

FileMaker® Server 13

Anpassad webbpublicering
med PHP



© 2007-2013 FileMaker, Inc. Med ensamrätt.

FileMaker, Inc.

5201 Patrick Henry Drive

Santa Clara, Kalifornien 95054, USA

FileMaker och Bento är varumärken som tillhör FileMaker, Inc. och är registrerade i USA och andra länder.

Filmappslogotypen, FileMaker WebDirect och Bento-logotypen är varumärken som tillhör FileMaker, Inc. Alla övriga varumärken tillhör respektive ägare.

FileMakers dokumentation skyddas av lagen om upphovsrätt. Det är därför inte tillåtet att mångfaldiga eller distribuera detta dokument utan FileMakers skriftliga medgivande. Dokumentationen får endast användas tillsammans med ett licensierat exemplar av FileMaker-programmet.

Samtliga personer, företag, e-postadresser och URL-adresser som förekommer i exempel är fiktiva och eventuella likheter med verkliga personer, företag, e-postadresser eller URL-adresser är fullständigt oavsiktliga. En lista över medverkande visas i dokumenten Tillskrivningar som medföljer den här programvaran. Omnämning av tredjepartsprodukter och URL-adresser förekommer endast i informationssyfte och ska inte tolkas som förslag eller rekommendationer. FileMaker, Inc. tar inget ansvar när det gäller prestandan hos dessa produkter.

Mer information finns på webbplatsen <http://www.filemaker.com/se/>.

Utgåva: 01

Innehåll

Förord	6
Om denna handbok	6
Kapitel 1	
<i>Introduktion till anpassad webbpublicering</i>	7
Om Web Publishing Engine	8
Hur en webbpublicerings-förfrågan behandlas	8
Anpassad webbpublicering med PHP	9
Anpassad webbpublicering med XML	9
Jämförelse mellan PHP och XML	9
Anledningar till att välja PHP	9
Anledningar till att välja XML	9
Kapitel 2	
<i>Anpassad webbpublicering med PHP</i>	11
Viktiga funktioner i anpassad webbpublicering med PHP	11
Krav för anpassad webbpublicering	11
Vad som krävs för att publicera en databas med anpassad webbpublicering	11
Vad webbanvändarna behöver för att få åtkomst till en anpassad webbpubliceringslösning	12
Om att ansluta till Internet eller ett intranät	12
Installera FileMaker Api för PHP manuellt	13
Så går du vidare	14
Kapitel 3	
<i>Förbereda databaser för anpassad webbpublicering</i>	15
Aktivera anpassad webbpublicering med PHP för databaser	15
Skapa layouter för anpassad webbpublicering med PHP	15
Skydda publicerade databaser	16
Få åtkomst till en skyddad databas	17
Publicera innehållet i containerfält på webben	18
Containerfält som bäddas in i en databas	18
Containerfält med externa filer	19
Containerfält med externt lagrade data	19
Hur webbanvändare kan visa containerfältobjekt	22
FileMaker-manus och anpassad webbpublicering	22
Tips och råd om manus	22
Funktionssätt i egna webbpubliceringslösningar	24
Manustriggers i egna webbpubliceringslösningar	24
Kapitel 4	
<i>Översikt över anpassad webbpublicering med PHP</i>	25
Hur Web Publishing Engine fungerar med PHP-lösningar	25
Allmänna steg för anpassad webbpublicering med PHP	25

Kapitel 5

Använda FileMaker Api för PHP	27
Mer information	27
Referens för FileMaker Api för PHP	27
Introduktion till FileMaker Api för PHP	28
Exempel till FileMaker Api för PHP	28
Använda FileMaker class	28
FileMaker class-objekt	28
FileMaker kommandoobjekt	29
Ansluta till en FileMaker-databas	29
Arbeta med poster	30
Skapa en post	30
Duplicera en post	30
Redigera en post	30
Radera en post	31
Köra FileMaker-manus	31
Hämta en lista på tillgängliga manus	31
Köra ett FileMaker-manus	32
Köra ett manus innan ett kommando utförs	32
Köra ett manus innan en resultatuppsättning sorteras	32
Köra ett manus efter resultatuppsättningen har genererats	32
Ordning för att utföra manus	33
Arbeta med FileMaker-layouter	33
Använda portaler	34
Lista de portaler som har definierats på en specifik layout	34
Hämta portalnamn för ett specifikt resultatobjekt	35
Hämta information om portaler för en specifik layout	35
Hämta information för en specifik portal	35
Hämta tabellnamnet för en portal	35
Hämta portalposterna för en specifik post	35
Skapa en ny post i en portal	36
Ta bort en post från en portal	36
Använda värdelistor	36
Hämta namnen på alla värdelistor för en specifik layout	36
Hämta en matris över alla värdelistor för en specifik layout	37
Hämta värdena för en namngiven värdelista	37
Utföra sökningar	38
Använda kommandot Visa alla	39
Använda kommandot Sök efter alla	39
Använda kommandot Sök	39
Använda kommandot Sammanfattad sökning	40
Behandla posterna i ett sökresultat	42
Filtrera portalrader som returnerats av en sökförfrågan	43
Förkontrollera kommandon, poster och fält	43
Förkontrollera poster i ett kommando	44
Förkontrollera poster	45
Förkontrollera fält	45
Behandla kontrollfel	45
Hantera fel	47

