.000 nel campo "Salary" nel formato "Budget". I dati risultanti vengono visualizzati usando il formato "ExecList", che non comprende il campo "Salary".

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=Budget&Salary=100000&Salary.op=gt&-find&-lay.response=ExecList
```



## Comprendere come una richiesta XML viene elaborata

Esistono diversi parametri query che incidono sull'elaborazione di una richiesta XML e sulla generazione di un documento XML.

L'ordine in cui FileMaker Server e il Motore per la Pubblicazione Web elaborano una richiesta XML è:

1. Elaborazione del parametro query `-lay`.
2. Impostazione di valori campo globali specificati nella query (parte `.global=` di un URL).
3. Elaborazione del parametro query `-script.prefind`, se specificato.
4. Elaborazione dei comandi query, quali `-find o -new`.
5. Elaborazione del parametro query `-script.presort`, se specificato.
6. Ordinamento dei dati risultanti, se è stato specificato un ordinamento.
7. Elaborazione del parametro query `-script`, se specificato.
8. Elaborazione del parametro query `-lay.response` per passare ad un formato diverso, se questo è specificato.
9. Generazione del documento XML.

Se una delle fasi elencate sopra genera un codice di errore, l'elaborazione delle richieste si arresta; tutte le fasi successive non vengono eseguite. Vengono tuttavia eseguite tutte le fasi precedenti della richiesta.

Esempio: una richiesta che elimina il record corrente, ordina i record, quindi esegue uno script. Se il parametro `-sortfield` specifica un campo che non esiste, la richiesta elimina il record corrente e restituisce il codice di errore 102 ("Manca un campo"), ma non esegue lo script.

## Risoluzione dei problemi di accesso a documenti XML

In caso di problemi di accesso a documenti XML con il Motore per la Pubblicazione Web, verificare che:

- I privilegi estesi nel database siano impostati per la Pubblicazione Web personalizzata XML e assegnati ad un account utente. Vedere "Attivazione della Pubblicazione Web personalizzata in un database" a pagina 13.
- Il database sia ospitato sul componente server database della distribuzione di FileMaker Server e venga aperto da FileMaker Server. Vedere la [Guida di FileMaker Server](#).
- Gli eventuali nomi utente e password utilizzati per il database siano corretti.
- Il componente server Web della distribuzione di FileMaker Server sia in esecuzione.
- Il componente Motore per la Pubblicazione Web della distribuzione di FileMaker Server sia in esecuzione.
- La Pubblicazione XML è abilitata nel componente Motore per la Pubblicazione Web. Vedere la [Guida di FileMaker Server](#).

# Capitolo 5

## Nomi validi utilizzati nelle stringhe di query XML

Questo capitolo descrive i nomi validi dei comandi e dei parametri query utilizzabili in una stringa di query XML quando si accede ai dati FileMaker utilizzando il Motore per la Pubblicazione Web.

### Comandi e parametri query

Di seguito è riportato un elenco completo dei nomi dei comandi e dei parametri query:

Nomi comandi query	Nomi parametri query
-dbnames (vedere pagina 47.)	-db (vedere pagina 51.)
-delete (vedere pagina 47.)	-field (vedere pagina 52.)
-dup (vedere pagina 48.)	fieldname (vedere pagina 52.)
-edit (vedere pagina 48.)	fieldname.op (vedere pagina 53.)
-find, -findall, -findany (vedere pagina 49.)	-lay (vedere pagina 54.)
-findquery (vedere pagina 49.)	-lay.response (vedere pagina 54.)
-layoutnames (vedere pagina 50.)	-lop (vedere pagina 55.)
-new (vedere pagina 50.)	-max (vedere pagina 55.)
-scriptnames (vedere pagina 51.)	-modid (vedere pagina 56.)
-view (vedere pagina 51.)	-query (vedere pagina 56.)
	-recid (vedere pagina 57.)
	-relatedsets.filter (vedere pagina 58.)
	-relatedsets.max (vedere pagina 59.)
	-script (vedere pagina 59.)
	-script.param (vedere pagina 60.)
	-script.prefind (vedere pagina 60.)
	-script.prefind.param (vedere pagina 60.)
	-script.presort (vedere pagina 61.)
	-script.presort.param (vedere pagina 61.)
	-skip (vedere pagina 62.)
	-sortfield. [1-9] (vedere pagina 62.)
	-sortorder. [1-9] (vedere pagina 63.)

**Importante** Il parametro `-lay` per specificare un formato è necessario con tutti i comandi query, tranne `-dbnames`, `-layoutnames` e `-scriptnames`.

## Linee guida per l'utilizzo dei comandi e dei parametri query

Durante l'uso dei comandi e dei parametri query in una stringa di query, tenere a mente le seguenti linee guida:

- Una stringa di query deve contenere un solo comando query; né uno di più né uno di meno. Ad esempio, una stringa di query può contenere `-new` per aggiungere un nuovo record, ma non può contenere `-new` e `-edit` nella stessa stringa.
- Per la maggior parte dei comandi query sono necessari parametri query corrispondenti nella stringa di query. Ad esempio, tutti i comandi query tranne `-dbnames` necessitano del parametro `-db`, che specifica il database in cui eseguire la query. Consultare la tabella dei parametri necessari nella sezione "Utilizzo di stringhe di query FileMaker per richiedere dati XML" a pagina 38.
- Per i parametri query e i nomi dei campi, specificare il valore particolare da utilizzare, come ad esempio `-db=employees`. Per i comandi query, non specificare un segno o un valore uguale ("=") dopo il nome del comando, come ad esempio `-findall`.
- Il Motore per la Pubblicazione Web converte tutte le parole riservate in lettere minuscole, compresi i comandi query, i parametri query e i valori dei comandi dove si prevedono valori specifici (ad esempio: `-lop=and`, `-lop=or`, `-sortorder=ascend`, `-sortorder=descend`, `-max=all`).
- I nomi dei database, i nomi dei formati e i nomi dei campi utilizzati nelle stringhe di query non fanno differenza tra maiuscole e minuscole, come ad esempio `-lay=mylayout` utilizzato per indicare il nome di formato `MyLayout`.
- Si raccomanda di non utilizzare punti o parentesi nei nomi dei campi. In alcuni casi, i nomi dei campi contenenti punti possono funzionare, ma i nomi dei campi con le seguenti eccezioni non possono mai essere utilizzati:
  - Il punto non può essere seguito da un numero. Ad esempio, `myfield.9` è un nome di campo non valido.
  - Il punto non può essere seguito dalla stringa di testo `op` (dalle lettere "op"). Ad esempio, `myfield.op` è un nome di campo non valido.
  - Il punto non può essere seguito dalla stringa di testo `global` (la parola "global"). Ad esempio, `myfield.global` è un nome di campo non valido.Non è possibile accedere attraverso XML ai nomi dei campi che contengono queste eccezioni utilizzando una query HTTP. Questi costrutti sono riservati agli ID dei record, come descritto nella sezione, "Sintassi di un nome di campo completamente qualificato" a pagina 45.
- Per il comando `-find`, per il valore di un campo non si fa differenza tra maiuscole e minuscole. Per esempio, è possibile utilizzare `Field1=Blue` o `Field1=blue`. Per i comandi `-new` e `-edit`, le maiuscole e le minuscole utilizzate per il valore di un campo vengono mantenute e memorizzate nel database esattamente nel modo in cui sono specificate nella stringa di query. Ad esempio, `LastName=Doe`.

## Analisi dei comandi query

Il Motore per la Pubblicazione Web analizza i comandi query nel seguente ordine e, al primo errore, interrompe l'analisi delle query XML. Se viene restituito un codice di errore, questo corrisponde al primo errore identificato.

### 1. La query ha un comando e il comando query è valido?

Si ha un errore se la query non ha il comando o utilizza un comando sconosciuto.

#### Esempio

```
-database
```

### 2. La query ha due comandi?

#### Esempio

```
-find&-edit
```

### 3. La query ha un valore non valido per un comando o un parametro?

#### Esempio

```
-lop=amd
```

### 4. La query non ha il parametro del nome del database richiesto (parametro `-db`)?

### 5. La query non ha il parametro del nome del formato richiesto (parametro `-lay`)?

### 6. La query ha un ordinamento non valido?

### 7. La query ha parametri di campo non validi?

**Nota** Se una query contiene informazioni valide ma estranee, viene elaborata senza errori. Ad esempio, se si specifica il parametro `-lop` in un comando `-delete`, il parametro `-lop` viene ignorato poiché non compromette la validità della query e non la rende ambigua.

Per informazioni sugli specifici codici di errore restituiti, vedere appendice A, "Codici di errore per la Pubblicazione Web personalizzata."

## Sintassi di un nome di campo completamente qualificato

Un nome di campo completamente qualificato identifica un'istanza esatta di un campo. Poiché i campi con nomi comuni possono essere basati su diverse tabelle, in alcuni casi è necessario utilizzare nomi completamente qualificati per evitare errori.

La sintassi per specificare un nome di campo completamente qualificato è:

`nome-tabella::nome-campo(numero-ripetizioni).id-record`

dove:

- `nome-tabella` è il nome della tabella che contiene il campo. Il nome della tabella è necessario solo se il campo non si trova nella tabella sottostante del formato specificato nella stringa di query.
- `nome-campo(numero-ripetizioni)` è il valore specifico in un campo multiplo; è necessario soltanto per i campi multipli. Il numero di ripetizioni comincia a contare dal numero 1. Ad esempio, `nome-campo(2)` indica il secondo valore nel campo multiplo. Se non si specifica un numero di ripetizioni per un campo multiplo, viene utilizzato il primo valore del campo multiplo. Il numero-ripetizioni è necessario per i comandi query `-new` e `-edit` che coinvolgono campi multipli, ma non è necessario per il comando `-find`.
- `id-record` è l'ID del record, ed è necessario soltanto se si usa una stringa di query per aggiungere o modificare record nei campi portale. Vedere le sezioni "Aggiunta di record a un portale", e "Modifica dei record in un portale". di seguito. `id-record` è necessario per i comandi query `-new` e `-edit` che coinvolgono campi portale, ma non è necessario per il comando `-find`.

**Nota** Per poter essere accessibili, i campi devono trovarsi sul formato specificato nella stringa di query.

## Utilizzo dei comandi query con campi portale

Le seguenti sezioni descrivono il modo in cui i comandi query funzionano con i campi portale.

### Aggiunta di record a un portale

Per aggiungere un nuovo record ad un portale mentre si aggiunge un record principale, usare il comando query `-new` ed eseguire le seguenti operazioni nella stringa di query per la richiesta:

- Utilizzare il nome di campo completamente qualificato per il campo del portale correlato.
- Specificare 0 per ID record dopo il nome del campo del portale correlato.
- Specificare almeno uno dei campi per il record principale prima di specificare il campo del portale correlato.
- Specificare i dati per il campo di confronto (campo chiave) nel record principale.

### Esempio

L'URL di seguito aggiunge contemporaneamente un nuovo record principale Employee per John Doe e un nuovo record correlato per Jane nel portale. Il nome della tabella correlata è Dependents e il nome del campo correlato nel portale è Names. Il campo di confronto, ID, memorizza il codice di un dipendente.

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family
&FirstName=John&LastName=Doe&ID=9756&Dependents::Names.0=Jane&-new
```

**Nota** Per ogni richiesta è possibile aggiungere un solo record correlato ad un portale.

## Modifica dei record in un portale

Per modificare uno o più record in un portale, usare il comando `-edit` e un ID record per specificare il record principale che contiene i record del portale da modificare. Specificare il particolare record del portale da modificare utilizzando il suo ID record in un nome di campo completamente qualificato. È possibile determinare un ID record dall'attributo ID record dell'elemento `<record>` nell'elemento `<relatedset>` nei dati XML. Vedere "Utilizzo della grammatica `fmresultset`" a pagina 29.

### Esempi

L'URL di seguito modifica un record in un portale dove l'ID del record principale è 1001. `Dependents` è il nome della tabella correlata, `Names` è il nome del campo correlato nel portale e il 2 in `Names.2` è l'ID di un record del portale.

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family
&-recid=1001&Dependents::Names.2=Kevin&-edit
```

Come utilizzare una richiesta per modificare i record di più portali attraverso il record principale:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family
&-recid=1001&Dependents::Names.2=Kevin&Dependents::Names.5=Susan&-edit
```

È anche possibile utilizzare il comando `-edit` e specificare 0 come ID del record del portale per aggiungere un nuovo record correlato nel portale per un record principale esistente.

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family
&-recid=1001&Dependents::Names.0=Timothy&-edit
```

## Eliminazione dei record del portale

Per eliminare i record del portale, utilizzare il parametro `-delete.related` con il comando `-edit` al posto del comando `-delete`.

### Esempi

L'URL di seguito elimina il record "1001" dalla tabella "employees":

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family
&-recid=1001&-delete
```

Il seguente URL, invece, elimina un record del portale con un ID record "3" dalla tabella correlata "Dependents" con ID record principale "1001".

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family
&-recid=1001&-delete.related=Dependents.3&-edit
```

Vedere "Parametro query `-delete.related` (Elimina record portale)" a pagina 52.

## Interrogazione dei campi del portale

In una soluzione con molti record correlati, l'interrogazione e l'ordinamento dei record del portale potrebbe richiedere molto tempo. Per limitare il numero di record e di righe da visualizzare nel set correlato, utilizzare i parametri `-relatedsets.filter` e `-relatedsets.max` con richieste di ricerca. Vedere il "Parametro query `-relatedsets.filter` (Filtra record portale)" a pagina 58 e il "Parametro query `-relatedsets.max` (Limita record portale)" a pagina 59.

## Sintassi per specificare un campo globale

La sintassi per specificare un campo globale è:

```
nome-tabella::nome-campo(numero-ripetizioni).global
```

dove `global` identifica un campo che usa la memorizzazione globale. Per informazioni su `nome-tabella` e `nome-campo(numero-ripetizioni)`, vedere “Sintassi di un nome di campo completamente qualificato” a pagina 45. Per informazioni sui campi globali, consultare la [Guida di FileMaker Pro](#).

È necessario usare la sintassi `.global` per identificare un campo globale in una stringa di query. Il Motore per la Pubblicazione Web imposta i valori dei parametri per i campi globali prima di eseguire il comando query o impostare qualsiasi altro valore di parametro nella stringa di query. Per le richieste XML dirette, i valori globali scadono immediatamente dopo l’invio della richiesta.

Se non si usa la sintassi `.global` per identificare un campo globale in una stringa di query, il Motore per la Pubblicazione Web valuta il campo globale insieme con il resto della stringa di query senza impostare prima il valore del campo globale.

### Esempio

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees  
&-lay=departments&Country.global=USA&-recid=1&-edit
```

## Riferimento comando query

Questa sezione contiene informazioni sui comandi query disponibili per le richieste XML.

### Comando query `-dbnames` (Nomi database)

Recupera i nomi di tutti i database ospitati da FileMaker Server e attivati per la Pubblicazione Web personalizzata con XML.

**Parametri query obbligatori:** (nessuno)

### Esempio

Per recuperare i nomi dei database:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-dbnames
```

### Comando query `-delete` (Elimina record)

Elimina il record specificato dal parametro `-recid`.

**Parametri query obbligatori:** `-db`, `-lay`, `-recid`

**Parametro query opzionale:** `-script`

### Esempio

Per eliminare un record:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees  
&-lay=departments&-recid=4&-delete
```

### Comando query `-dup` (Duplica record)

Duplica il record specificato dal parametro `-recid`.

**Parametri query obbligatori:** `-db`, `-lay`, `-recid`

**Parametro query opzionale:** `-script`

#### Esempio

Per duplicare il record specificato:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-recid=14&-dup
```

### Comando query `-edit` (Modifica record)

Aggiorna il record specificato dal parametro `-recid`, popolando i campi con il contenuto di qualsiasi coppia nome/valore campo. Il parametro `-recid` indica il record da modificare.

**Parametri query obbligatori:** `-db`, `-lay`, `-recid`, uno o più nomi di campo

**Parametro query opzionale:** `-modid`, `-script`, nome di campo

**Nota** Il comando `-edit` può essere utilizzato per modificare i record in un portale. Vedere “Modifica dei record in un portale” a pagina 46.

#### Esempio

Per modificare un record:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-recid=13&Country=USA&-edit
```



## Comandi query `-find`, `-findall` o `-findany` (Trova record)

Invia una richiesta di ricerca utilizzando determinati criteri.

**Parametri query obbligatori:** `-db`, `-lay`

**Parametri query opzionali:** `-recid`, `-lop`, `-op`, `-max`, `-skip`, `-sortorder`, `-sortfield`, `-script`, `-script.prefind`, `-script.presort`, nome di campo

### Esempi

Per trovare un record in base al nome del campo:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=family&Country=USA&-find
```

Per trovare un record in base all'ID:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family
&-recid=427&-find
```

Per trovare tutti i record del database, utilizzare `-findall`:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family
&-findall
```

Per trovare un record a caso, utilizzare `-findany`:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-lay=family
&-findany
```

### Note

- Non è possibile specificare più volte il nome di un campo in una singola richiesta; FileMaker Server analizza tutti i valori, ma utilizza solo l'ultimo valore analizzato.
- Quando si utilizza il comando `-findall`, evitare problemi di sovraccarico della memoria del computer specificando un numero massimo predefinito di record da restituire per pagina utilizzando il parametro query `-max`.

## Comando query `-findquery` (Ricerca composta)

Invia una richiesta di ricerca utilizzando più richieste di ricerca e richieste di omissione record.

**Parametri query obbligatori:** `-db`, `-lay`, `-query`

**Parametri query opzionali:** `-max`, `-skip`, `-sortorder`, `-sortfield`, `-script`, `-script.prefind`, `-script.presort`

### Esempio

Ricerca i record per gatti o cani che non hanno il nome "Fluffy":

```
http://host/fmi/xml/fmresultset.xml?-db=vetclicnic&-lay=animals
&-query=(q1);(q2);!(q3)&-q1=typeofanimal&-q1.value=Cat&-q2=typeofanimal
&-q2.value=Dog&-q3=name&-q3.value=Fluffy&-findquery
```

### Utilizzo del comando `-findquery` per le ricerche composte

Un'istruzione `-findquery` è costituita da quattro parti nel seguente ordine:

- Parametro `-query`.
- Dichiarazioni di richiesta query, costituite dalle dichiarazioni dell'identificatore di query e dalle operazioni della richiesta.
- Definizioni del valore e del campo di ricerca per tutti gli identificativi delle query.
  - Definire gli identificatori di query. Un identificatore di query è la lettera "q" seguita da un numero. Ad esempio: `-q1`
  - Definire i valori degli identificatori di query con il parametro. Ad esempio: `-q1.value=fieldvalue`
  - Definire gli operatori degli identificatori di query includendoli come parte dell'espressione `fieldvalue`. Ad esempio, per utilizzare un asterisco come operatore "inizia con": `-q1.value=fieldvalue*`
- Comando `-findquery`, alla fine dell'istruzione completa.

Per informazioni sull'utilizzo del parametro `-query`, vedere "Parametro query `-query` (Richiesta di ricerca composta)" a pagina 56.

### Comando query `-layoutnames` (Nomi formati)

Recupera i nomi di tutti i formati disponibili per un determinato database ospitato da FileMaker Server e attivato per la Pubblicazione Web personalizzata con XML.

**Parametri query obbligatori:** `-db`

#### Esempio

Per recuperare i nomi dei formati disponibili:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-layoutnames
```

### Comando query `-new` (Nuovo record)

Crea un nuovo record e lo popola con il contenuto di qualsiasi coppia nome/valore campo.

**Parametri query obbligatori:** `-db`, `-lay`

**Parametro query opzionale:** uno o più nomi di campo, `-script`

**Nota** Per informazioni su come aggiungere nuovi dati in un portale, vedere "Aggiunta di record a un portale" a pagina 45.

#### Esempio

Per aggiungere un nuovo record:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees  
&-lay=departments&Country=Australia&-new
```

## Comando query `-scriptnames` (Nomi script)

Recupera i nomi di tutti gli script disponibili per un determinato database ospitato da FileMaker Server e attivato per la Pubblicazione Web personalizzata con XML.

**Parametri query obbligatori:** `-db`

### Esempio

Per recuperare i nomi di tutti gli script:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees&-scriptnames
```

## Comando query `-view` (Visualizza le informazioni sul formato)

Se è specificata la grammatica `FMPXMLLAYOUT`, recupera le informazioni sul formato da un database e le visualizza nella grammatica `FMPXMLLAYOUT`. Se è stata specificata una grammatica di dati (`fmresultset` o `FMPXMLRESULT`), recupera la sezione di metadati del documento XML e un gruppo di record vuoto.

**Parametri query obbligatori:** `-db`, `-lay`

**Parametro query opzionale:** `-script`

### Esempi

Per recuperare informazioni sul formato:

```
http://192.168.123.101/fmi/xml/FMPXMLLAYOUT.xml?-db=employees  
&-lay=departments&-view
```

Per recuperare informazioni sui metadati:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees  
&-lay=departments&-view
```

## Riferimento parametro query

Questa sezione contiene informazioni sui parametri query disponibili per le richieste XML.

### Parametro query `-db` (Nome database)

Specifica il database a cui è applicato il comando query.

**Valore:** Nome del database senza l'eventuale estensione del nome del file.

**Nota** Per specificare il nome del database per il parametro `-db` nelle stringhe di query, non comprendere l'estensione del nome del file. Il nome effettivo del database può includere un'estensione; tuttavia le estensioni non sono consentite come valori per il parametro `-db`.

**Obbligatorio con:** Tutti i comandi query eccetto `-dbnames`

### Esempio

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees  
&-lay=departments&-findall
```

### Parametro query `-delete.related` (Elimina record portale)

Elimina un record dal un campo del portale.

**Opzionale con:** Comando query `-edit`

**Richiede:** Un nome tabella correlato e un id record

#### Esempio

L'esempio di seguito elimina il record di un portale con ID "20" dalla tabella correlata "jobtable" con ID record principale "7".

```
http://host/fmi/xml/fmresultset.xml?-db=career&-lay=applications&-recid=7
&-delete.related=jobtable.20&-edit
```

### Parametro query `-field` (Nome campo Contenitore)

Specifica il nome di un campo Contenitore.

**Obbligatorio con:** Richieste di dati in un campo Contenitore

Vedere "Sintassi URL per dati XML e oggetti Contenitore" a pagina 25.

### Parametro query `fieldname` (Nome campo non Contenitore)

I nomi dei campi sono utilizzati per controllare i criteri del comando query `-find` o per modificare il contenuto di un record. Per specificare un valore per un campo non Contenitore per un comando o un parametro query, usare il nome del campo senza il trattino (-) come porzione del nome della coppia nome/valore.

**Nome:** Nome del campo nel database FileMaker. Se il campo non è nella tabella sottostante del formato specificata nella stringa di query, il nome del campo deve essere completamente qualificato.

Si raccomanda di non utilizzare punti o parentesi nei nomi dei campi. In alcuni casi, i nomi dei campi contenenti punti possono funzionare, ma i nomi dei campi con le seguenti eccezioni non possono mai essere utilizzati:

- Il punto non può essere seguito da un numero. Ad esempio, `myfield.9` è un nome di campo non valido.
- Il punto non può essere seguito dalla stringa di testo `op` (dalle lettere "op"). Ad esempio, `myfield.op` è un nome di campo non valido.
- Il punto non può essere seguito dalla stringa di testo `global` (la parola "global"). Ad esempio, `myfield.global` è un nome di campo non valido.

Non è possibile accedere attraverso XML ai nomi dei campi che contengono queste eccezioni utilizzando una query HTTP. L'utilizzo dei punti nei nomi dei campi dovrebbe essere riservato agli ID dei record, come descritto nella sezione "Sintassi di un nome di campo completamente qualificato" a pagina 45.

**Valore:** Per i comandi query `-new` e `-edit`, specificare il valore da memorizzare nel campo nel record corrente. Per i comandi query `-find`, specificare il valore da ricercare nel campo. I valori dei campi Data, Ora o Indicatore data e ora devono essere indicati nel formato "fm" per quel tipo di campo. I formati "fm" sono MM/DD/YYYY per la data, hh:mm:ss per l'ora e MM/DD/YYYY hh:mm:ss per l'indicatore data e ora.

**Obbligatorio con:** Comando query `-edit`

**Opzionale con:** Comandi query `-new` e `-find`

**Esempio**

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-op=eq&FirstName=Sam&-max=1&-find
```

**Nota** Non è possibile specificare più volte il nome di un campo in una singola richiesta; FileMaker Server analizza tutti i valori, ma utilizza solo l'ultimo valore analizzato.

**Parametro query fieldname.op (Operatore di confronto)**

Specifica l'operatore di confronto da applicare al nome del campo che precede l'operatore. Gli operatori di confronto sono usati con il comando query `-find`.

**Valore:** L'operatore che si desidera utilizzare. Gli operatori validi sono:

Parola chiave	Operatore FileMaker Pro equivalente
eq	=parola
cn	*parola*
bw	parola*
ew	*parola
gt	> parola
gte	>= parola
lt	< parola
lte	<= parola
neq	ometti, parola

**Opzionale con:** Comando query `-find`

**Richiede:** Un nome di campo e un valore

La sintassi per specificare un operatore di confronto è:

nome-tabella::nome-campo=nome-valore&tabella::nome-campo.op=simbolo-op  
dove:

- nome-tabella è la tabella che contiene il campo; è necessario soltanto se il campo non si trova nella tabella di origine del formato specificato nella stringa di query.
- simbolo-op è una delle parole chiave nella tabella precedente, come ad esempio cn.

**Esempio**

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&name=Tim&name.op=cn&-find
```

**Nota** La parola chiave bw non funziona con stringhe data, ora o indicatore data e ora o con l'operatore di ricerca data corrente (//).

È possibile utilizzare qualsiasi operatore di ricerca di FileMaker Pro includendolo come parte dei criteri di ricerca invece di specificare la parola chiave operatore fieldname.op. Ad esempio, per trovare una serie di valori utilizzando l'operatore di ricerca intervallo ( . . . ), non specificare nessuna parola chiave operatore. Utilizzare invece i caratteri " . . . " tra i valori dell'intervallo nei criteri di ricerca.

### Esempio

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&IDnum=915...925&-find
```

Per informazioni sugli operatori che è possibile utilizzare per trovare il testo, consultare la [Guida di FileMaker Pro](#).

### Parametro query -lay (Formato)

Specifica il formato del database da utilizzare.

**Valore:** Nome del formato

**Obbligatorio con:** Tutti i comandi query tranne -dbnames, -layoutnames e -scriptnames

### Esempio

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-view
```

### Parametro query -lay.response (Applicazione del formato per la risposta)

Specifica che FileMaker Server deve usare il formato specificato dal parametro -lay per l'elaborazione di una richiesta e applicare il formato specificato dal parametro -lay.response per l'elaborazione della risposta XML.

Se non si include il parametro -lay.response , FileMaker Server usa il formato specificato dal parametro -lay durante l'elaborazione simultanea della richiesta e della risposta.

È possibile usare il parametro -lay.response per le richieste XML.

**Valore:** Nome del formato

**Opzionale con:** Tutti i comandi query eccetto -dbnames, -layoutnames e -scriptnames.

### Esempio

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=Budget&Salary=100000&Salary.op=gt&-find&-lay.response=ExecList
```

### Parametro query `-lop` (Operatore logico)

Specifica il modo in cui i criteri di ricerca nel comando query `-find` sono combinati come ricerca “and” oppure “or”.

**Valore:** and o or

Se il parametro query `-lop` non è compreso, il comando query `-find` utilizza il valore “and”.

**Opzionale con:** Comando query `-find`

**Nota** Non supportato dal comando query `-findquery`.

#### Esempio

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&Last+Name=Smith&Birthdate=2/5/1972&-lop=and&-find
```

### Parametro query `-max` (Record max)

Specifica il numero massimo di record restituiti.

**Valore:** Un numero, oppure utilizzare il valore `all` per fare in modo che vengano restituiti tutti i record. Se `-max` non è specificato, vengono restituiti tutti i record.

**Opzionale con:** Comandi query `-find`, `-findall` e `-findquery`

**Nota** Il parametro query `-max` non influisce sui valori restituiti per i record del portale. Per limitare il numero delle righe restituite per i record del portale, vedere “Parametro query `-relatedsets.max` (Limita record portale)” a pagina 59.

#### Esempio

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-max=10&-findall
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-max=all&-findall
```

### Parametro query -modid (ID modifica)

L'ID modifica è un contatore incrementale che specifica la versione corrente di un record. Specificando un ID modifica quando si usa un comando query `-edit`, è possibile essere certi di modificare la versione corrente di un record. Se il valore ID modifica specificato non corrisponde al valore ID modifica corrente del database, il comando query `-edit` non viene accettato e viene restituito un codice di errore.

**Valore:** Un ID modifica, che è un identificatore univoco della versione corrente di un record in un database FileMaker.

**Opzionale con:** Comando query `-edit`

**Richiede:** Parametro `-recid`

#### Esempio

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-recid=22&-modid=6&last_name=Jones&-edit
```

### Parametro query -query (Richiesta di ricerca composta)

Specifica i nomi delle query e i criteri di ricerca per una richiesta di ricerca composta. Vedere "Comando query `-findquery` (Ricerca composta)" a pagina 49.

**Valore:** Un'espressione query.

**Obbligatorio con:** Comando query `-findquery`

La sintassi per una richiesta di ricerca composta è:

```
-query=<dichiarazioni-richiesta><definizioni-richiesta>&-findquery
```

dove:

`<dichiarazioni-richiesta>` è per due o più dichiarazioni di richiesta.

- Ogni dichiarazione di richiesta è costituita da uno o più identificatori di query separati da virgole e racchiusi tra parentesi. Un identificatore di query è la lettera "q" seguita da un numero. Ad esempio: `q1`
- Racchiuse tra parentesi, più query funzionano come ricerche AND logiche che limitano il gruppo trovato. Ad esempio, `(q1, q2)` restituisce i record che corrispondono a `q1` e `q2`.

**Nota** Si raccomanda di non utilizzare gli stessi campi per più variabili `q` nello stesso criterio di ricerca "and".

- Come per FileMaker Pro, ogni richiesta può essere una richiesta di ricerca oppure una richiesta di omissione. Una richiesta di ricerca aggiunge i record corrispondenti al gruppo trovato; una richiesta di omissione elimina i record corrispondenti dal gruppo trovato. L'impostazione predefinita è una richiesta di ricerca. Per una richiesta di omissione, inserire un punto esclamativo (!) davanti alla parentesi di apertura.

#### Esempio

```
(q1) ; ! (q2)
```

`q1` è una richiesta di ricerca; `q2` è una richiesta di omissione perché è preceduta da un punto esclamativo.



- Le richieste sono separate da punti e virgola. Più ricerche di richiesta funzionano come ricerche OR logiche che ampliano il gruppo trovato. Ad esempio, (q1) ; (q2) restituisce i record che corrispondono a q1 o q2. Le richieste di omissione non funzionano come ricerche logiche OR dato che le richieste di omissione rimuovono i record dal gruppo trovato.
- Le richieste vengono eseguite nell'ordine specificato; il gruppo trovato comprende i risultati di tutte le richieste di ricerca composte.

<definizioni-richiesta> è una definizione di richiesta per ogni dichiarazione di richiesta. Ogni definizione di richiesta è costituita da un campo di ricerca e da una definizione del valore. Il segno meno (-) avvia la definizione di richiesta.

Sintassi:

```
-<id-query>=<nomecampo>&-<id-query>.value=<valore>
```

### Esempi

```
-q1=typeofanimal&-q1.value=Cat
-q2=name&-q2.value=Fluffy
```

Ricerca i record per i gatti grigi che non hanno il nome "Fluffy":

```
http://host/fmi/xml/fmresultset.xml?-db=petclinic&-lay=Patients
&-query=(q1, q2);!(q3)&-q1=typeofanimal&-q1.value=Cat&-q2=color
&-q2.value=Gray&-q3=name&-q3.value=Fluffy&-findquery
```

### Parametro query -recid (ID record)

Specifica il record che si desidera elaborare. È utilizzato principalmente dai comandi query -edit e -delete. Utilizzato dal comando -view per recuperare i dati della lista valori correlata nella grammatica FMPXMLLAYOUT.

**Valore:** L'ID di un record, cioè l'indicatore univoco del record in un database FileMaker Pro

**Obbligatorio con:** Comandi query -edit, -delete e -dup

**Opzionale con:** Query -find e comandi -view

### Esempi

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-recid=22&-delete
http://localhost/fmi/xml/FMPXMLLAYOUT.xml?-db=test&-lay=empty&-view
&-recid=9
```

## Parametro query `-relatedsets.filter` (Filtra record portale)

Specifica se limitare i record del portale da restituire nei risultati di questa query.

**Valore:** `layout` o `none`

- Se `-relatedsets.filter` è impostato su `layout`, viene rispettata l'impostazione **Riga iniziale** specificata nella finestra di dialogo Impostazione portale di FileMaker Pro.
  - Se l'impostazione **Consenti scorrimento verticale** è selezionata nella finestra di dialogo Impostazione portale, utilizzare l'opzione `-relatedsets.max` per specificare il numero massimo di record da restituire. Vedere "Parametro query `-relatedsets.max` (Limita record portale)" di seguito.
  - Se l'impostazione **Consenti scorrimento verticale** non è selezionata o se non si utilizza l'opzione `-relatedsets.max`, l'impostazione **Numero di righe** nella finestra di dialogo Impostazione portale determina il numero di record del portale da restituire.
- Il valore predefinito è `none` se questo parametro non è indicato. Se `-relatedsets.filter` è impostato su `none`, il Motore per la Pubblicazione Web restituisce tutti i record del portale. I valori **Riga iniziale** e **Numero di righe** specificati nella finestra di dialogo Impostazione portale vengono ignorati.