Kapitel 6

Iscensätta, testa och övervaka en webbplats

Iscensätta en anpassad webbpubliceringsplats	48
Testa en anpassad webbpubliceringsplats	49
Övervaka webbplatsen	50
Använda webbserveråtkomst och felloggar	50
Använda Web Publishing Engine-loggen	50
Använda felloggen för webbservern	52
Använda Tomcat-loggar	52
Felsöka webbplatsen	53

Bilaga A

Felkoder för anpassad webbpublicering med PHP

Felkodsnummer för FileMaker-databaser	54
Felkodsnummer för PHP-komponenter	61

Index

62

Förord

Om denna handbok

Den här handboken riktar sig till dig som känner till hur PHP fungerar, som kan utveckla webbplatser och skapa databaser med hjälp av FileMaker® Pro. Du bör förstå grunderna i databasdesign i FileMaker Pro, och förstå begreppen fält, relationer, layouter, portaler och containrar. Information om FileMaker Pro finns i FileMaker Pro Hjälp.

Handboken innehåller följande information om anpassad webbpublicering med PHP på FileMaker Server:

- vad som krävs för att du ska kunna utveckla en anpassad webbpubliceringslösning med PHP
- hur man publicerar databaser med PHP
- vad webbanvändare behöver för att få åtkomst till en anpassad webbpubliceringslösning
- hur man använder FileMaker Api för PHP för att hämta data från databaser som FileMaker Server är värd för.

Viktigt! Du kan hämta dokumentationen för FileMaker i PDF-format från <http://www.filemaker.com/se/support/index.html>. Uppdateringar till det här dokumentet kan också hämtas från den webbplatsen.

Dokumentationen för FileMaker Server innehåller följande information:

Information om	Se
Installera och konfigurera FileMaker Server	<i>FileMaker Server Startbok</i> <i>FileMaker Server - Hjälp</i>
Göra layouter från FileMaker Pro- och FileMaker Pro Advanced-databaser tillgängliga för användare med webbläsare i ett intranät eller på Internet	<i>FileMaker WebDirect™ Handbok</i>
Anpassad webbpublicering med PHP	<i>FileMaker Server Anpassad webbpublicering med PHP</i> (denna bok)
Anpassad webbpublicering med XML	<i>FileMaker Server Anpassad webbpublicering med XML</i>
Installera och konfigurera drivrutiner för ODBC och JDBC och använda ODBC och JDBC	<i>Handbok för FileMaker ODBC och JDBC</i>
SQL-uttryck och standarder som stöds av FileMaker-programvaran	<i>FileMaker SQL-referens</i>

Kapitel 1

Introduktion till anpassad webbpublicering

Med FileMaker Server kan du publicera din FileMaker-databas på Internet eller ett intranät på följande sätt.

FileMaker WebDirect: Med FileMaker WebDirect publicerar du layouter från en databas snabbt och smidigt på webben. Du behöver inte installera ytterligare program – med en kompatibel webbläsare och tillgång till Internet eller ett intranät kan webbanvändare ansluta sig till din FileMaker WebDirect-lösning och visa, redigera, sortera och söka i poster om du ger dem behörighet.

Med FileMaker WebDirect måste värddatorn köra FileMaker Server. Användargränssnittet liknar FileMaker Pro-programmet. Webbsidorna och formulär som webbanvändaren samverkar med är beroende på de layouter och vyer som definierats i FileMaker Pro-databasen. Mer information finns i *FileMaker WebDirect Handbok*.