### Note

- Il parametro `-relatedsets.filter` non influisce sull'ordinamento dei record del portale nelle query XML. L'ordinamento definito nel grafico delle relazioni viene rispettato se il valore del parametro `-relatedsets.filter` è `layout` o `none`.
- L'impostazione **Filtra record portale** nella finestra di dialogo Impostazione portale non è supportata per le query XML. I calcoli specificati per l'impostazione **Filtra record portale** sono ignorati.

**Opzionale con:** `-find`, `-edit`, `-new`, `-dup` e `-findquery`.

### Esempi

```
http://localhost/fmi/xml/fmresultset.xml?-db=FMPHP_Sample&-lay=English
&-relatedsets.filter=none&-findany
http://localhost/fmi/xml/fmresultset.xml?-db=FMPHP_Sample
&-lay=English&-relatedsets.filter=layout&-relatedsets.max=all&-findany
http://localhost/fmi/xml/fmresultset.xml?-db=FMPHP_Sample&-lay=English
&-relatedsets.filter=layout&-relatedsets.max=10&-findany
```

### Parametro query `-relatedsets.max` (Limita record portale)

Specifica il numero massimo di record portale da restituire nei risultati della query.

**Il valore è:** un intero o `all`.

- Il parametro `-relatedsets.max` viene rispettato solo se l'impostazione **Consenti scorrimento verticale** è abilitata nella finestra di dialogo Impostazione portale di FileMaker Pro e il parametro `-relatedsets.filter` è `layout`.
  - Se il parametro `-relatedsets.max` specifica un numero intero, il Motore per la Pubblicazione Web restituisce un numero di record portale pari a quel numero a partire dalla riga iniziale.
  - Se il parametro `-relatedsets.max` specifica `all`, il Motore per la Pubblicazione Web restituisce tutti i record del portale.

**Nota** Per informazioni sul filtro di record portale, vedere “Parametro query `-relatedsets.filter` (Filtra record portale)” sopra.

**Opzionale con:** `-find`, `-edit`, `-new`, `-dup` e `-findquery`.

#### Esempi

```
http://localhost/fmi/xml/fmresultset.xml?-db=FMPHP_Sample
&-lay=English&relatedsets.filter=layout&-relatedsets.max=all&-findany
http://localhost/fmi/xml/fmresultset.xml?-db=FMPHP_Sample&-lay=English
&-relatedsets.filter=layout&-relatedsets.max=10&-findany
```

### Parametro query `-script` (Script)

Specifica lo script FileMaker da eseguire dopo l'esecuzione del comando query e l'ordinamento. Vedere “Comprendere come una richiesta XML viene elaborata” a pagina 41.

**Valore:** Nome script

**Opzionale con:** Tutti i comandi query tranne `-dbnames`, `-layoutnames` e `-scriptnames`

#### Esempio

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-script=myscript&-findall
```

### Parametro query `-script.param` (Trasferisci parametro a script)

Trasferisce un parametro allo script FileMaker specificato da `-script`.

**Valore:** Un parametro di testo singolo.

- Per trasferire più parametri, è possibile creare una stringa che delimita i parametri e fare in modo che lo script analizzi il singolo parametro. Ad esempio, trasferire "param1|param2|param3 " come lista con il carattere "|" con codifica URL come questo: `param1%7Cparam2%7Cparam3`
- Per trattare il parametro di testo come valore che non sia testo, lo script può convertire il valore del testo. Ad esempio, per convertire il valore di testo in un numero, lo script potrebbe comprendere i seguenti elementi: `RicavaComeNumero ( Get ( ParametroScript ) )`
- Se la query contiene `-script.param` senza `-script`, `-script.param` viene ignorato.
- Se la query contiene più di un `-script.param`, il Motore per la Pubblicazione Web utilizza l'ultimo valore analizzato.

**Opzionale con:** `-script`

#### Esempio

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-script=myscript&-script.param=Smith%7CChatterjee%7CSu
&-findall
```

### Parametro query `-script.prefind` (Script prima di trovare)

Specifica lo script FileMaker da eseguire prima della ricerca e dell'ordinamento dei record (se specificati) durante l'elaborazione del comando query `-find`.

**Valore:** Nome script

**Opzionale con:** Tutti i comandi query tranne `-dbnames`, `-layoutnames` e `-scriptnames`

#### Esempio

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-script.prefind=myscript&-findall
```

### Parametro query `-script.prefind.param` (Trasferisci parametro a script prima di trovare)

Trasferisce un parametro allo script FileMaker specificato da `-script.prefind`.

**Valore:** Un parametro di testo singolo.

- Per trasferire più parametri, è possibile creare una stringa che delimita i parametri e fare in modo che lo script analizzi il singolo parametro. Ad esempio, trasferire "param1|param2|param3 " come lista con il carattere "|" con codifica URL come questo: `param1%7Cparam2%7Cparam3`
- Per trattare il parametro di testo come valore che non sia testo, lo script può convertire il valore del testo. Ad esempio, per convertire il valore di testo in un numero, lo script potrebbe comprendere i seguenti elementi: `RicavaComeNumero ( Get ( ParametroScript ) )`
- Se la query contiene `-script.prefind.param` senza `-script.prefind`, `-script.prefind.param` viene ignorato.

- Se la query contiene più di un `-script.prefind.param`, il Motore per la Pubblicazione Web utilizza l'ultimo valore analizzato.

**Opzionale con:** `-script.prefind`

#### Esempio

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-script.prefind=myscript&-script.prefind.param=payroll
&-findall
```

#### Parametro query `-script.presort` (Script prima di ordinare)

Specifica lo script FileMaker da eseguire dopo la ricerca dei record (se specificata) e prima dell'ordinamento dei record durante l'elaborazione del comando query `-find`.

**Opzionale con:** Tutti i comandi query tranne `-dbnames`, `-layoutnames` e `-scriptnames`

#### Esempio

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-script.presort=myscript&-sortfield.1=dept
&-sortfield.2=rating&-findall
```

#### Parametro query `-script.presort.param` (Trasferisci parametro a script prima di ordinare)

Trasferisce un parametro allo script FileMaker specificato da `-script.presort`.

**Valore:** Un parametro di testo singolo.

- Per trasferire più parametri, è possibile creare una stringa che delimita i parametri e fare in modo che lo script analizzi il singolo parametro. Ad esempio, trasferire "param1|param2|param3 " come lista con il carattere "|" con codifica URL come questo: `param1%7Cparam2%7Cparam3`
- Per trattare il parametro di testo come valore che non sia testo, lo script può convertire il valore del testo. Ad esempio, per convertire il valore di testo in un numero, lo script potrebbe comprendere i seguenti elementi: `RicavaComeNumero ( Get ( ParametroScript ) )`
- Se la query contiene `-script.presort.param` senza `-script.presort`, `-script.presort.param` viene ignorato.
- Se la query contiene più di un `-script.presort.param`, il Motore per la Pubblicazione Web utilizza l'ultimo valore analizzato.

**Opzionale con:** `-script.presort`

#### Esempio

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-script.presort=myscript&-script.presort.param=18%7C65
&-sortfield.1=dept&-sortfield.2=rating&-findall
```

### Parametro query `-skip` (Salta record)

Specifica il numero di record da saltare nel gruppo trovato.

**Valore:** Un numero. Se il valore è maggiore del numero di record nel gruppo trovato, non viene visualizzato nessun record. Il valore predefinito è 0.

**Opzionale con:** Comando query `-find`

#### Esempio

I primi 10 record nel gruppo trovato vengono saltati e vengono restituiti i record da 11 a 15:

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=departments&-skip=10&-max=5&-findall
```

### Parametro query `-sortfield` (Campo ordinamento)

Specifica il campo da utilizzare per l'ordinamento.

**Il valore è:** nome di campo

**Opzionale con:** Comandi query `-find` o `-findall`

Il parametro `-sortfield` può essere usato più volte per eseguire diversi ordinamenti. La sintassi per specificare la precedenza dei campi di ordinamento è:

`-sortfield.numero-precedenza=nome-campo-completamente-qualificato`

dove `numero-precedenza` nel parametro query `-sortfield.numero-precedenza` è un numero che specifica la precedenza da usare per più campi di ordinamento. Il valore per `numero-precedenza`:

- deve iniziare da 1
- deve incrementarsi sequenzialmente
- non deve essere maggiore di 9

#### Esempio

Il campo "dept" viene ordinato per primo, seguito dal campo "rating". Entrambi i campi vengono ordinati in ordine crescente perché il parametro query `-sortorder` non è specificato.

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=performance&-sortfield.1=dept&-sortfield.2=rating&-findall
```

## Parametro query `-sortorder` (Ordinamento)

Indica il criterio di un ordinamento.

**Valore:** Il criterio di ordinamento. Di seguito sono riportati i criteri di ordinamento validi, dove `<nome-lista-valori>` è il nome di una lista valori come ad esempio `Custom`:

Parola chiave	Operatore FileMaker Pro equivalente
<code>ascend</code>	Ordinamento da a a z, da -10 a 10
<code>descend</code>	Ordinamento da z a a, da 10 a -10
<code>&lt;nome-lista-valori&gt;</code>	L'ordinamento usa la lista valori associata al campo nel formato

**Opzionale con:** Comandi query `-find o -findall`

**Richiede:** Parametro query `-sortfield`

Il parametro query `-sortorder` può essere usato con il parametro `-sortfield` per specificare il criterio di ordinamento di più campi di ordinamento. La sintassi per specificare il criterio di ordinamento di un campo è:

`-sortorder.numero-precedenza=metodo-ordinamento`

dove:

- `numero-precedenza` nel parametro `-sortorder.numero-precedenza` è un numero compreso tra 1 e 9 che specifica il parametro query `-sortfield` a cui è applicato il parametro query `-sortorder`
- `metodo-ordinamento` è una delle parole chiave nella tabella precedente per specificare il criterio di ordinamento, come ad esempio `ascend`

### Esempio

Il criterio di ordinamento del campo di ordinamento con la precedenza più alta (`dept`) è crescente e il criterio di ordinamento del campo di ordinamento con la seconda precedenza (`rating`) è decrescente. Il `numero-precedenza 2` in `-sortorder.2` indica che il parametro query `-sortorder.2=descend` viene applicato al parametro query `-sortfield.2=rating`.

```
http://192.168.123.101/fmi/xml/fmresultset.xml?-db=employees
&-lay=performance&-sortfield.1=dept&-sortorder.1=ascend
&-sortfield.2=rating&-sortorder.2=descend&-findall
```

**Nota** Se per un campo di ordinamento non è specificato un parametro query `-sortorder`, viene usato il criterio di ordinamento crescente predefinito.

# Capitolo 6

## Pubblicazione Web personalizzata con PHP

La Pubblicazione Web personalizzata con PHP consente di utilizzare il linguaggio di scripting PHP per integrare i dati dei database FileMaker con i formati delle pagine Web personalizzate. La Pubblicazione Web personalizzata con PHP fornisce FileMaker API per PHP che è una classe PHP creata da FileMaker che consente di accedere ai database ospitati da FileMaker Server. Questa classe PHP si collega al Motore per la Pubblicazione Web di FileMaker Server e rende disponibili i dati sul motore PHP del server Web.

### Caratteristiche principali della Pubblicazione Web personalizzata con PHP

- Creare le applicazioni Web che utilizzano il linguaggio di scripting open source PHP. Utilizzare la versione di PHP supportata da FileMaker Server o utilizzare la propria versione di PHP. (Se si sceglie di utilizzare la propria versione di PHP, vedere “Installazione manuale di FileMaker API per PHP” a pagina 66.)
- Ospitare i database su FileMaker Server. FileMaker Pro non è necessario per la Pubblicazione Web personalizzata perché FileMaker Server ospita i database.
- Scrivere il codice PHP che consente di creare, eliminare, modificare e duplicare i record in un database FileMaker ospitato. Il codice può eseguire la convalida dei campi e dei record prima di salvare le modifiche nel database ospitato.
- Scrivere il codice PHP con cui si accede ai formati, ai portali, alle liste valori e ai campi correlati. Come in FileMaker Pro, l’accesso ai dati, ai formati e ai campi dipende dalle impostazioni degli account utente definite nei privilegi di accesso del database. Per il Motore per la Pubblicazione Web sono stati inoltre introdotti altri miglioramenti per la sicurezza. Vedere “Protezione dei database pubblicati” a pagina 14.
- Scrivere il codice PHP con cui vengono eseguiti script complessi, a più istruzioni. FileMaker supporta molte istruzioni di script nella Pubblicazione Web personalizzata. Vedere “Script FileMaker e Pubblicazione Web personalizzata” a pagina 18.
- Scrivere il codice PHP con cui vengono eseguiti script complessi.

### Requisiti della Pubblicazione Web personalizzata

Questa sezione spiega quali sono i requisiti per sviluppare una soluzione di Pubblicazione Web personalizzata utilizzando PHP, ciò di cui gli utenti Web hanno bisogno per accedere alla soluzione di Pubblicazione Web personalizzata e l’impatto che può avere l’hosting di una soluzione di pubblicazione Web sul server.

#### Requisiti per la pubblicazione di un database mediante Pubblicazione Web personalizzata

Per pubblicare i database utilizzando la Pubblicazione Web personalizzata con PHP, è necessario:

- Una distribuzione di FileMaker Server che comprende tre componenti:
  - Un server Web, Microsoft IIS (Windows) o Apache (macOS) — Il Modulo Server Web di FileMaker è installato sul server Web;
  - Il Motore per la Pubblicazione Web di FileMaker;
  - Il server Database FileMaker;



- Uno o più database FileMaker Pro ospitati da FileMaker Server;
- Indirizzo IP o nome di dominio dell'host del server Web;
- Un browser Web e la possibilità di accedere al server Web per sviluppare e controllare la soluzione di Pubblicazione Web personalizzata;
- PHP installato sul server Web — FileMaker Server può installare la versione di PHP supportata o si può utilizzare la propria versione;
  - Per la versione minima richiesta di PHP, vedere le [specifiche tecniche di FileMaker Server](#);
  - Per informazioni su PHP, visitare il sito [php.net](http://php.net);
  - La versione di PHP installata sul server Web deve supportare le funzioni cURL (client URL library). Per informazioni su cURL, visitare il sito [php.net/curl](http://php.net/curl).

**Importante** Se si installa la versione di PHP supportata da FileMaker Server, questa non viene visualizzata nello strumento Server Admin di macOS; presumibilmente non è nell'elenco. Se si utilizza lo strumento Server Admin di macOS per attivare PHP, viene disattivata la versione di PHP supportata da FileMaker Server e attivata la propria versione di PHP.

Vedere la [Guida all'installazione e alla configurazione di FileMaker Server](#).

### Cosa è richiesto agli utenti Web per accedere a una soluzione di Pubblicazione Web personalizzata

Per accedere alla soluzione di Pubblicazione Web personalizzata che fa uso di PHP, gli utenti Web devono disporre di:

- un browser Web
- accesso a Internet o a una rete intranet e al server Web
- indirizzo IP o nome di dominio dell'host del server Web.

Se il database è protetto da password, gli utenti Web devono anche introdurre un nome utente e una password per l'account database.

### Connessione a Internet o a una rete intranet

Quando si pubblicano database su Internet o su una rete intranet, è necessario che sul computer host sia in esecuzione FileMaker Server e i database da condividere devono essere ospitati e disponibili. Inoltre:

- Pubblicare il database su un computer con una connessione continua a Internet o a un'intranet. È possibile pubblicare database anche in assenza di una connessione continua, ma in tal caso i database risulteranno disponibili agli utenti solo quando il computer che li ospita è connesso a Internet o alla rete intranet.
- Il computer host del server Web, che fa parte della distribuzione di FileMaker Server, deve avere un nome di dominio o un indirizzo IP statico (permanente) dedicato. Se ci si collega a Internet con un provider di servizi Internet (ISP), l'indirizzo IP potrebbe essere assegnato in modo dinamico (ossia essere diverso ad ogni connessione). L'utilizzo di un indirizzo IP dinamico rende più difficoltosa l'individuazione dei database da parte degli utenti. Se non si è certi del tipo di accesso di cui si dispone, contattare il provider di servizi Internet o l'amministratore della rete.

## Installazione manuale di FileMaker API per PHP

Se si installa FileMaker Server, viene offerta l'opzione di installare la versione di PHP supportata da FileMaker. Se il motore PHP è già stato installato e configurato e si desidera aggiungere solo FileMaker API per PHP, installare manualmente la classe FileMaker API per PHP per renderla disponibile agli script PHP.

Se non viene installata la versione di PHP supportata da FileMaker, assicurarsi di eseguire le seguenti operazioni di configurazione sulla propria versione del motore PHP:

- Attivare il modulo cURL in php.ini.
- Selezionare la posizione di FileMaker API per PHP nella variabile `include_path` in php.ini.
- Se si accede a database che contengono date e ore, installare il [pacchetto pear date](#).

**Nota** Per la versione minima richiesta di PHP, vedere le [specifiche tecniche di FileMaker Server](#). Per ottenere i migliori risultati, utilizzare la versione di PHP appropriata.

### Per rendere accessibile FileMaker API per PHP agli script PHP

Se è stato installato FileMaker Server, il pacchetto FileMaker API per PHP è compreso sotto forma di file `.zip` nella seguente posizione:

- Per IIS (Windows):

```
[unità]:\Programmi\FileMaker\FileMaker Server\Web  
Publishing\FM_API_for_PHP_Standalone.zip
```

dove [unità] è l'unità in cui risiede il componente server Web della distribuzione di FileMaker Server.

- Per Apache (macOS):

```
/Libreria/FileMaker Server/Web  
Publishing/FM_API_for_PHP_Standalone.zip
```

Il file `FM_API_for_PHP_Standalone.zip` contiene un file chiamato `FileMaker.php` e una cartella chiamata `FileMaker`. Dezippare il file e copiare il file `FileMaker.php` e la cartella `FileMaker` in una di queste posizioni:

- la cartella in cui risiedono gli script PHP.

- Per IIS (Windows) con HTTP o HTTPS:

```
[unità]:\Programmi\FileMaker\FileMaker Server\HTTPServer\Conf  
dove [unità] è l'unità in cui risiede il componente Motore per la Pubblicazione Web della  
distribuzione di FileMaker Server.
```

- Per Apache (macOS) con HTTP:

```
/Libreria/FileMaker Server/HTTPServer/htdocs
```

- Per Apache (macOS) con HTTPS:

```
/Libreria/FileMaker Server/HTTPServer/htdocs/httpsRoot
```

**Nota** Se si seleziona **Usare HSTS per client Web** nell'Admin Console, utilizzare la directory `HTTPS` per ospitare i file del sito PHP.

- una delle directory `include_path` dell'installazione PHP. La posizione predefinita per macOS è `/usr/lib/php`.

## Operazioni successive

- Utilizzare Admin Console di FileMaker Server per attivare la Pubblicazione Web personalizzata. Vedere la [Guida di FileMaker Server](#) e la [Guida all'installazione e alla configurazione di FileMaker Server](#).
- In FileMaker Pro, aprire il database FileMaker che si desidera pubblicare e assicurarsi che il database abbia i privilegi estesi appropriati attivati per la Pubblicazione Web personalizzata. Vedere “Attivazione della Pubblicazione Web personalizzata in un database” a pagina 13.
- Per sapere come accedere ai dati nei database FileMaker utilizzando FileMaker API per PHP, vedere capitolo 8, “Utilizzo di FileMaker API per PHP.”

# Capitolo 7

## Panoramica della Pubblicazione Web personalizzata con PHP

FileMaker API per PHP è un supporto per l'integrazione dei dati dei database FileMaker Pro in soluzioni PHP. Questo capitolo descrive il modo in cui PHP lavora con il Motore per la Pubblicazione Web personalizzata di FileMaker Server. Per ulteriori informazioni su FileMaker API per PHP, vedere capitolo 8, "Utilizzo di FileMaker API per PHP."

### Utilizzo del Motore per la Pubblicazione Web con soluzioni PHP

FileMaker Server è costituito da tre componenti: un server Web, il Motore per la Pubblicazione Web e il server database. Vedere la [Guida all'installazione e alla configurazione di FileMaker Server](#). Per supportare le soluzioni PHP, con il server Web sulla macchina master viene installato un motore PHP. FileMaker Server ospita una soluzione PHP quando i file PHP vengono posizionati sul server Web della macchina master.

- Quando un utente Web apre una soluzione PHP, il server Web invia la richiesta al motore PHP che elabora il codice PHP.
- Se il codice PHP contiene dei richiami a FileMaker API per PHP, questi vengono interpretati e inviati come richieste al Motore per la Pubblicazione Web.
- Il Motore per la Pubblicazione Web richiede i dati ai database ospitati sul server Database.
- Il server Database invia i dati richiesti al Motore per la Pubblicazione Web.
- Il Motore per la Pubblicazione Web invia i dati al motore PHP sul server Web in risposta alla chiamata API.
- La soluzione PHP elabora i dati e li visualizza per l'utente Web.

### Operazioni generali per la Pubblicazione Web personalizzata con PHP

1. Nell'Admin Console, assicurarsi che **Attiva Pubblicazione PHP** sia selezionato. Vedere la [Guida all'installazione e alla configurazione di FileMaker Server](#).
2. Nell'Admin Console, selezionare il riquadro **Database** e assicurarsi che ogni database FileMaker che si sta pubblicando abbia il privilegio esteso fmphp attivato per la Pubblicazione Web personalizzata con PHP.

Se necessario, utilizzare FileMaker Pro per attivare la Pubblicazione Web personalizzata per un database. Vedere capitolo 2, "Preparazione dei database per la Pubblicazione Web personalizzata."

**Nota** Assicurarsi di utilizzare set di privilegi del database FileMaker equivalenti quando si sviluppano soluzioni PHP che verranno consegnate all'utente finale. In caso contrario, si potrebbe avere l'accesso a formati e funzioni nel database FileMaker che non sarà disponibile all'utente finale, provocando un comportamento incoerente.

3. Utilizzare gli strumenti di authoring PHP per creare la propria soluzione PHP, incorporando le funzioni FileMaker API nel codice PHP per accedere ai dati FileMaker. Vedere capitolo 8, "Utilizzo di FileMaker API per PHP."

4. Copiare o spostare la struttura di directory del sito e i file nella seguente cartella sul server Web della macchina master.

- Per IIS (Windows) con HTTP o HTTPS:  
[unità] : \Programmi\FileMaker\FileMaker Server\HTTPServer\Conf  
dove [unità] è l'unità sulla macchina master in cui risiede la distribuzione di FileMaker Server.
- Per Apache (macOS) con HTTP:  
/Libreria/FileMaker Server/HTTPServer/htdocs
- Per Apache (macOS) con HTTPS:  
/Libreria/FileMaker Server/HTTPServer/htdocs/httpsRoot

**Nota** Utilizzare la directory HTTPS per ospitare i file del sito PHP quando **Usare HSTS per client Web** è selezionato nell'Admin Console. Se **Usare HSTS per client Web** è selezionato, quando un client Web si collega a FileMaker Server con HTTPS, il browser Web impedisce al client di utilizzare una connessione HTTP per tutti i contenuti Web ospitati da FileMaker Server.

5. Se un campo Contenitore del database contiene un riferimento al file anziché il file reale, l'oggetto Contenitore di riferimento deve essere memorizzato nella cartella Web di FileMaker Pro quando il record viene creato o modificato. Copiare o spostare l'oggetto in una cartella con la stessa posizione relativa nella cartella principale del software del server Web.

Vedere "Pubblicazione del contenuto dei campi Contenitore sul Web" a pagina 15.

6. Assicurarsi che i meccanismi della sicurezza per il sito o per il programma siano a posto.

7. Controllare il sito utilizzando gli stessi account e gli stessi privilegi definiti per gli utenti Web.

8. Rendere il sito disponibile e noto agli utenti. L'URL che l'utente Web inserisce ha il seguente formato:

```
http://<server>/<percorso_sito>
```

- <server> è la macchina in cui risiede FileMaker Server.
- <percorso\_sito> è il percorso relativo alla home page del sito, determinato dalla struttura di directory utilizzata al passo 4 sopra.

### Esempio

Se il server Web è 192.168.123.101 e la home page del sito si trova sul server Web nel percorso c:\inetpub\wwwroot\customers\index.php, l'utente Web deve inserire l'URL:

```
http://192.168.123.101/customers/index.php
```

**Nota** PHP utilizza la codifica Latin-1 (ISO-8859-1). FileMaker Server restituisce dati Unicode (UTF-8). Utilizzare Admin Console di FileMaker Server per indicare la codifica caratteri predefinita per il sito. Per i siti PHP, è possibile indicare UTF-8 o ISO-8859-1; si raccomanda di impostare UTF-8. Indicare la stessa impostazione per l'attributo `charset` nella sezione `<HEAD>` dei file PHP del sito.

Per informazioni sulla distribuzione e sull'utilizzo di una soluzione PHP, vedere capitolo 9, "Preparazione, test e monitoraggio di un sito."

# Capitolo 8

## Utilizzo di FileMaker API per PHP

FileMaker API per PHP implementa una classe PHP—la classe FileMaker—che fornisce un'interfaccia orientata agli oggetti per i database FileMaker. FileMaker API per PHP consente di accedere ai dati e alla logica memorizzati nei database FileMaker Pro e di pubblicarli sul Web o di esportarli in altre applicazioni.

FileMaker API per PHP permette al codice PHP di eseguire lo stesso tipo di funzioni già disponibili nei database FileMaker Pro:

- creare, eliminare, modificare e duplicare i record
- eseguire richieste di ricerca
- convalidare campi e record
- utilizzare formati
- eseguire script FileMaker
- visualizzare portali e record correlati
- utilizzare liste valori

Questo capitolo descrive come utilizzare gli oggetti classe FileMaker e i metodi per aggiungere queste funzioni comuni a una soluzione PHP. Questo capitolo non tratta tutti gli argomenti relativi a FileMaker API per PHP, ma introduce le nozioni ed i metodi più importanti.

### Dove ottenere informazioni aggiuntive

Per ulteriori informazioni su FileMaker API per PHP, consultare le seguenti risorse.

Se il motore PHP è già stato installato e configurato e si desidera aggiungere solo FileMaker API per PHP, vedere “Installazione manuale di FileMaker API per PHP” a pagina 66.

### Informazioni di riferimento a FileMaker API per PHP

Se si è installato FileMaker API per PHP, è possibile trovare informazioni di riferimento sul componente server Web della distribuzione di FileMaker Server.

- Per IIS (Windows):  
[unità]:\Programmi\FileMaker\FileMaker Server\Documentation\PHP API Documentation\index.html  
dove [unità] è l'unità in cui risiede il componente server Web della distribuzione di FileMaker Server.
- Per Apache (macOS): /Libreria/FileMaker Server/Documentation/PHP API Documentation/index.html

### Supporto FileMaker API per PHP

Per ulteriori informazioni su FileMaker API per PHP, visitare la pagina del [supporto](#) di FileMaker.

## Utilizzo della classe FileMaker

Per utilizzare la classe FileMaker nella soluzione PHP, aggiungere al codice PHP la seguente istruzione:

```
require_once ('FileMaker.php');
```

### Oggetti classe FileMaker

La classe FileMaker definisce gli oggetti classe utilizzati per recuperare i dati dai database FileMaker Pro.

Oggetto classe	Utilizzare l'oggetto per
Database FileMaker	Definire le proprietà del database Collegarsi a un database FileMaker Pro Ottenere informazioni su FileMaker API per PHP
Comando	Creare comandi per aggiungere, eliminare, duplicare e modificare i record, eseguire richieste di ricerca ed eseguire script
Formato	Lavorare con i formati di database
Record	Lavorare con i dati dei record
Campo	Lavorare con i dati dei campi
Set correlato	Lavorare con i record del portale
Risultato	Elaborare i record restituiti da una richiesta di ricerca
Errore	Controllare se si è verificato un errore Elaborare eventuali errori

### Oggetti comando FileMaker

La classe FileMaker definisce un oggetto comando base utilizzato per creare l'istanza di un comando specifico e indicare i parametri del comando. Per eseguire il comando, richiamare il metodo `execute()`.

La classe FileMaker definisce i seguenti comandi specifici:

- Comando Add
- Comando Compound Find
- Comando Delete
- Comando Duplicate
- Comando Edit
- Comando Find, comando Find All, comando Find Any
- Comando Find Request, che ottiene delle aggiunte al comando di ricerche composte
- Comando Esegui script

**Importante** I comandi restituiscono valori differenti, come definito dalla classe `FileMaker.php`. Ad esempio, alcuni comandi restituiscono il valore booleano VERO o un oggetto `FileMaker_Error`. Altri comandi restituiscono un oggetto `FileMaker_Result` che può contenere un intero “gruppo trovato” di record in un formato. Per evitare problemi di sovraccarico della memoria del computer, considerare i valori restituiti previsti dai comandi utilizzati. Per informazioni dettagliate sui valori restituiti da ciascun comando, vedere “Informazioni di riferimento a FileMaker API per PHP”.

Le operazioni base che la maggior parte delle applicazioni PHP devono eseguire sono descritte nelle seguenti sezioni:

- “Utilizzo dei record” a pagina 73
- “Esecuzione di script FileMaker” a pagina 75
- “Esecuzione di richieste di ricerca” a pagina 81

### Decodifica dei dati per l'uso in FileMaker API

Se l'applicazione PHP recupera i dati da un sito Web, questi potrebbero avere una codifica URL. FileMaker API per PHP si aspetta che i dati siano stringhe decodificate, non stringhe con codifica URL. Come prassi generale, si potrebbe voler richiamare la funzione `urldecode()` quando si recuperano i dati nell'applicazione PHP.

#### Esempio

```
$user = urldecode($_GET['user']);  
$event = urldecode($_GET['event']);
```

**Nota** Evitare di utilizzare stringhe contenenti e commerciali (&) con FileMaker API per PHP. Utilizzare una barra rovesciata come carattere di escape prima dei caratteri speciali nelle stringhe passate a FileMaker API per PHP.

## Collegamento a un database FileMaker

La classe FileMaker definisce un oggetto del database sul quale viene creata un'istanza per collegarsi a un server o a un database. Definire le proprietà dell'oggetto con il costruttore della classe o richiamando il metodo `setProperty()`.

#### Esempi

Collegamento a un server per ricavare un elenco di database:

```
$u$fm = new FileMaker();  
$databases = $fm->listDatabases();
```

Collegamento a un database specifico su un server:

```
$fm = new FileMaker();  
$fm->setProperty('database', 'questionnaire');  
$fm->setProperty('hostspec', 'http://192.168.100.110');  
$fm->setProperty('username', 'web');  
$fm->setProperty('password', 'web');
```

Le proprietà di `username` e `password` determinano il set di privilegi per la connessione.

**Nota** La proprietà `hostspec` è preimpostata sul valore `http://localhost`. Poiché il motore PHP è installato con il componente server Web della macchina master, non è necessario specificare la proprietà `hostspec`.



## Utilizzo dei record

La classe FileMaker definisce un oggetto record sul quale viene creata un'istanza per utilizzare i record. L'istanza di un oggetto record rappresenta un record di un database FileMaker Pro. Utilizzare un oggetto record con i comandi Add, Delete, Duplicate e Edit per modificare i dati del record. I comandi di ricerca —Find, Find All, Find Any e Compound Find—restituiscono una matrice di oggetti record.

### Creazione di un record

Esistono due modi per creare un record:

- Utilizzare il metodo `createRecord()`, specificando un nome formato e, opzionalmente, una matrice dei valori di campo. È inoltre possibile impostare i valori singolarmente nel nuovo oggetto record.

Il metodo `createRecord()` non consente di salvare il nuovo record nel database. Per salvare il record nel database, utilizzare il metodo `commit()`.

#### Esempio

```
$rec = $fm->createRecord('Form View', $values);  
$result = $rec->commit();
```

Se si utilizza il metodo `commit()` FileMaker\_Record, alla variabile `$result` viene assegnato il valore booleano VERO e, se non si verificano errori, il nuovo record viene creato nel database FileMaker.

Se si verifica un errore, la variabile `$result` contiene un oggetto FileMaker\_Error. Dopo aver eseguito il metodo `commit()`, eseguire una verifica degli errori.

- Utilizzare il comando Add. Utilizzare il metodo `newAddCommand()` per creare un oggetto FileMaker\_Command\_Add, indicando il nome del formato e una matrice con i dati del record. Per salvare il record nel database, utilizzare il metodo `execute()`.

#### Esempio

```
$newAdd = $fm->newAddCommand('Respondent', $respondent_data);  
$result = $newAdd->execute();
```

Se si utilizza il metodo `execute()` FileMaker\_Command, se non si verificano errori, la variabile `$result` contiene un oggetto FileMaker\_Result con tutte le informazioni sul record creato.

Se si verifica un errore, la variabile `$result` contiene un oggetto FileMaker\_Error. Dopo aver eseguito il metodo `execute()`, eseguire una verifica degli errori.

### Duplicazione di un record

Duplicare un record esistente utilizzando il comando Duplicate. Utilizzare il metodo `newDuplicateCommand()` per creare un oggetto FileMaker\_Command\_Duplicate, indicando il nome del formato e l'ID del record da duplicare. Quindi duplicare il record utilizzando il metodo `execute()`.

### Esempio

```
$newDuplicate = $fm->newDuplicateCommand('Respondent', $rec_ID);  
$result = $newDuplicate->execute();
```

### Modifica di un record

Esistono due modi per modificare un record:

- Utilizzando il comando Edit. Utilizzare il metodo `newEditCommand()` per creare un oggetto `FileMaker_Command_Edit`, indicando il nome del formato, l'ID del record da modificare e una matrice di valori da aggiornare. Quindi modificare il record utilizzando il metodo `execute()`.

### Esempio

```
$newEdit = $fm->newEditCommand('Respondent', $rec_ID, $respondent_data);  
$result = $newEdit->execute();
```

- Utilizzando un oggetto record. Recuperare un record dal database, cambiare i valori campo e modificare il record utilizzando il metodo `commit()`.

### Esempio

```
$rec = $fm->getRecordById('Form View', $rec_ID);  
$rec->setField('Name', $nameEntered);  
$result = $rec->commit();
```

### Eliminazione di un record

Esistono due modi per eliminare un record:

- Recuperare un record dal database e utilizzare il metodo `delete()`.