Statisk publicering: Om du ändrar dina data väldigt sällan eller om du inte vill att användarna ska ha en direktanslutning till din databas kan du använda dig av statisk publicering. Vid statisk publicering exporterar du data från en FileMaker Pro-databas och skapar en webbsida som du kan anpassa ytterligare med HTML. Webbsidan ändras inte när informationen i din databas ändras, och användarna öppnar inte din databas. (Med FileMaker WebDirect uppdateras data i webbläsaren när dessa data uppdateras i databasen.) Mer information finns i direkthjälpen för FileMaker Pro.

Anpassad webbpublicering: Du kan integrera en FileMaker-databas med en anpassad webbplats genom att använda teknikerna för anpassad webbpublicering som finns tillgängliga i FileMaker Server. FileMaker Server, som är värd för de publicerade databaserna, kräver inte att FileMaker Pro är installerat eller igång för att anpassad webbpublicering ska vara tillgänglig.

Med Anpassad webbpublicering kan du:

- integrera databasen med en annan webbplats
- avgöra hur användarna ska få tillgång till data
- kontrollera hur data ska visas i webbläsare.

FileMaker Server innehåller två tekniker för anpassad webbpublicering:

- Anpassad webbpublicering med PHP: Använd FileMaker Api för PHP, med ett objektorienterat PHP-gränssnitt för FileMaker Pro-databaser, för att integrera dina FileMaker-data i ett PHP-webbprogram. Eftersom du kodar PHP-webbsidorna själv har du full kontroll över användargränssnittet och användarens möjligheter att interagera med alla data.
- Anpassad webbpublicering med XML: Använd XML-datapublicering för att utbyta FileMaker-data med andra webbplatser och program. Genom att använda HTTP URL-förfrågningar tillsammans med FileMaker-frågekommandon och tillhörande frågeparametrar kan du ställa frågor till en databas som drivs med FileMaker Server, hämta resultatet i XML-format och använda dessa data på valfritt sätt.

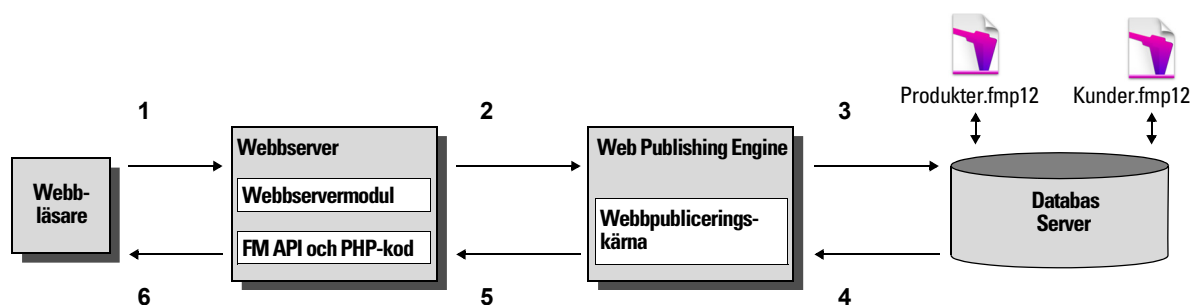
Om Web Publishing Engine

För att stödja FileMaker WebDirect och anpassad webbpublicering använder FileMaker Server en uppsättning programvarukomponenter som kallas för *FileMaker Server Web Publishing Engine*. Web Publishing Engine hanterar samspillet mellan en webbanvändares webbläsare, din webserver och FileMaker Server.

Anpassad webbpublicering med XML: Webbanvändare får åtkomst till din anpassade webbpubliceringslösning genom att klicka på en HREF-länk eller skriva en URL-adress till webbservern och en FileMaker-frågesträng. Web Publishing Engine returnerar de XML-data som anges i frågesträngen.

Anpassad webbpublicering med PHP: När en webbanvändare ansluter till din anpassade webbpubliceringslösning ansluter PHP på FileMaker Server till Web Publishing Engine och svarar genom FileMaker API för PHP.