### Esempio

```
$rec = $fm->getRecordById('Form View', $rec_ID);  
$rec->delete();
```

- Eliminare un record esistente utilizzando il comando Delete. Utilizzare il metodo `newDeleteCommand()` per creare un oggetto `FileMaker_Command_Delete`, indicando il nome del formato e l'ID del record da eliminare. Quindi eliminare il record utilizzando il metodo `execute()`.

### Esempio

```
$newDelete = $fm->newDeleteCommand('Respondent', $rec_ID);  
$result = $newDelete->execute();
```

## Esecuzione di script FileMaker

Uno script FileMaker è un set denominato di istruzioni di script. La classe FileMaker definisce vari metodi che consentono di utilizzare gli script FileMaker definiti in un database FileMaker Pro. Per informazioni sulle istruzioni di script compatibili con il Web (istruzioni di script che possono essere eseguite in una soluzione Web), vedere “Script FileMaker e Pubblicazione Web personalizzata” a pagina 18.

### Recupero dell’elenco degli script disponibili

Utilizzare il metodo `listScripts()` per ottenere una lista degli script disponibili dal database al momento collegato. Il metodo `listScripts()` restituisce una matrice che può essere eseguita con il nome utente e la password indicati quando è stato definito il collegamento con il database. (Vedere “Collegamento a un database FileMaker” a pagina 72.)

#### Esempio

```
$scripts = $fm->listScripts();
```

### Esecuzione di uno script FileMaker

Utilizzare il metodo `newPerformScriptCommand()` per creare un oggetto FileMaker\_Command\_PerformScript, specificando il formato, il nome dello script ed eventuali parametri di script. Quindi, eseguire lo script utilizzando il metodo `execute()`.

**Importante** Quando si esegue uno script FileMaker, le dimensioni dell’oggetto FileMaker\_Result restituito variano in base al comportamento dello script FileMaker. Ad esempio, se uno script FileMaker passa a un formato specificato, il gruppo trovato potrebbe contenere tutti i record della tabella di quel formato, e l’oggetto FileMaker\_Result potrebbe restituire tutti i record di quel gruppo trovato. Per evitare problemi di sovraccarico della memoria del computer, considerare i dati restituiti da uno script FileMaker prima di eseguirlo in un’applicazione PHP.

#### Esempio

```
$newPerformScript = $fm->newPerformScriptCommand('Order Summary',  
'ComputeTotal');  
$result = $newPerformScript->execute();
```

### Esecuzione di uno script prima di eseguire un comando

Utilizzare il metodo `setPreCommandScript()` per indicare uno script che viene eseguito prima che venga eseguito un comando. Nell’esempio seguente viene utilizzato un comando Find, ma può essere utilizzato il metodo `setPreCommandScript()` con un comando qualsiasi.

#### Esempio

```
$findCommand = $fm->newFindCommand('Students');  
$findCommand->addFindCriterion('GPA', $searchValue);  
$findCommand->setPreCommandScript('UpdateGPA');  
$result = $findCommand->execute();
```

## Esecuzione di uno script prima di ordinare un gruppo di risultati

Utilizzare il metodo `setPreSortScript()` per specificare uno script che viene eseguito dopo che è stato generato un gruppo di risultati Find, ma prima che tale gruppo venga ordinato. Vedere “Utilizzo del comando Find” a pagina 82.

### Esempio

```
$findCommand = $fm->newFindCommand('Students');  
$findCommand->setPreSortScript('RemoveExpelled');
```

## Esecuzione di uno script dopo la creazione di un gruppo di risultati

Utilizzare il metodo `setScript()` per specificare uno script eseguito dopo la creazione di un gruppo di risultati Find. Vedere “Utilizzo del comando Find” a pagina 82.

### Esempio

```
$findCommand = $fm->newFindCommand('Students');  
$findCommand->setScript('myScript', 'param1|param2|param3');
```

## Ordine di esecuzione degli script

Per un singolo comando si possono indicare i metodi `setPreCommandScript()`, `setPreSortScript()` e `setScript()` insieme ai metodi `setResultLayout()` e `addSortRule()`.

Di seguito è indicato l'ordine in cui FileMaker Server e il Motore per la Pubblicazione Web elaborano questi metodi:

1. Eseguire lo script indicato con il metodo `setPreCommandScript()`, se specificato.
2. Elaborare il comando come un comando Find o Delete Record.
3. Eseguire lo script indicato con il metodo `setPreSortScript()`, se specificato.
4. Ordinare il gruppo di risultati Find, se è stato indicato il metodo `addSortRule()`.
5. Elaborare il metodo `setResultLayout()` per passare ad un formato diverso, se specificato.
6. Eseguire lo script indicato con il metodo `setScript()`, se specificato.
7. Restituire il gruppo di risultati Find finale.

Se in uno dei passi indicati in precedenza viene generato un codice di errore, l'esecuzione del comando viene arrestata; eventuali passi successivi non vengono eseguiti. Vengono tuttavia eseguite tutte le fasi precedenti della richiesta.

Ad esempio, considerare un comando che elimina il record corrente, ordina i record e poi esegue uno script. Se il metodo `addSortRule()` specifica un campo inesistente, la richiesta elimina il record corrente e restituisce il codice di errore 102 (“Manca un campo”), ma non esegue lo script.

Il formato specificato per il metodo `newFindCommand()` viene utilizzato durante l'elaborazione della richiesta di ricerca. Quando il metodo `setResultLayout()` passa a un altro formato, l'oggetto errore per la richiesta di ricerca basata sul formato originale non è più disponibile. Per testare l'oggetto errore dalla richiesta di ricerca basata sul formato originale, controllare l'oggetto errore prima di cambiare il formato.

### Esempio

```

request = $fm->newFindCommand('Students');
$request->addFindCriterion('Day', 'Wednesday');

// Perform the Find
$result = $request->execute();

if (FileMaker::isError($result)) {
    if ($result->code = 401) {
        $findError = 'There are no Records that match that request: ' . ' (' .
            $result->code . ')';
    } else {
        $findError = 'Find Error: ' . $result->getMessage() . ' (' . $result->code
            . ')';
    }
}
$request->setResultLayout('Teachers');
// Switch to the result layout
$result = $request->execute();

```

## Utilizzo dei formati FileMaker

Un formato è un insieme di campi, oggetti, immagini e sezioni che rappresenta il modo in cui le informazioni vengono organizzate e presentate durante l'utilizzo, la visualizzazione in anteprima e la stampa dei record. La classe FileMaker definisce vari metodi che consentono di utilizzare i formati definiti in un database FileMaker Pro. Molti oggetti classe FileMaker consentono di ottenere informazioni sui formati.

Con questo oggetto classe	Utilizzare questi metodi
Database	<ul style="list-style-type: none"> <li>▪ <code>listLayouts()</code> consente di ottenere una lista di nomi formato disponibili.</li> <li>▪ <code>getLayout()</code> consente di ottenere un oggetto del formato indicando un nome formato.</li> </ul>
Formato	<ul style="list-style-type: none"> <li>▪ <code>getName()</code> recupera il nome formato di un determinato oggetto del formato.</li> <li>▪ <code>listFields()</code> recupera una matrice di tutti i nomi dei campi utilizzati in un formato.</li> <li>▪ <code>getFields()</code> recupera una matrice associativa con i nomi di tutti i campi come chiave e gli oggetti FileMaker_Field associati come valori di matrice.</li> <li>▪ <code>listValueLists()</code> recupera una matrice di nomi lista valori.</li> <li>▪ <code>listRelatedSets()</code> recupera una matrice dei nomi dei set correlati.</li> <li>▪ <code>getDatabase()</code> restituisce il nome del database.</li> </ul>
Record	<ul style="list-style-type: none"> <li>▪ <code>getLayout()</code> restituisce l'oggetto del formato associato ad un record specifico.</li> </ul>
Campo	<ul style="list-style-type: none"> <li>▪ <code>getLayout()</code> restituisce l'oggetto del formato contenente un campo specifico.</li> </ul>
Comando	<ul style="list-style-type: none"> <li>▪ <code>setResultLayout()</code> restituisce i risultati del comando in un formato diverso da quello corrente.</li> </ul>

## Utilizzo dei portali

Un portale è una tabella che visualizza righe di dati di uno o più record correlati. La classe FileMaker definisce un oggetto set correlato e diversi metodi che consentono di utilizzare i portali definiti in un database FileMaker Pro.

Un oggetto set correlato è una matrice di oggetti record del portale a cui si fa riferimento; ogni oggetto record rappresenta una riga di dati del portale.

### Elenco dei portali definiti in un formato specifico

Per un determinato oggetto del formato, utilizzare il metodo `listRelatedSets()` per recuperare una lista dei nomi tabella per tutti i portali definiti in questo formato.

#### Esempio

```
$tableNames = $currentLayout->listRelatedSets();
```

### Recupero dei nomi dei portali per un oggetto risultato specifico

Per un determinato oggetto FileMaker\_Result, usare il metodo `getRelatedSets()` per recuperare i nomi di tutti i portali di questo record.

#### Esempio

```
$relatedSetsNames = $result->getRelatedSets();
```

### Recupero delle informazioni sui portali per un formato specifico

Per un determinato oggetto del formato, utilizzare il metodo `getRelatedSets()` per recuperare una matrice di oggetti FileMaker\_RelatedSet che descrivono i portali del formato. La matrice restituita è una matrice associativa con i nomi tabella come chiavi di matrice e gli oggetti associati FileMaker\_RelatedSet come valori di matrice.

#### Esempio

```
$relatedSetsArray = $currentLayout->getRelatedSets();
```

### Recupero delle informazioni per un portale specifico

Per un determinato oggetto del formato, utilizzare il metodo `getRelatedSet()` per recuperare l'oggetto FileMaker\_RelatedSet che descrive uno specifico portale.

#### Esempio

```
$relatedSet = $currentLayout->getRelatedSet('customers');
```

## Recupero del nome della tabella per un portale

Per un oggetto set correlato, utilizzare il metodo `getName()` per ottenere il nome tabella del portale.

### Esempio

```
$tableName = $relatedSet->getName();
```

## Recupero dei record del portale per un record specifico

Per un determinato oggetto record, utilizzare il metodo `getRelatedSet()` per recuperare una matrice di record correlati per un portale specifico su tale record.

### Esempio

```
$relatedRecordsArray = $currentRecord->getRelatedSet('customers');
```

## Creazione di un nuovo record in un portale

Utilizzare il metodo `newRelatedRecord()` per creare un nuovo record nel set correlato indicato e salvare le modifiche nel database utilizzando il metodo `commit()`.

### Esempio

```
//create a new portal row in the 'customer' portal
$new_row = $currentRecord->newRelatedRecord('customer');

//set the field values in the new portal row
$new_row->setField('customer::name', $newName);
$new_row->setField('customer::company', $newCompany);

$result = $new_row->commit();
```

## Eliminazione di un record da un portale

Utilizzare il metodo `delete()` per eliminare un record di un portale.

### Esempio

```
$relatedSet = $currentRecord->getRelatedSet('customers');
/* Runs through each of the portal rows */
foreach ($relatedSet as $nextRow) {
    $nameField = $nextRow->getField('customer::name')
    if ($nameField == $badName) {
        $result = $newRow->delete();
    }
}
```

## Utilizzo delle liste valori

Una lista valori è un set di scelte predefinite. La classe FileMaker definisce vari metodi che consentono di utilizzare le liste valori definite in un database FileMaker Pro.

### Recupero dei nomi di tutte le liste valori per un formato specifico

Per un determinato oggetto del formato, utilizzare il metodo `listValueLists()` per recuperare una matrice che contiene i nomi delle liste valori.

#### Esempio

```
$valueListNames = $currentLayout->listValueLists();
```

### Recupero di una matrice di tutte le liste valori per un formato specifico

Per un determinato oggetto del formato, utilizzare il metodo `getValueListsTwoFields()` per recuperare una matrice che contiene i valori di tutte le liste valori. La matrice restituita è una matrice associativa. Le chiavi della matrice sono nomi della lista valori e i valori della matrice sono matrici associative che elencano i nomi da visualizzare e le scelte corrispondenti di ciascuna lista valori.

#### Esempio

```
$valueListsArray = $currentLayout->getValueListsTwoFields();
```

**Nota** Sebbene il metodo `getValueLists()` sia ancora supportato in FileMaker API per PHP, è obsoleto. Utilizzare invece il metodo `getValueListsTwoFields()`.

### Recupero dei valori per una lista valori indicata

Per un determinato oggetto del formato, utilizzare il metodo `getValueListTwoFields()` per ottenere una matrice di scelte definite per la lista valori indicata. La matrice restituita è una matrice associativa che contiene i valori visualizzati del secondo campo della lista valori come chiavi, e i valori memorizzati associati del primo campo, come valori della matrice.

A seconda delle opzioni selezionate nella finestra di dialogo Specifica i campi per la lista valori nel database FileMaker, il metodo `getValueListTwoFields()` restituisce, come valori memorizzati e visualizzati, il valore solo nel primo campo, solo nel secondo campo o in entrambi i campi di una lista valori.

- Se non è selezionato **Visualizza anche i valori secondo campo**, il metodo `getValueListTwoFields()` restituisce il valore del primo campo della lista valori sia come valore memorizzato che come valore visualizzato.
- Se **Visualizza anche i valori secondo campo** e **Mostra valori solo dal secondo campo** sono entrambi selezionati, il metodo `getValueListTwoFields()` restituisce il valore del primo campo come valore memorizzato e il valore del secondo campo come valore visualizzato.
- Se è selezionato **Visualizza anche i valori secondo campo**, ma non **Mostra valori solo dal secondo campo**, il metodo `getValueListTwoFields()` restituisce il valore del primo campo come valore memorizzato ed entrambi i valori sia del primo che del secondo campo come valore visualizzato.

Per trovare il valore visualizzato e il valore memorizzato, utilizzare un iteratore con il metodo `getValueListTwoFields()`.



### Esempio

```
$layout = $fm->getLayout('customers');
$valuearray = $layout->getValueListTwoFields("region", 4);
foreach ($valuearray as $displayValue => $value) {
    ....
}
```

### Note

- Sebbene il metodo `getValueList()` sia ancora supportato in FileMaker API per PHP, è obsoleto. Utilizzare invece il metodo `getValueListTwoFields()`.
- Quando si utilizza il metodo `getValueListTwoFields()`, assicurarsi di utilizzare un ciclo `foreach` per scorrere attraverso la matrice associativa. Non utilizzare un ciclo `for` dato che può restituire risultati non previsti.

## Esecuzione di richieste di ricerca

La classe FileMaker definisce quattro tipi di oggetti comando Find:

- Comando Find All. Vedere “Utilizzo del comando Find All” a pagina 82.
- Comando Find Any. Vedere “Utilizzo del comando Find Any” a pagina 82.
- Comando Find. Vedere “Utilizzo del comando Find” a pagina 82.
- Comando Compound Find. Vedere “Utilizzo di un comando Compound Find” a pagina 83.

La classe FileMaker definisce anche vari metodi che possono essere utilizzati per tutti e quattro i tipi di comando Find:

- Utilizzare il metodo `addSortRule()` per aggiungere una regola che definisce il modo in cui ordinare il gruppo di risultati. Utilizzare il metodo `clearSortRules()` per cancellare tutti i criteri di ordinamento definiti.
- Utilizzare il metodo `setLogicalOperator()` per scegliere ricerche con operatore logico AND oppure OR.
- Utilizzare il metodo `setRange()` per richiedere solo una parte del gruppo di risultati. Utilizzare il metodo `getRange()` per recuperare la definizione dell'intervallo corrente.

Usando il metodo `setRange()` si possono migliorare le prestazioni della soluzione riducendo il numero di record restituiti dalla richiesta di ricerca. Ad esempio, se una richiesta di ricerca restituisce 100 record, è possibile suddividere il gruppo di risultati in cinque gruppi da 20 record piuttosto che elaborare tutti i 100 record insieme.

- È possibile eseguire script FileMaker insieme a comandi Find.
  - Per eseguire uno script prima di eseguire il comando Find, utilizzare il metodo `setPreCommandScript()`.
  - Per eseguire uno script prima di ordinare il gruppo di risultati, utilizzare il metodo `setPreSortScript()`.
  - Per eseguire uno script dopo che un gruppo di risultati è stato generato, ma non ancora ordinato, utilizzare il metodo `setScript()`.

## Utilizzo del comando Find All

Utilizzare il comando Find All per recuperare tutti i record di un determinato formato. Utilizzare il metodo `newFindAllCommand()`, indicando un formato specifico, per creare un oggetto `FileMaker_Command_FindAll`. Quindi, eseguire la richiesta di ricerca utilizzando il metodo `execute()`.

### Esempio

```
$findCommand = $fm->newFindAllCommand('Form View');  
$result = $findCommand->execute;
```

**Nota** Quando si utilizza il comando Find All, evitare problemi di sovraccarico della memoria del computer specificando un numero massimo predefinito di record da restituire per pagina.