Använda FileMaker Servers Web Publishing Engine för anpassad webbpublicering



Hur en webbpubliceringsförfrågan behandlas

1. En förfrågan skickas från en webbläsare eller ett program till webbservern.
2. Webbservern vidarebefordrar förfrågan genom FileMakers webbservermodul till Web Publishing Engine.
3. Web Publishing Engine efterfrågar data på databasen som databasservern är värd för.
4. FileMaker Server skickar efterfrågade FileMaker-data till Web Publishing Engine.
5. Web Publishing Engine konverterar FileMaker-data för att svara på förfrågan.
 - För PHP-förfrågningar svarar Web Publishing Engine på API-förfrågan.
 - För XML-förfrågningar skickar Web Publishing Engine XML-data direkt till webbservern.
6. Webbservern skickar utdata tillbaka till webbläsaren eller programmet.

Viktigt! Säkerheten är mycket viktig när du publicerar information på webben. Läs igenom säkerhetsriktlinjerna i *FileMaker Pro Handbok*, som kan hämtas som PDF-fil på <http://www.filemaker.com/se/support/index.html>.

Anpassad webbpublicering med PHP

FileMaker Api för PHP ger ett objektorienterat PHP-gränssnitt för FileMaker-databaser. FileMaker Api för PHP gör att man kan få åtkomst till både uppgifter och logik som har lagrats i en FileMaker Pro-databas och publicera dem på Internet, eller exportera dem till andra program. API stöder också komplexa och sammansatta sökkommandon för att extrahera och filtrera data som har lagrats i FileMaker Pro-databaser.

PHP utformades ursprungligen som ett procedurmässigt programmeringsspråk, och har utvecklats som ett objektorienterat webbutvecklingsspråk. PHP ger programspråksfunktionalitet för att skapa praktiskt taget alla typer av logik på en webbsida. Till exempel kan du använda villkorliga logikkonstruktioner för att kontrollera sidgenerering, datarouting eller arbetsflöde. PHP ger även möjlighet till webbplatsadministration och säkerhet.

Anpassad webbpublicering med XML

FileMaker anpassad webbpublicering med XML gör att du kan skicka förfrågningar till en FileMaker Pro-databas som FileMaker Server är värd för, och visa, ändra eller hantera resulterande data. Med en HTTP-förfrågan som innehåller rätt frågekommandon och -parametrar kan du hämta FileMaker-data som ett XML-dokument. Därefter kan du exportera dina XML-data till andra program.

Jämförelse mellan PHP och XML

Följande avsnitt ger riktlinjer för hur du avgör vilken lösning som passar bäst för din webbplats.

Anledningar till att välja PHP

- PHP är ett mer kraftfullt, objektorienterat procedurmässigt manusspråk och är relativt enkelt att lära sig. Det finns många tillgängliga resurser för utbildning, utveckling och support.
- FileMaker Api för PHP gör att man kan få åtkomst till uppgifter och logik som har lagrats i en FileMaker Pro-databas och publicera dem på Internet, eller exportera dem till andra program.
- Med PHP kan man använda villkorslogik för att kontrollera sidkonstruktionen eller flödet.
- PHP ger programspråksfunktionalitet för att skapa många typer av logik på en webbsida.
- PHP är ett av de populäraste webbmanusspråken.
- PHP är ett språk med öppen källkod, som finns på <http://php.net>.
- PHP ger åtkomst till ett stort utbud av komponenter från tredje part som du kan integrera i dina lösningar.

Anledningar till att välja XML

- FileMaker XML frågeparametersyntax är utformat för interaktion mellan databaser, vilket underlättar lösningsutveckling.
- XML är en W3C-standard.
- XML är ett format som kan läsas av både datorer och människor och som stöder Unicode, vilket gör att data kan kommuniceras på alla skriftliga språk.
- XML passar bra för att presentera poster, listor och trädstrukturerad data.

- Du kan använda FMPXMLRESULT för att komma åt XML-data med Anpassad webbpublicering och för XML-export från FileMaker Pro-databaser.

Obs! Mer information om Anpassad webbpublicering med XML finns i *FileMaker Server Anpassad webbpublicering med XML*.

Kapitel 2

Anpassad webbpublicering med PHP

Med anpassad webbpublicering med PHP kan du använda programmeringsspråket PHP för att integrera data från FileMaker-databaser i dina egna webbplatslayouter. Med anpassad webbpublicering med PHP får du FileMaker Api för PHP, som är en PHP-klass som har skapats av FileMaker som ger åtkomst till databaser som FileMaker Server är värd för. Denna PHP-klass ansluter till FileMaker Servers Web Publishing Engine och gör data tillgängliga för din webbservers PHP-motor.

Viktiga funktioner i anpassad webbpublicering med PHP

- Skapa webbapplikationer som använder programmeringsspråket PHP med öppen källkod. Använd den version av PHP 5 som stöds av FileMaker Server, eller använd din egen version av PHP 5. (Om du vill att använda din egen version av PHP, se "Installera FileMaker Api för PHP manuellt" på sidan 13.)
- Databaser på FileMaker Server FileMaker Pro krävs inte för anpassad webbpublicering eftersom FileMaker Server är värd för databaserna.
- Skriv PHP-kod som kan skapa, radera, redigera och duplicera poster i en delad FileMaker-databas. Din kod kan utföra kontroll av fält och poster innan ändringar verkställs i den delade databasen.
- Skriv PHP-kod som ger åtkomst till layouter, portaler, värdelistor och relaterade fält. Precis som i FileMaker Pro är åtkomsten till data, layouter och fält baserad på användarkontots inställningar som anges i databasens behörighetsinställningar. Web Publishing Engine har också flera andra säkerhetsförbättringar. Se "Skydda publicerade databaser" på sidan 16.
- Skriv PHP-kod som utför komplexa manus med flera steg. FileMaker stöder över 65 manussteg för anpassad webbpublicering. Se "FileMaker-manus och anpassad webbpublicering" på sidan 22.
- Skriv PHP-kod som utför komplexa sökningar.

Krav för anpassad webbpublicering

Detta avsnitt förklarar vad som krävs för att utveckla en anpassad webbpubliceringslösning med hjälp av PHP, vad webbanvändarna behöver för att få åtkomst till en anpassad webbpubliceringslösning och vilken påverkan det kan ha på din server att vara värd för en webbpubliceringslösning.

Vad som krävs för att publicera en databas med anpassad webbpublicering

Om du ska publicera databaser med anpassad webbpublicering med PHP, behöver du följande:

- en FileMaker Server-lösning, som innehåller tre komponenter.
 - en webbserver, antingen Microsoft IIS (Windows) eller Apache (OS X). FileMakers webbservermotul har installerats på webbservern.
 - Web Publishing Engine för FileMaker
 - databasserver för FileMaker

- PHP installerat på webbservern. FileMaker Server kan installera den version av PHP 5 som stöds, du kan även använda din anpassade version. Det lägsta kravet för PHP på OS X är PHP 5.3.15. Det lägsta kravet för PHP på Windows är PHP 5.3.27. Information om PHP finns på <http://php.net>. Den version av PHP som är installerad på webbservern måste stödja cURL-funktioner (client URL library). Information om cURL finns på <http://php.net/curl>.

Viktigt! När du installerar den version av PHP 5 som stöds av FileMaker Server visas det inte i verktyget Mac OS X Server Admin; den ska inte finnas med på listan. Om du använder verktyget OS X Server Admin för att aktivera PHP inaktiverar du den version av PHP 5 som stöds av FileMaker, och aktiverar din anpassade version av PHP.

- en eller fler FileMaker Pro-databaser på FileMaker Server
- IP-adressen eller domännamnet på värden där webbservern körs
- en webbläsare och åtkomst till webbservern så att du kan utveckla och testa den egna webbpubliceringslösningen

Mer information finns i *FileMaker Server startbok*.

Vad webbanvändarna behöver för att få åtkomst till en anpassad webbpubliceringslösning

Webbanvändare som vill ha åtkomst till en anpassad webbpubliceringslösning som använder PHP behöver följande:

- en webbläsare
- åtkomst till Internet eller ett intranät och webbservern
- IP-adressen eller domännamnet på värden där webbservern körs

Om databasen är lösenordsskyddad måste webbanvändare även uppge ett användarnamn och ett lösenord för ett databaskonto

Om att ansluta till Internet eller ett intranät

När du publicerar databaser på Internet eller ett intranät måste värddatorn köra FileMaker Server och de databaser du vill dela måste köras och vara tillgängliga. Dessutom gäller följande:

- Publicera din databas på en dator med fast Internet- eller intranätanslutning. Det går att publicera databaser utan att ha en fast anslutning, men dessa blir bara tillgängliga för webbanvändarna när din dator är ansluten till Internet eller ett intranät.
- Värddatorn för den webbserver som är en del av FileMaker Server-driftsättningen måste ha en dedicerad statisk (permanent) IP-adress eller ett domännamn. Om du ansluter till Internet via en Internetleverantör (ISP) kan din IP-adress vara dynamiskt allokerad (dvs. olika varje gång du ansluter). Med en dynamisk IP-adress blir det svårare för webbanvändarna att hitta dina databaser. Om du inte är säker på vilken typ av anslutning du ska välja kontaktar du din Internetleverantör eller nätverksadministratör.