## Utilizzo del comando Find Any

Utilizzare il comando Find Any per recuperare un record a caso da un determinato formato. Utilizzare il metodo `newFindAnyCommand()`, indicando un formato specifico, per creare un oggetto `FileMaker_Command_FindAny`. Quindi, eseguire la richiesta di ricerca utilizzando il metodo `execute()`.

### Esempio

```
$findCommand = $fm->newFindAnyCommand('Form View');  
$result = $findCommand->execute;
```

## Utilizzo del comando Find

Utilizzare il metodo `newFindCommand()`, indicando un formato specifico, per creare un oggetto `FileMaker_Command_Find`. Quindi, eseguire la richiesta di ricerca utilizzando il metodo `execute()`.

**Nota** Assicurarsi che il nome del formato sia univoco. Se il database ha due formati con lo stesso nome, FileMaker API per PHP non riesce a distinguerli. Inoltre, l'API non fa distinzione tra maiuscole e minuscole. Ad esempio, se il database ha un formato Sitiweb e un altro formato SitiWeb, l'API non riesce a distinguerli.

Utilizzare il metodo `addFindCriterion()` per aggiungere dei criteri alla richiesta di ricerca. Utilizzare il metodo `clearFindCriteria()` per cancellare tutti i criteri di ricerca definiti.

### Esempi

Ricerca di un record in base al nome del campo:

```
$findCommand = $fm->newFindCommand('Form View');  
$findCommand->addFindCriterion('Questionnaire ID',  
$active_questionnaire_id);  
$result = $findCommand->execute();
```

Aggiunta di un criterio di ordinamento:

```
$findCommand = $fm->newFindCommand('Customer List');  
$findCommand->addSortRule('Title', 1, FILEMAKER_SORT_ASCEND);  
$result = $findCommand->execute();
```

### Utilizzo di un comando Compound Find

Il comando Compound Find consente di combinare più oggetti Find Request in un singolo comando. Esistono vari modi per creare un comando Compound Find:

- Creare un oggetto `FileMaker_Command_CompoundFind` utilizzando il metodo `newCompoundFindCommand()`.
- Creare uno o più oggetti `FileMaker_Command_FindRequest` utilizzando il metodo `newFindRequest()`.
- Utilizzare il metodo `add()` per aggiungere oggetti Find Request a un oggetto comando Compound Find.
- Eseguire il comando Compound Find utilizzando il metodo `execute()`.

## Esempio

### Comando Compound Find:

```
// Create the Compound Find command object
$compoundFind = $fm->newCompoundFindCommand('Form View');

// Create first find request
$findreq1 = $fm->newFindRequest('Form View');

// Create second find request
$findreq2 = $fm->newFindRequest('Form View');

// Create third find request
$findreq3 = $fm->newFindRequest('Form View');

// Specify search criterion for first find request
$findreq1->addFindCriterion('Quantity in Stock', '<100');

// Specify search criterion for second find request
$findreq2->addFindCriterion('Quantity in Stock', '0');

// Specify search criterion for third find request
$findreq3->addFindCriterion('Cover Photo Credit', 'The London Morning
News');

// Add find requests to compound find command
$compoundFind->add(1,$findreq1);
$compoundFind->add(2,$findreq2);
$compoundFind->add(3,$findreq3);

// Set sort order
$compoundFind->addSortRule('Title', 1, FILEMAKER_SORT_DESCEND);

// Execute compound find command
$result = $compoundFind->execute();

// Get records from found set
$records = $result->getRecords();

// Print number of records found
echo 'Found '. count($records). " results.<br><br>";
```

## Elaborazione dei record in un gruppo di risultati

- Recuperare una matrice contenente tutti i record del gruppo di risultati utilizzando il metodo `getRecords()`. Ogni elemento della matrice è un oggetto `FileMaker_Record` o un'istanza del nome classe impostato nell'API per creare un'istanza di record. La matrice può essere vuota se il gruppo di risultati non contiene nessun record.
- Ottenere una lista dei nomi dei campi per tutti i campi del gruppo di risultati utilizzando il metodo `getFields()`. Il metodo restituisce solo i nomi dei campi. Per ottenere ulteriori informazioni sui campi, utilizzare l'oggetto del formato associato.
- Ottenere il numero di record di tutto il gruppo trovato utilizzando il metodo `getFoundSetCount()`.
- Ottenere il numero di record del gruppo trovato filtrato utilizzando il metodo `getFetchCount()`. Se nel comando Find non sono stati indicati dei parametri di intervallo, questo valore è uguale al risultato ottenuto con il metodo `getFoundSetCount()`. È sempre uguale al valore di `count($response->getRecords())`.
- Per un determinato record, utilizzare il metodo `getField()` per restituire i contenuti di un campo sotto forma di stringa.
- Per un record specifico utilizzare il metodo `getFieldAsTimestamp()` per restituire i contenuti di un campo sotto forma di indicatore data e ora Unix (rappresentazione interna PHP di una data).
  - Se il campo è un campo data, l'indicatore data e ora per la data del campo è impostato su mezzanotte.
  - Se il campo è un campo ora, l'indicatore data e ora per l'ora è impostato su 1 gennaio 1970.
  - Se il campo è un campo indicatore data e ora, il valore dell'indicatore data e ora FileMaker applica direttamente i valori dell'indicatore data e ora Unix.
  - Se il campo indicato non è un campo data o ora o se l'indicatore data e ora generato non è compreso nell'intervallo, il metodo `getFieldAsTimestamp()` restituisce un oggetto `FileMaker_Error`.
- Per un record specifico, utilizzare il metodo `getContainerData()` per restituire un oggetto del campo Contenitore come dati binari:

```
<IMG src="img.php?-url=<?php echo urlencode($record->getField('Cover
Image')); ?>">
echo $fm->getContainerData($_GET['-url']);
```

- Per un record specifico, utilizzare il metodo `getContainerDataURL()` per restituire un URL completamente qualificato per l'oggetto del campo Contenitore:

```
// For images, use the HTML img tag
echo '';
// For movies and PDF files, use the HTML embed tag
//echo '<embed src="' . $fm->
getContainerDataURL($record->getField('container')) . '">';
```

## Limitazione delle righe del portale restituite dalle richieste di ricerca

In una soluzione con molti record correlati, l'interrogazione e l'ordinamento dei record del portale potrebbe richiedere molto tempo. Per limitare il numero di record da visualizzare in un gruppo correlato, utilizzare il metodo `setRelatedSetsFilters()` con le richieste di ricerca. Il metodo `setRelatedSetsFilters()` comprende due argomenti:

- un valore del filtro dei set correlati: `layout` o `none`
  - Indicando il valore `none`, il Motore per la Pubblicazione Web restituisce tutte le righe del portale e i record del portale non vengono preordinati.
  - Indicando il valore `layout`, vengono applicate le impostazioni specificate nella finestra di dialogo Impostazione portale di FileMaker Pro. I record sono ordinati in base all'ordinamento definito nella finestra di dialogo Impostazione portale, con il gruppo di record filtrato in modo da partire con la riga iniziale specificata.
- il numero minimo di record del portale restituito: un valore intero o `all`
  - Questo valore è utilizzato solo se l'impostazione **Consenti scorrimento verticale** è selezionata nella finestra di dialogo Impostazione portale. Se viene indicato un valore intero, viene restituito un numero di righe pari a quel numero a partire dalla riga iniziale. Indicando `all`, il Motore per la Pubblicazione Web restituisce tutti i record correlati.
  - Se l'impostazione **Consenti scorrimento verticale** non è selezionata, l'impostazione **Numero di righe** nella finestra di dialogo Impostazione portale determina il numero massimo di record correlati restituiti.

**Nota** L'impostazione **Filtra record portale** nella finestra di dialogo Impostazione portale non è supportata per le query PHP. Eventuali calcoli specificati per l'impostazione **Filtra record portale** vengono ignorati.

## Pre-convalida di comandi, record e campi

La classe FileMaker consente di *pre-convalidare* i dati di un campo di una soluzione PHP sul server Web prima di salvare i dati nel database.

Se si decide di usare una pre-convalida, considerare il numero dei valori dei dati che l'utente Web sta inserendo. Se l'utente sta aggiornando un numero ridotto di campi, le prestazioni possono essere migliorate evitando di utilizzare una pre-convalida. Ma se l'utente sta inserendo dati in molti campi, la pre-convalida fa sì che non accada che un record venga respinto dal database a causa di errori di convalida.

Con la classe FileMaker, il motore PHP esegue la pre-convalida delle seguenti limitazioni di campo:

- Non vuoto  
I dati validi sono una stringa di caratteri non vuota. I dati devono contenere almeno un carattere.
- Solo numerico  
I dati validi contengono solo caratteri numerici.
- Numero massimo di caratteri  
I dati validi contengono per lo più il numero massimo di caratteri specificato.

- Anno a quattro cifre

I dati validi sono una stringa di caratteri che indica una data con un anno a quattro cifre nel formato m/g/aaaa, dove m è un numero tra 1 e 12 compreso, g è un numero tra 1 e 31 compreso e aaaa è un numero a quattro cifre tra 0001 e 4000 compreso. Ad esempio, 1/30/3030 è un valore anno a quattro cifre valido. Al contrario, 4/31/2017 è un valore anno a quattro cifre non valido perché aprile non ha 31 giorni. La convalida della data supporta come delimitatori i caratteri barra (/), barra rovesciata (\) e trattino (-). Tuttavia la stringa non può contenere una combinazione di delimitatori. Ad esempio, 1\30-2017 non è valido.

- Ora del giorno

I dati validi sono una stringa di caratteri che indica un valore ora da 12 ore in uno di questi formati:

- h
- h:m
- h:m:s
- h:m:s AM/PM
- h:m AM/PM

dove h è un numero tra 1 e 12 compresi; m e s sono numeri tra 1 e 60 compresi.

La pre-convalida del motore PHP supporta il controllo automatico dei dati dei campi in base al tipo di campo:

- Data

Un campo definito come campo Data è convalidato in base al criterio “anno a quattro cifre”, fatta eccezione per il valore anno che può contenere da 0 a 4 cifre (il valore dell’anno può essere vuoto). Ad esempio 1/30 è una data valida anche se non viene specificato nessun anno.

- Ora

Un campo definito come campo Ora è convalidato in base al criterio “ora del giorno”, fatta eccezione per il componente ora (H) che può essere un numero tra 1 e 24 compreso, per supportare i valori dei formati a 24 ore.

- Indicatore data e ora

Un campo definito come campo Indicatore data e ora è convalidato in base al criterio “ora” per il componente ora e in base al criterio “data” per il componente data.

La classe FileMaker non può eseguire una pre-convalida di tutte le opzioni di convalida campo disponibili in FileMaker Pro. Le seguenti opzioni di convalida non possono essere pre-convalidate perché dipendono dallo stato di tutti i dati del database al momento del salvataggio dei dati:

- valore unico
- valore esistente
- da-a
- elemento della lista valori
- verifica mediante calcolo

### Pre-convalida dei record in un comando

Per un oggetto comando, utilizzare il metodo `validate()` per convalidare un campo o l'intero comando in base ai criteri di pre-convalida applicati dal motore PHP. Se viene utilizzato l'argomento del nome di campo opzionale, viene convalidato solo quel campo.

Se la pre-convalida riesce, il metodo `validate()` restituisce `TRUE`. Se la pre-convalida fallisce, il metodo `validate()` restituisce l'oggetto `FileMaker_Error_Validation` che contiene le informazioni su ciò che non ha reso possibile la convalida.

### Pre-convalida dei record

Per un oggetto record, utilizzare il metodo `validate()` per convalidare un campo o tutti i campi del record in base ai criteri di convalida applicati dal motore PHP. Se viene utilizzato l'argomento del nome di campo opzionale, viene convalidato solo quel campo.

Se la pre-convalida riesce, il metodo `validate()` restituisce `TRUE`. Se la pre-convalida fallisce, il metodo `validate()` restituisce l'oggetto `FileMaker_Error_Validation` che contiene le informazioni su ciò che non ha reso possibile la convalida.

### Pre-convalida dei campi

Per un oggetto campo, utilizzare il metodo `validate()` per stabilire se un determinato valore è valido per un campo oppure no.

Se la pre-convalida riesce, il metodo `validate()` restituisce `TRUE`. Se la pre-convalida fallisce, il metodo `validate()` restituisce l'oggetto `FileMaker_Error_Validation` che contiene le informazioni su ciò che non ha reso possibile la convalida.

### Elaborazione degli errori di convalida

Se la pre-convalida fallisce, l'oggetto `FileMaker_Error_Validation` restituito contiene una matrice a tre elementi per ogni errore di convalida:

1. L'oggetto del campo che non ha superato la pre-convalida
2. Un valore di convalida costante che indica il criterio di convalida che ha provocato l'errore:
  - 1 - `FILEMAKER_RULE_NOTEMPTY`
  - 2 - `FILEMAKER_RULE_NUMERICONLY`
  - 3 - `FILEMAKER_RULE_MAXCHARACTERS`
  - 4 - `FILEMAKER_RULE_FOURDIGITYEAR`
  - 5 - `FILEMAKER_RULE_TIMEOFDAY`
  - 6 - `FILEMAKER_RULE_TIMESTAMP_FIELD`
  - 7 - `FILEMAKER_RULE_DATE_FIELD`
  - 8 - `FILEMAKER_RULE_TIME_FIELD`
3. Il valore effettivo inserito nel campo che non ha superato la pre-convalida



È inoltre possibile utilizzare i seguenti metodi con un oggetto `FileMaker_Error_Validation`:

- Utilizzare il metodo `isValidatationError()` per verificare se l'errore è un errore di convalida.
- Utilizzare il metodo `numErrors()` per ottenere il numero di criteri di convalida che hanno dato errore.

### Esempio

```
//Create an Add request
$addrequest = $fm->newAddCommand('test', array('join' => 'added', 'maxchars'
=> 'abcx', 'field' => 'something', 'numericonly' => 'abc'));

//Validate all fields
$result = $addrequest->validate();

//If the validate() method returned any errors, print the name of the field,
the error number, and the value that failed.
if(FileMaker::isError($result)){
    echo 'Validation failed:'. "\n";
    $validationErrors= $result->getErrors();
    foreach ($validationErrors as $error) {
        $field = $error[0];
        echo 'Field Name: ' . $field->getName(). "\n";
        echo 'Error Code: ' . $error[1]. "\n";
        echo 'Value: ' . $error[2]. "\n";
    }
}
```

### Risultato

```
Validation failed:
Field Name: numericonly
Error Code: 2
Value: abc
Field Name: maxchars
Error Code: 3
Value: abcx
```

## Gestione degli errori

La classe FileMaker definisce l'oggetto FileMaker\_Error per supportare la gestione degli errori che si verificano in una soluzione PHP.

L'errore può verificarsi durante l'esecuzione di un comando. Se si verifica un errore, il comando restituisce l'oggetto FileMaker\_Error. Si consiglia di controllare l'errore che viene restituito quando viene eseguito un comando.

Utilizzare i seguenti metodi per avere ulteriori informazioni sull'errore indicato nell'oggetto FileMaker\_Error.

- Controllare se una variabile è un oggetto FileMaker Error utilizzando il metodo `isError()`.
- Ricavare il numero di errori che si sono verificati utilizzando il metodo `numErrors()`.
- Recuperare una matrice di matrici che descrivono gli errori che si sono verificati utilizzando il metodo `getErrors()`.
- Visualizzare un messaggio di errore utilizzando il metodo `getMessage()`.

### Esempio

```
$result = $findCommand->execute();
if (FileMaker::isError($result)) {
    echo "<p>Error: " . $result->getMessage() . "</p>";
    exit;
}
```

Per informazioni sui codici di errore restituiti con l'oggetto FileMaker Error, vedere appendice A, "Codici di errore per la Pubblicazione Web personalizzata."

# Capitolo 9

## Preparazione, test e monitoraggio di un sito

In questo capitolo sono contenute le istruzioni per preparare e testare un sito Pubblicazione Web personalizzata prima della sua distribuzione in un ambiente di produzione. Sono anche presenti le istruzioni per utilizzare i file di registro per controllare il sito durante il test o dopo la sua distribuzione.

### Preparazione di un sito Pubblicazione Web personalizzata

Prima di poter testare correttamente il sito, è necessario copiare o spostare i file richiesti nelle posizioni corrette sul/sui server di preparazione.

1. Completare tutti i passi riportati nel capitolo 2, “Preparazione dei database per la Pubblicazione Web personalizzata.”
2. Controllare che la Pubblicazione Web personalizzata sia stata attivata e configurata correttamente nella FileMaker Server Admin Console. Vedere la [Guida di FileMaker Server](#).
3. Verificare che il server Web e il Motore per la Pubblicazione Web siano in esecuzione.
4. Copiare o spostare i file del sito sul componente server Web della distribuzione di FileMaker Server.

Copiare o spostare i file del sito nella seguente directory:

- IIS (Windows) con HTTP o HTTPS:  
`[unità]:\Programmi\FileMaker\FileMaker Server\HTTPServer\Conf`  
dove [unità] è l'unità della macchina master FileMaker Server.
- Apache (macOS) con HTTP:  
`/Libreria/FileMaker Server/HTTPServer/htdocs`
- Apache (macOS) con HTTPS:  
`/Libreria/FileMaker Server/HTTPServer/htdocs/httpsRoot`

**Nota** Se si seleziona **Usare HSTS per client Web** nell'Admin Console, utilizzare la directory HTTPS per ospitare i file del sito.

5. Se non lo si è già fatto, copiare o spostare eventuali oggetti dei campi Contenitore di riferimento nella directory corretta sulla macchina master.
  - Se il file di database è correttamente ospitato e accessibile sul componente server database della distribuzione di FileMaker Server e se i campi Contenitore memorizzano i file effettivi nel database FileMaker, non è necessario riposizionare i contenuti dei campi Contenitore.
  - Se un campo Contenitore del database contiene un riferimento al file anziché il file reale, l'oggetto Contenitore di riferimento deve essere memorizzato nella cartella Web di FileMaker Pro quando il record viene creato o modificato. Per preparare il sito, copiare o spostare i contenitori a cui si fa riferimento ad una cartella con la stessa posizione relativa nella cartella principale del software del server Web.
  - Se si utilizza FileMaker Pro per caricare un database con campi Contenitore che memorizzano oggetti esternamente, i dati dei campi Contenitore memorizzati esternamente vengono caricati su FileMaker Server come parte del processo. Per informazioni sul trasferimento dei file di database in FileMaker Server, consultare la [Guida di FileMaker Pro](#).
  - Se si carica manualmente un database che utilizza un campo Contenitore con oggetti memorizzati esternamente, è necessario copiare o spostare gli oggetti a cui fa riferimento in una sottocartella della cartella RC\_Data\_FMS, come descritto in “Campi Contenitore con dati memorizzati esternamente” a pagina 16.
6. Copiare eventuali componenti aggiuntivi dell'applicazione Web sulla macchina master. Per la Pubblicazione Web personalizzata con XML, l'applicazione Web elabora i dati XML prima di inviarli a un'altra applicazione o al client.

## Test di un sito Pubblicazione Web personalizzata

Prima di comunicare agli utenti che un sito Pubblicazione Web Immediata è disponibile, è importante verificare che abbia il layout desiderato e che funzioni nel modo previsto.

- Testare le funzioni come ad esempio la ricerca, l'aggiunta, l'eliminazione e l'ordinamento dei record con account e set di privilegi diversi.
- Verificare che i set di privilegi funzionino come previsto in base al tipo di account utilizzato. Assicurarsi che gli utenti non autorizzati non possano accedere o modificare i dati.
- Controllare tutti gli script per verificare che producano il risultato previsto. Per informazioni su come progettare script compatibili con il Web consultare la sezione “Script FileMaker e Pubblicazione Web personalizzata” a pagina 18.
- Testare il sito con sistemi operativi e browser Web diversi.
- Quando si creano soluzioni che utilizzano FileMaker API per PHP, si consiglia di crearle con il supporto dei cookie attivato. Con i cookie attivati, infatti, FileMaker API per PHP ha tempi di risposta migliori. I cookie non sono necessari per utilizzare le funzioni della Pubblicazione Web personalizzata, ma permettono al Motore per la Pubblicazione Web di memorizzare nella cache le informazioni della sessione.

**Nota** È possibile visualizzare e testare il sito sulla macchina master senza utilizzare una connessione di rete, inserendo `http://127.0.0.1/` nell'URL.

- Per le soluzioni PHP, utilizzare `http://127.0.0.1/<percorso_sito>` dove `<percorso_sito>` è il percorso relativo alla home page del sito.
- Per informazioni sulla sintassi URL nelle soluzioni XML, vedere “Sintassi URL per dati XML e oggetti Contenitore” a pagina 25.

## Fogli di stile per testare l'output XML

### Esempi

Di seguito vi sono due esempi di fogli di stile XSLT utili per testare l'output XML.

L'esempio di foglio di stile che segue genera i dati XML richiesti senza fare nessuna trasformazione. Questo foglio di stile è utile per visualizzare i dati XML che il Motore per la Pubblicazione Web sta effettivamente utilizzando.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fmrs="http://www.filemaker.com/xml/fmresultset">
  <xsl:output method="xml"/>
  <xsl:template match="/">
    <xsl:copy-of select="."/>
  </xsl:template>
</xsl:stylesheet>
```

Per eseguire il debug di un foglio di stile, è possibile utilizzare il seguente esempio di un tag `<textarea>` HTML per visualizzare il documento origine XML al quale è stato eseguito l'accesso dal foglio di stile in un'area di testo a scorrimento.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fmrs="http://www.filemaker.com/xml/fmresultset">
  <xsl:output method="html"/>
  <html>
    <body>
      <xsl:template match="/fmrs:fmresultset">
        <textarea rows="20" cols="100">
          <xsl:copy-of select="."/>
        </textarea><br/>
      </xsl:template>
    </body>
  </html>
</xsl:stylesheet>
```

## Monitoraggio di un sito

È possibile utilizzare i seguenti tipi di file registro per monitorare un sito Pubblicazione Web personalizzata e raccogliere informazioni sugli utenti Web che visitano il sito:

- Registri degli accessi e degli errori del server Web
- Registro Motore per la Pubblicazione Web
- Registro errori Modulo Server Web
- Registri Tomcat

### Uso dei registri degli accessi e degli errori del server Web

IIS (Windows): Il server Web Microsoft IIS genera file di registro degli accessi e mostra gli errori nel Visualizzatore eventi di Windows anziché registrarli in un file di registro. Il file di registro degli accessi, incluso per impostazione predefinita nel formato di file registro esteso W3C, è un registro di tutte le richieste HTTP in ingresso sul server Web. È anche possibile usare il formato di file registro comune W3C per il registro degli accessi. Consultare la documentazione del server Web Microsoft IIS.

Apache (macOS): Il server Web Apache genera un file di registro degli accessi e un file di registro degli errori. Il file di registro degli accessi Apache, incluso per impostazione predefinita nel formato di file registro comune W3C, è un registro di tutte le richieste HTTP in ingresso sul server Web. Il registro errori Apache è un registro dei problemi che riguardano richieste HTTP di elaborazione. Consultare la documentazione del server Web Apache.

**Nota** Per informazioni sul formato di file di registro comune W3C e sul formato di file di registro esteso W3C, vedere il sito Web del World Wide Web Consortium all'indirizzo [www.w3.org](http://www.w3.org).

### Utilizzo del registro del Motore per la Pubblicazione Web

Per impostazione predefinita, il Motore per la Pubblicazione Web genera un file di registro chiamato `wpe.log` che è un registro degli errori del Motore per la Pubblicazione Web, compresi errori dell'applicazione, errori di utilizzo e errori di sistema. È anche possibile fare in modo che il Motore per la Pubblicazione Web comprenda informazioni relative alla Pubblicazione Web personalizzata, come le richieste XML dell'utente finale per generare dati di uscita di pubblicazione Web o modifiche alle impostazioni della Pubblicazione Web personalizzata.

Il file `wpe.log` si trova sul componente Motore per la Pubblicazione Web della distribuzione di FileMaker Server:

- **Windows:** `[unità]:\Programmi\FileMaker\FileMaker Server\Logs\wpe.log` dove `[unità]` è l'unità principale da cui viene avviato il sistema.
- **macOS:** `/Libreria/FileMaker Server/Logs/wpe.log`

## Impostazioni del registro Motore per la Pubblicazione Web

Il file `wpe.log` viene generato se l'impostazione **Consenti registrazione per Pubblicazione Web** è abilitata nell'Admin Console.

Opzione registro abilitata	Informazioni registrate nel <code>wpe.log</code>
Messaggi livello errore	Gli errori Motore per la Pubblicazione Web, compresi gli errori di applicazione, errori di utilizzo ed errori di sistema.
Messaggi livello errore e info	Gli errori sopra descritti e le informazioni sull'accesso al Motore per la Pubblicazione Web. Contiene un record con tutte le richieste degli utenti finali XML per generare dati di uscita della Pubblicazione Web personalizzata.

L'impostazione **Messaggi livello errore** è attivata in modo predefinito. Per informazioni sull'impostazione di queste opzioni utilizzando l'Admin Console, vedere la [Guida di FileMaker Server](#).

**Importante** Col tempo il file `wpe.log` può diventare molto grande. Utilizzare l'Admin Console per impostare le dimensioni massime del file `wpe.log`. Quando il file `wpe.log` raggiunge queste dimensioni massime, il Motore per la Pubblicazione Web copia il file `wpe.log` in un singolo file di backup, `wpe.log.1`, e crea un nuovo file `wpe.log`. Se si desidera più di una copia di backup, salvare regolarmente un archivio del file `wpe.log.1`.

## Formato del registro del Motore per la Pubblicazione Web

Il file `wpe.log` utilizza la seguente formattazione per ogni voce:

```
[TIMESTAMP_GMT] [WPC_HOSTNAME] [CLIENT_IP:PORT] [ACCOUNT_NAME] [MODULE_TYPE]
[SEVERITY] [FM_ERRORCODE] [RETURN_BYTES] [MESSAGE]
```

dove:

- `[TIMESTAMP_GMT]` è la data e l'ora dell'inserimento, ora di Greenwich (GMT).
- `[WPC_HOSTNAME]` è il nome della macchina master.
- `[CLIENT_IP:PORT]` è l'indirizzo IP e la porta del client dove è stata originata la richiesta XML.
- `[ACCOUNT_NAME]` è il nome dell'account utilizzato per l'accesso al database FileMaker ospitato.
- `[MODULE_TYPE]` è: XML, per la Pubblicazione Web personalizzata con richieste XML, oppure PHP, per la Pubblicazione Web personalizzata con le richieste PHP.
- `[SEVERITY]` è INFO, che indica un messaggio informativo, oppure ERROR, che indica un messaggio di errore.
- `[FM_ERROR_CODE]` è il numero dell'errore restituito per un messaggio di errore. Il numero dell'errore può essere un codice di errore per i database FileMaker (vedere "Codici di errore per database FileMaker" a pagina 99).  
Inoltre, il numero dell'errore può essere un numero di errore HTTP, preceduto da una stringa "HTTP:".
- `[RETURN_BYTES]` è il numero dei byte restituiti dalla richiesta.
- `[MESSAGE]` fornisce informazioni aggiuntive sulla voce di registro.

## Registro messaggi del Motore per la Pubblicazione Web

### Esempi

Gli esempi di seguito mostrano i tipi di messaggio che possono essere inclusi nel file wpe.log.

Quando il Motore per la Pubblicazione Web viene avviato e arrestato:

```
2017-06-02 15:15:31 -0700 - - - - INFO - - FileMaker Server
Web Publishing Engine started.
2017-06-02 15:46:52 -0700 - - - - INFO - - FileMaker Server
Web Publishing Engine stopped.
```

Richieste query XML riuscite o fallite:

```
2017-06-02 15:21:08 -0700 WPC_SERVER 192.168.100.101:0 jdoe XML INFO
0 3964 "/fmi/xml/fmresultset.xml?-db=Contacts&-lay=Contact_Details&-
findall"
2017-06-02 15:26:31 -0700 WPC_SERVER 192.168.100.101:0 jdoe XML
ERROR 5 596 "/fmi/xml/fmresultset.xml?-db=Contacts&-
layout=Contact_Details&-findall"
```

Errori di script:

```
2017-06-02 17:33:12 -0700 WPC_SERVER 192.168.100.101:0 jdoe - ERROR
4 - Web Scripting Error: 4, File: "10b_MeetingsUpload", Script: "OnOpen",
Script Step: "Show Custom Dialog"
```

Modifiche alle impostazioni della Pubblicazione Web personalizzata:

```
2017-06-09 10:59:49 -0700 WPC_SERVER 192.168.100.101:0 jdoe - INFO
- - XML Web Publishing Engine is enabled.
```

Errori di sistema:

```
2017-06-02 15:30:42 -0700 WPC_SERVER 192.168.100.101:0 jdoe XML
ERROR - - Communication failed
```

### Utilizzo del registro errori del Modulo Server Web

Se il server Web non è in grado di connettersi al Motore per la Pubblicazione Web, il Modulo Server Web genera un file di registro che è un registro di tutti gli errori che si sono verificati in questa operazione. Questo file si chiama `web_server_module_log.txt` e si trova nella cartella Logs all'interno della cartella FileMaker Server sull'host del server Web.



## Utilizzo dei registri Tomcat

Quando FileMaker Server ha un problema provocato da un errore nel server Web interno, potrebbe essere utile visualizzare i registri Tomcat. I registri Tomcat si trovano sul componente server Web della distribuzione di FileMaker Server:

- Windows:
  - [unità]:\Programmi\FileMaker\FileMaker Server\Admin\admin-master-tomcat\logs\  
dove [unità] è l'unità principale da cui viene avviato il sistema.
  - [unità]:\Programmi\FileMaker\FileMaker Server\Web Publishing\publishing-engine\jwpc-tomcat\logs  
dove [unità] è l'unità principale da cui viene avviato il sistema.
- macOS:
  - /Libreria/FileMaker Server/Admin/admin-master-tomcat/logs/
  - /Libreria/FileMaker Server/Web Publishing/publishing-engine/jwpc-tomcat/logs

# Appendice A

## Codici di errore per la Pubblicazione Web personalizzata

Il Motore per la Pubblicazione Web genera codici di errore per errori di database e stringhe di query che possono verificarsi durante una richiesta dati XML.

Per un elenco dei codici di errore aggiornati, vedere la [Knowledge Base di FileMaker](#).

### Codici di errore in formato XML

Il Motore per la Pubblicazione Web genera un codice di errore per i database pubblicati in formato XML ogni volta che si verifica una richiesta di dati. Questo tipo di valore di codice di errore è inserito all'inizio del documento XML nell'elemento `<error code>` per la grammatica `fmresultset`, oppure nell'elemento `<ERRORCODE>` per le grammatiche `FMPXMLRESULT` o `FMPXMLLAYOUT`. Un codice di errore pari a 0 indica che non si è verificato nessun errore.

#### Esempi

Codice di errore database nella grammatica `fmresultset`:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE fmresultset PUBLIC "-//FMI//DTD fmresultset//EN"
"http://192.168.123.101/fmi/xml/fmresultset.dtd">
<fmresultset xmlns="http://www.filemaker.com/xml/fmresultset"
version="1.0">
  <error code="0"></error>
```

Codice di errore database nella grammatica `FMPXMLRESULT`:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE FMPXMLRESULT PUBLIC "-//FMI//DTD FMPXMLRESULT//EN"
"http://192.168.123.101/fmi/xml/FMPXMLRESULT.dtd">
<fmpxmlresult xmlns="http://www.filemaker.com/fmpxmlresult">
  <ERRORCODE>0</ERRORCODE>
```

Spetta allo sviluppatore della soluzione di Pubblicazione Web personalizzata controllare il valore dell'elemento `<error code>` o `<ERRORCODE>` e gestirlo opportunamente. Il Motore per la Pubblicazione Web non gestisce gli errori di database.

## Codici di errore per database FileMaker

Per i codici di errore di FileMaker Pro, consultare la [Guida di FileMaker Pro](#).

FileMaker Server restituisce il codice di errore 959 per indicare che una tecnologia è stata disabilitata.

Ad esempio, se l'amministratore server disabilita la Pubblicazione Web personalizzata con XML utilizzando **Pubblicazione Web > scheda XML** nella FileMaker Server Admin Console, le query XML restituiscono il codice di errore 959.