Installera FileMaker Api för PHP manuellt

När du installerar FileMaker Server har du möjlighet att installera den version av PHP som stöds av FileMaker (PHP 5). Om du redan har en PHP-motor installerad och konfigurerad och du endast vill lägga till FileMaker Api för PHP kan du installera FileMaker Api för PHP-klass manuellt för att göra det tillgängligt för dina PHP-manus.

Om du inte har installerat den version av PHP som stöds av FileMaker, ska du se till att göra följande konfigureringsåtgärder i din version av PHP-motorn:

- Aktivera cURL-modulen i php.ini.
- Ange platsen för FileMaker API för PHP i variabeln `include_path` i php.ini.
- Om du använder en databas som innehåller datum och tidpunkter ska du installera paketet "pear date". Mer information finns på: <http://pear.php.net/package/date/>

Obs! FileMaker Server har testats med PHP version 5.3.15 för OS X 10.8, med PHP version 5.4.17 för OS X 10.9 och med PHP version 5.3.27 för Windows. Systemet fungerar bäst om du använder de rekommenderade versionerna av PHP.

Att göra FileMaker Api för PHP tillgängligt för dina PHP-manus

När du installerade FileMaker Server följde FileMaker Api för PHP-paketet med som en .zip-fil på följande plats:

- För IIS (Windows):
`[enhet] : \Program\FileMaker\FileMaker Server\Web Publishing\FM_API_for_PHP_Standalone.zip`
där [enhet] är den enhet där webbserverkomponenten för din FileMaker Server-driftsättning finns.
- För Apache (OS X):
`/Bibliotek/FileMaker Server/Web Publishing/FM_API_for_PHP_Standalone.zip`

Filen `FM_API_for_PHP_Standalone.zip` innehåller filen `FileMaker.php` och mappen `FileMaker`. Packa upp filen och kopiera filen `FileMaker.php` och mappen `FileMaker` till någon av dessa platser:

- mappen där dina PHP-manus finns.
 - För IIS (Windows) via HTTP eller HTTPS:
`[enhet] : \Program\FileMaker\FileMaker Server\HTTPServer\Conf`
där [enhet] är den enhet där Web Publishing Engine-komponenten för din FileMaker Server-driftsättning finns.
 - För Apache (OS X) via HTTP: `/Bibliotek/FileMaker Server/HTTPServer/htdocs`
 - För Apache (OS X) via HTTPS:
`/Bibliotek/FileMaker Server/HTTPServer/htdocs/httpsRoot`
- en av `include_path`-katalogerna i din PHP-installation. Standardplatsen för Mac OS X är `/usr/lib/php`.

Så går du vidare

Här följer några förslag för att komma igång med att utveckla egna webbpubliceringslösningar:

- Använd FileMaker Server Admin Console för att aktivera anpassad webbpublicering. Se FileMaker Server Hjälpsida och *FileMaker Server Startbok*.
- I FileMaker Pro öppnar du varje FileMaker-databas som du vill publicera och ser till att databasen har rätt utökad behörighet aktiverad för anpassad webbpublicering. Se "Aktivera anpassad webbpublicering med PHP för databaser" på sidan 15.
- Mer information om hur man får åtkomst till data i FileMaker-databaser med FileMaker Api för PHP finns i kapitel 5, "Använda FileMaker Api för PHP".

Kapitel 3

Förbereda databaser för anpassad webbpublicering

Innan du kan använda anpassad webbpublicering med en databas måste du förbereda databasen och skydda den från obehörig åtkomst.

Aktivera anpassad webbpublicering med PHP för databaser

Du måste aktivera anpassad webbpublicering med PHP i varje databas som du vill publicera. Annars kan webbanvändare inte använda anpassad webbpublicering för att få åtkomst till databasen, även om en FileMaker Server som är konfigurerad för att stödja en Web Publishing Engine är värd för den.

Så här aktiverar du anpassad webbpublicering för en databas:

1. Starta FileMaker Pro och öppna databasen som du vill publicera med hjälp av ett konto som har behörighetsuppsättningen Full åtkomst eller Hantera utökad behörighet.
2. Tilldela utökade **fmphp**-behörigheter till en eller flera behörighetsuppsättningar för att möjliggöra anpassad webbpublicering med PHP.
3. Tilldela behörighetsuppsättningarna med utökad behörighet för anpassad webbpublicering för lämpliga konton (till exempel Admin- och gästkonton).