### Esempio

Codice di errore 959 nella grammatica FMPXMLLAYOUT:

```
<FMPXMLLAYOUT>
  <ERRORCODE>959</ERRORCODE>
  <LAYOUT DATABASE="" NAME="" />
  <VALUELISTS />
</FMPXMLLAYOUT>
```

# Indice

## A

- account e privilegi
  - account Ospite 14
  - attivazione per Pubblicazione Web personalizzata 13
  - script 18
- account Ospite
  - attivazione 14
  - con Pubblicazione Web personalizzata 14
  - disattivazione 14
- Admin Console 14, 25, 67
- Admin Console di FileMaker Server 14, 25
- applicazione dei formati per una risposta XML 40
- attivazione della Pubblicazione Web personalizzata in un database 13
- attributo auto-enter 31
- attributo four-digit-year 31
- attributo max-characters 31
- attributo max-repeat 31
- attributo name 31
- attributo not-empty 31
- attributo numeric-only 31
- attributo result 31
- attributo time-of-day 31
- attributo type 31
- autenticazione di base per utenti Web 13
- autenticazione di utenti Web 13

## C

- campi
  - anno a quattro cifre 87
  - attributi 31
  - Calcolo 28, 29
  - Contenitore 15, 26, 33
  - correlati in PHP 78
  - correlati in XML 31, 45
  - Data 33, 36, 87
  - globali 29
  - Indicatore data e ora 33, 36, 87
  - multipli 26, 45
  - nomi di campo completamente qualificati 45
  - non vuoti 86
  - Numero 33
  - numero massimo di caratteri 86
  - Ora 33, 36, 87
  - Ora del giorno 87
  - portale 31
  - portali 45
  - Riassunto 28, 29
  - solo numerici 86
  - Testo 33

- campi Contenitore
  - accesso ai dati da parte degli utenti Web 17
  - con dati memorizzati esternamente 16
  - con file di riferimento 16
  - download progressivo 17
  - pubblicazione dei contenuti di 15
  - sintassi URL per accedere a soluzioni XML 26
- campi globali
  - nella definizione del campo 31
  - sintassi 47
- campo anno a quattro cifre 87
- campo Data 87
- campo Indicatore data e ora 85, 87
- campo non vuoto 86
- campo numero massimo di caratteri 86
- campo Ora 87
- campo Ora del giorno 87
- campo solo numerico 86
- caratteri ASCII, in documenti XML 38
- cartella Web, copia degli oggetti campo Contenitore 16
- classe FileMaker 71
- client URL library 65
- codici di errore database 29, 98
- codifica
  - dati XML 28, 38
  - URL 27
- codifica Latin-1 69
- codifica testo
  - dati XML generati 28
  - URL 27
- codifica testo URL 27
- codifica UTF-8 69
- comandi per le query. *Vedere* stringhe di query
- comando Add 73
- comando Delete 74
- comando Duplicate 73
- comando Edit 74
- comando Find 82
- comando Find All 82
- comando Find Any 82
- comando query `-dbnames` 47
- comando query `-delete` 47
- comando query `-dup` 48
- comando query `-edit` 48
- comando query `-find` 49
- comando query `-findall` 49
- comando query `-findany` 49
- comando query `-findquery` 49
- comando query `-layoutnames` 50
- comando query `-new` 50
- comando query `-scriptnames` 51
- comando query `-view` 51

- Compound Find
  - comando 83
  - esempio 84
- confronto tra grammatiche XML 28
- connessione a un database FileMaker con PHP 72
- convalida
  - anno a quattro cifre 87
  - campi 88
  - comandi 86
  - Data 87
  - Indicatore data e ora 87
  - non vuoto 86
  - numero massimo di caratteri 86
  - Ora 87
  - Ora del giorno 87
  - record 88
  - solo numerico 86
- convalida da-a 87
- convalida elemento della lista valori 87
- convalida valore esistente 87
- convalida valore univoco 87
- creazione di un record
  - con PHP 73
  - con XML 50
- Crittografia SSL (Secure Sockets Layer) 15
- cURL 65

## D

- database, protezione dei database pubblicati 14
- <datasource> element 30
- determinazione del formato per richiedere dati XML 40
- directory HTTP per PHP 69
- directory HTTPS per PHP 69
- documentazione di FileMaker Server 8
- documentazione online 8
- documentazione, FileMaker 8
- download progressivo 17
- DTD (Document Type Definition) 29, 33
- duplicazione di un record 73

## E

- e commerciali, in PHP 72
- elaborazione di un gruppo di risultati 85
- elaborazione di una richiesta del Motore per la Pubblicazione Web 10
- elementi
  - codice di errore database 29
  - nella grammatica FMPXMLLAYOUT 35
  - nella grammatica FMPXMLRESULT 33
  - nella grammatica fmresultset 30
- elementi <error code> e <ERRORCODE> 98
- elemento <field-definition> 31
- elemento <metadata> 31
- elemento <relatedset-definition> 31
- elemento <resultset> 31
- eliminazione dei record del portale 46
- eliminazione di un record 74

- errore memoria esaurita 18
- errori
  - codici di errore database 98
  - descrizione 98
  - elementi codice di errore database 29
  - file registro per server Web 94
  - gestione 90
- esecuzione di richieste di ricerca 81
- esempi di
  - dati generati con la grammatica FMPXMLLAYOUT 37
  - dati generati con la grammatica FMPXMLRESULT 34
  - dati generati con la grammatica fmresultset 32
- esportazione di dati XML 23

## F

- file registro 92
  - accesso server Web 94
  - descrizione 94
  - Tomcat 97
  - web\_server\_module\_log.txt 96
- file registro accessi per server Web, descrizione 94
- file registro web\_server\_module\_log.txt 96
- FileMaker API per PHP 11
  - informazioni di riferimento 70
  - installazione manuale 66
- FileMaker Pro, contrasti con Motore per la Pubblicazione Web 23
- FileMaker WebDirect 9
- filtro dei record campo portale 58
- fogli di stile, test 92
- formati
  - applicazione per una risposta XML 40
  - utilizzo in PHP 77
- formati database disponibili 50
- formato UTF-8 (Unicode Transformation a 8 bit) 27, 38

## G

- gestione degli errori 90
- grammatica FMPXMLLAYOUT 23, 28, 35–37
- grammatica FMPXMLRESULT 23, 28, 33–34
- grammatica fmresultset 23, 28, 29–32
- grammatiche per XML, descrizione 28
- gruppo di risultati 85
- gruppo fmsadmin 17

## H

- HSTS con PHP 69

## I

- importazione di dati XML 23
- impostazione Consenti scorrimento verticale 58, 86
- impostazione Filtra record portale 58, 86
- impostazione Numero di righe 58, 86
- impostazione Riga iniziale 58
- indicatore data e ora Unix 85

- indirizzo IP dinamico 65
- indirizzo IP statico 65
- informazioni di riferimento 70
- installazione di FileMaker API per PHP 66
- installazione manuale del FileMaker API per PHP 66
- interrogazione dei campi del portale 46
- istruzione di script Nuovo accesso 14
- istruzioni di script
  - Modifica password 14
  - Nuovo accesso 14

## L

- limitazione dei record campo portale 59
- liste valori
  - utilizzo in PHP 80
  - utilizzo in XML 35

## M

metodi

- add() 83
- addSortRule() 81
- clearSortRules() 81
- commit() 73
- createRecord() 73
- delete() 74, 79
- getContainerData() 85
- getContainerDataURL() 85
- getDatabase() 77
- getErrors() 90
- getFetchCount() 85
- getField() 85
- getFieldAsTimestamp() 85
- getFields() 77, 85
- getFoundSetCount() 85
- getLayout() 77
- getMessage() 90
- getName() 77, 79
- getRange() 81
- getRecords() 85
- getRelatedSet() 78
- getRelatedSets() 78
- getValueListsTwoFields() 80
- getValueListTwoFields() 80
- isError() 90
- isValidationError() 89
- listFields() 77
- listLayouts() 77
- listRelatedSets() 77, 78
- listScripts() 75
- listValueLists() 77, 80
- newAddCommand() 73
- newCompoundFindCommand() 83
- newDeleteCommand() 74
- newDuplicateCommand() 73
- newEditCommand() 74
- newFindAllCommand() 82
- newFindAnyCommand() 82
- newFindCommand() 82
- newFindRequest() 83

- newPerformScriptCommand() 75
- newRelatedRecord() 79
- numErrors() 89, 90
- setLogicalOperator() 81
- setPreCommandScript() 75, 81
- setPreSortScript() 76, 81
- setProperty() 72
- setRange() 81
- setRelatedSetsFilters() 86
- setResultsLayout() 77
- setScript() 76, 81
- validate() 88
- metodo add() 83
- metodo addSortRule() 81
- metodo clearSortRules() 81
- metodo commit() 73
- metodo createRecord() 73
- metodo delete() 74, 79
- metodo getContainerData() 85
- metodo getContainerDataURL() 85
- metodo getDatabase() 77
- metodo getErrors() 90
- metodo getFetchCount() 85
- metodo getField() 85
- metodo getFieldAsTimestamp() 85
- metodo getFields() 77, 85
- metodo getFoundSetCount() 85
- metodo getLayout() 77
- metodo getMessage() 90
- metodo getName() 77, 79
- metodo getRange() 81
- metodo getRecords() 85
- metodo getRelatedSet() 78
- metodo getRelatedSets() 78
- metodo getValueListsTwoFields() 80
- metodo getValueListTwoFields() 80
- metodo isError() 90
- metodo isValidationError() 89
- metodo listFields() 77
- metodo listLayouts() 77
- metodo listRelatedSets() 77, 78
- metodo listScripts() 75
- metodo listValueLists() 77, 80
- metodo newAddCommand() 73
- metodo newCompoundFindCommand() 83
- metodo newDeleteCommand() 74
- metodo newDuplicateCommand() 73
- metodo newEditCommand() 74
- metodo newFindAllCommand() 82
- metodo newFindAnyCommand() 82
- metodo newFindCommand() 82
- metodo newFindRequest() 83
- metodo newPerformScriptCommand() 75
- metodo newRelatedRecord() 79
- metodo numErrors() 89, 90
- metodo setLogicalOperator() 81
- metodo setPreCommandScript() 75, 81

- metodo setPreSortScript() 76, 81
- metodo setProperty() 72
- metodo setRange() 81
- metodo setRelatedSetsFilters() 86
- metodo setResultsLayout() 77
- metodo setScript() 76, 81
- metodo validate() 88
- modifica di un record 74
- moduli HTML per richieste XML 25
- monitoraggio di siti Web 94
- Motore per la Pubblicazione Web
  - Admin Console 25
  - codici di errore generati 98
  - descrizione 10
  - elaborazione di una richiesta 10
  - generazione di dati XML 24
  - generazione di documenti XML 25
  - registro dell'applicazione 94
  - vantaggi del 20
- Motore per la Pubblicazione Web personalizzata (CWPE) 24

## N

- namespace per XML 28
- nome di campo completamente qualificato, sintassi di 45
- nomi dei campi, completamente qualificati, sintassi 45
- nomi utente
  - autenticazione di base per utenti Web 13
  - definizione per la Pubblicazione Web personalizzata 13
- nuove funzioni nella Pubblicazione Web personalizzata 21

## O

- oggetti classe FileMaker
  - database 72
  - definizione 71
  - record 73
  - set correlato 78
- oggetti comando FileMaker
  - Add 73
  - Compound Find 83
  - Delete 74
  - Duplicate 73
  - Edit 74
  - Find 81, 82
  - Find All 82
  - Find Any 82
- oggetti comando Find 81
- oggetto del database 72
- oggetto record 73
- oggetto set correlato 78
- operatori di confronto per i campi 53
- operatori, di confronto 53
- ordinamento dei record campo portale 58
- ordine di elaborazione delle richieste XML 41

## P

- panoramica
  - passi per accedere ai dati XML 25
  - Pubblicazione Web personalizzata 9
- parametri per le query. *Vedere* stringhe di query
- parametro query –db 51
- parametro query –delete.related 46
- parametro query –field (contenitore) 52
- parametro query fieldname (non Contenitore) 52
- parametro query –fieldname.op 53
- parametro query –lay 40, 54
- parametro query –lay.response 40, 54
- parametro query –lop 55
- parametro query –max 55
- parametro query –modid 56
- parametro query –query 56
- parametro query –recid 57
- parametro query –relatedsets.filter 58
- parametro query –relatedsets.max 59
- parametro query –script 59
- parametro query –script.param 60
- parametro query –script.prefind 60
- parametro query –script.prefind.param 60
- parametro query –script.presort 61
- parametro query –script.presort.param 61
- parametro query –skip 62
- parametro query –sortfield 62
- parametro query –sortorder 63
- parola chiave fmphp per l'attivazione della pubblicazione PHP 13
- parola chiave fmxml per l'attivazione della pubblicazione XML 13, 25
- parole chiave per l'attivazione di Pubblicazione Web personalizzata 13, 25
- password
  - autenticazione di base per utenti Web 13
  - definizione per la Pubblicazione Web personalizzata 13
  - password non necessaria per l'accesso 14
  - script Modifica password 14
- PDF 8
- PHP
  - attivazione in un database 13
  - Pubblicazione Web personalizzata, descrizione 11
  - versione supportata 66
- plug-in 18
- portali
  - aggiunta dei record 45
  - eliminazione dei record 46
  - formato 58
  - modifica dei record 46
  - numero di record 58
  - ordinamento dei record 58
  - riga iniziale 58
  - utilizzo in PHP 78

pre-convalida  
 anno a quattro cifre 87  
 campi 88  
 comandi 86  
 Data 87  
 Indicatore data e ora 87  
 non vuoto 86  
 numero massimo di caratteri 86  
 Ora 87  
 Ora del giorno 87  
 record 88  
 solo numerico 86

privilegi di accesso 14

privilegio esteso per la Pubblicazione Web personalizzata 13

proprietà hostspec per PHP 72

protezione dei database pubblicati 14

pubblicazione statica, descrizione 9

pubblicazione sul Web  
 codici di errore database 98  
 con XML 25  
 connessione a Internet o intranet 22  
 oggetti campo Contenitore 15  
 protezione dei database 14  
 requisiti per 21

Pubblicazione Web personalizzata  
 accesso a soluzioni da parte di utenti Web 13  
 account Ospite 14  
 attivazione in un database 13  
 attivazione nel Motore per la Pubblicazione Web 14  
 con PHP 11  
 con XML 11, 23  
 definizione 9  
 limitazione dell'accesso degli indirizzi IP nel server Web 14  
 nuove funzioni 21  
 panoramica 9  
 plug-in per le soluzioni di pubblicazione Web 18  
 privilegio esteso per 13  
 requisiti per 21  
 script 19  
 utilizzo degli script 18  
 utilizzo di un indirizzo IP statico 22

## Q

query campo portale 58, 59

## R

rappresentazione della data 85

record  
 creazione in PHP 73  
 creazione in XML 50  
 duplicazione in PHP 73  
 duplicazione in XML 48  
 eliminazione in PHP 74  
 eliminazione in XML 47  
 modifica in PHP 74  
 modifica in XML 48  
 ricerca in PHP 81  
 ricerca in XML 49  
 salto in XML 62

recupero  
 informazioni sul formato 51  
 nomi degli script disponibili 51  
 nomi dei formati 50

registro dell'applicazione 94

requisiti per la Pubblicazione Web personalizzata 21

requisiti server 64

ricerca composta  
 comando query 49  
 parametro query 56

richieste di dati XML 25

riepilogo dei passi per accedere ai dati XML 25

risoluzione dei problemi  
 accesso a documenti XML 41  
 siti Web Pubblicazione Web personalizzata 92

ruolo del browser Web nelle richieste XML 24

## S

SAT. *Vedere* FileMakerServer Admin Console

script  
 account e privilegi 18  
 nella Pubblicazione Web personalizzata 18  
 suggerimenti e considerazioni 18  
 utilizzo in PHP 75  
 utilizzo nelle richieste XML 25

script disponibili 51

script Modifica password 14

Server Admin di macOS 65

server Web  
 file registro 94  
 ruolo nelle richieste XML 24  
 supporto tipo MIME 15

set di privilegi, assegnazione per la Pubblicazione Web personalizzata 13

sicurezza  
 account e password 14  
 documentazione 11  
 limitazione dell'accesso dagli indirizzi IP 14  
 linee guida per la protezione dei database pubblicati 14

sintassi URL per  
 oggetti Contenitore in soluzioni XML 26  
 richieste XML 25

siti Web  
 creazione con Motore per la Pubblicazione Web 20  
 monitoraggio 94  
 pagine di supporto per FileMaker 8  
 test 92

stringhe di query  
 aggiunta dei record a un portale 45  
 campi globali, sintassi 47  
 comandi e parametri 38, 42  
 linee guida per 43  
 modifica dei record nei portali 46  
 nome di campo completamente qualificato, sintassi 45  
 richiesta dati XML 38, 42

strumento Server Admin. *Vedere* Server Admin di macOS



**T**

- test
  - output XML 93
  - siti Web 92
- tipi MIME (Multipurpose Internet Mail Extensions) 15
- Tomcat, utilizzo dei file registro 97
- trigger di script 19

**U**

- UAC. *Vedere* FileMakerServer Admin Console
- Unicode
  - caratteri utilizzati nei parser XML 38
  - formato data restituito da FileMakerServer 69
- utenti Web
  - accesso ai database protetti 13
  - requisiti per accedere alle soluzioni di Pubblicazione Web personalizzata 22
  - utilizzo dei dati dei campi Contenitore 17

**V**

- vantaggi PHP 12
- verifica mediante calcolo 87
- Versione di PHP 65

**W**

- Web Publishing Core, illustrazione 24

**X**

- XML
  - attivazione in un database 13
  - codifica nel formato UTF-8 28, 38
  - codifica testo URL 27
  - DTD (Document Type Definition) 29, 33
  - fmresultset grammar
    - <datasource> element 30
  - generazione di dati XML da una richiesta 24
  - grammatica FMPXMLLAYOUT 35
  - grammatica FMPXMLRESULT 33
  - grammatica fmresultset 30
    - elemento <field-definition> 31
    - elemento <metadata> 31
    - elemento <relatedset-definition> 31
    - elemento <resultset> 31
  - grammatiche a confronto 28
  - namespace per 28
  - ordine di elaborazione delle richieste 41
  - parser 25, 38
  - Pubblicazione Web personalizzata, descrizione 11
  - richiesta di dati 25
  - richiesta, determinazione del formato 40
  - riepilogo dei passi per accedere ai dati XML 25
  - risoluzione dei problemi di accesso a documenti XML 41
  - risposta, applicazione del formato 40
  - specifica XML 1.0 23
  - stringhe di query 38, 42
  - vantaggi 11

- XML (Extensible Markup Language). *Vedere* XML
- <xsl:stylesheet> element 93
- <xsl:template> element 93