Viktigt! När du definierar kontonamn och lösenord för lösningar för anpassad webbpublicering använder du utskriftbara ASCII-tecken, till exempel, **a-z**, **A-Z** och **0-9**. För att få säkrare kontonamn och lösenord ska du ta med några icke alfanumeriska tecken, som ett utropstecken (!) eller procenttecken (%). Kolon (:) är inte tillåtet. Information om hur du konfigurerar konton finns i FileMaker Pro Hjälp.

4. När du använder FileMaker Server Admin Console ska du kontrollera att värden för databasen är korrekt konfigurerad, och att den är tillgänglig för FileMaker Server. Instruktioner finns i FileMaker Server Hjälp.

Obs! Eftersom anpassad webbpublicering med PHP inte använder beständiga databassessioner, kan referenser till en extern ODBC-datakälla i FileMaker Pro-relationsdiagram begränsa de funktioner som är tillgängliga i din PHP-lösning. Om din databas har åtkomst till data från en extern SQL-datakälla kan det hända att du inte kan uppdatera den externa tabellens postdata.

Skapa layouter för anpassad webbpublicering med PHP

Anpassad webbpublicering med PHP ger inte direkt tabellåtkomst till data i en FileMaker Pro-databas, utan använder de layouter som har definierats i databasen. Trots att det inte är något krav på att skapa en unik layout för anpassad webbpublicering med PHP kan det av flera skäl vara praktiskt att skapa en layout speciellt för en PHP-lösning:

- Du förbättrar prestandan genom att skapa en layout som är begränsad till de fält, etiketter, beräkningar och portaler som du behöver ta med i PHP-lösningen.
- Du förenklar din PHP-kod genom att utföra mindre databehandling eftersom posterna har färre fält.
- Du separerar utformningen av gränssnittet från dina data, så att du kan skräddarsy gränssnittet för webbanvändarna.

Skydda publicerade databaser

Med anpassad webbpublicering med PHP kan du begränsa åtkomsten till dina publicerade databaser. Du kan använda dessa metoder:

- Kräv lösenord för databaskonton som används för anpassad webbpublicering med PHP.
- Aktivera bara utökad behörighet i anpassad webbpublicering med PHP för de behörighetsuppsättningar som du vill ge åtkomst.
- Inaktivera anpassad webbpublicering med PHP för en specifik databas genom att avmarkera utökad fmphp-behörighet för alla behörighetsuppsättningar i den databasen. Se *FileMaker Pro Hjälp*.
- Aktivera eller inaktivera Anpassad webbpublicering för alla Lösningar för anpassad webbpublicering med FileMaker Server Admin Console. Se *FileMaker Server startbok* och *FileMaker Server Hjälp*.
- Konfigurera webbservern för att begränsa vilka IP-adresser som ska kunna använda databasen via Web Publishing Engine. Du kan exempelvis ange att endast webbanvändare från IP-adress 192.168.100.101 ska ha tillgång till databaserna. Information om hur du begränsar åtkomst från IP-adresser finns i dokumentationen till webbservern.

FileMaker Server stöder kryptering för data som sparas på disk och för data som överförs till klienter.

- Kryptera databasen med funktionen Databaskryptering i FileMaker Pro Advanced. Krypteringen skyddar FileMaker-databasfilen och alla temporära filer som sparas på disk. Mer information om hur du krypterar en databas finns i *FileMaker Pro Handbok*, *FileMaker Server Startbok* och i direkthjälpen för FileMaker Pro.
 - En krypterad databas som FileMaker Server är värd för öppnas med hjälp av Admin Console eller kommandoradsgränssnittet (CLI). Som FileMaker Server-administratör öppnar du filen med krypteringslösenordet, så att FileMaker-klienter kan använda den krypterade databasen.
 - När FileMaker Server-administratören har öppnat den krypterade FileMaker-databasen med krypteringslösenordet, behöver FileMaker-klienterna inte ange krypteringslösenordet för att komma åt den krypterade databasen. Mer information om hur du öppnar en krypterad databas finns i direkthjälpen för FileMaker Pro Server.
- Använd SSL-kryptering (Secure Sockets Layer) för kommunikationen mellan webbservern och webbläsaren. Vid SSL-kryptering konverteras informationen som utväxlas mellan webbservern och klienter till obegriplig information med matematiska formler, så kallade *chiffer*. Dessa chiffer används när informationen konverteras tillbaka till begripliga data med hjälp av krypteringsnycklar. SSL-anslutningar nås genom en HTTPS-anslutning. Inga åtgärder krävs av klienten när den är konfigurerad och fungerar som den ska. Information om hur du aktiverar, konfigurerar och underhåller SSL-anslutningarna finns i dokumentationen till webbservern.

Mer information om hur du skyddar din databas finns i *FileMaker Pro Handbok* som finns som PDF-fil på <http://www.filemaker.com/se/support/index.html>.

Få åtkomst till en skyddad databas

Anpassad webbpublicering med PHP gör det möjligt för dig att begränsa åtkomsten till publicerade databaser genom lösenordsskydd, databaskryptering och säkra anslutningar. När en webbanvändare får åtkomst till en databas med en PHP-lösning måste PHP-koden uppge identifieringsuppgifterna till databasen med FileMaker Api för PHP. Om gästkontot för databasen är inaktiverat, eller inte har aktiverat utökad **fmphp**-behörighet returnerar FileMaker Api för PHP ett felmeddelande och din PHP-kod måste uppge inloggningsuppgifter för användaren.

I introduktionen till FileMaker Api för PHP finns ett exempel som visar hur man använder metoden `setProperty()` för att ställa in användarnamn och lösenord för en skyddad databas. Se "Introduktion till FileMaker Api för PHP" på sidan 28.

I följande lista sammanfattas processen som sätts igång när man använder anpassad webbpublicering för att få åtkomst till en lösenordsskyddad databas:

- Om inget lösenord har tilldelats för ett konto med aktiverad anpassad webbpublicering, behöver PHP-lösningen bara uppge kontonamnet.
- Om gästkontot är inaktiverat måste PHP-lösningen uppge ett kontonamn och ett lösenord. PHP-lösningen kan antingen uppmana webbanvändaren att ange kontonamnet och lösenordet, eller så kan det lagras kontonamnet och lösenordet i PHP-koden. Kontonamnet måste ha den utökade behörigheten **fmphp** aktiverad.
- Om gästkontot är aktiverat och har den utökade behörigheten **fmphp** aktiverad:
 - PHP-lösningen behöver inte uppmana webbanvändarna att ange kontonamn och lösenord när en fil öppnas. Alla webbanvändare loggas automatiskt in med gästkontot och får den behörighet som hör till gästkontot.
 - Standardbehörigheten för gästkonton ger endast läsbehörighet. Du kan ändra standardbehörigheten, men även den utökade behörigheten, för detta konto. Se FileMaker Pro Hjälp.
- PHP-lösning kan använda manussteget Logga in igen för att tillåta användare att logga in med ett annat konto (till exempel för att växla från gästkontot till ett konto med större behörighet). Se FileMaker Pro Hjälp. Eftersom PHP-anslutningar inte använder bestående databassessioner måste PHP-lösningen lagras kontonamnet och lösenordet för att använda dem för varje kommande förfrågan.

Obs! Som standardinställning kan webbanvändare inte ändra sina egna kontolösenord från en webbläsare. Denna funktion kan aktiveras för en databas med hjälp av manussteget Ändra lösenord, som gör att webbanvändarna kan ändra sina lösenord från webbläsaren. Se FileMaker Pro Hjälp.

Publicera innehållet i containerfält på webben

Innehållet i ett containerfält kan bäddas in i databasen med en länk som anges med en relativ sökväg, men det kan också lagras externt.

Containerfält som bäddas in i en databas

Om ett containerfält lagrar de faktiska filerna i FileMaker-databasen följer du stegen nedan om du vill använda containerfältobjekten i en PHP-lösning:

- Använd FileMaker API för PHP för att definiera databasobjektet (`$fm`) med de rätta identifieringsuppgifterna (kontonamn och lösenord).

```
$fm = new FileMaker();
$fm->setProperty('database', $databaseName);
$fm->setProperty('username', $userName);
$fm->setProperty('password', $password);
```

- Använd korrekta HTML-taggar för att ange vilken typ av webbkompatibelt objekt som containerfältet innehåller, och skapa en URL-sträng som anger sökvägen till HTML-taggens källattribut.

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

- Använd metoden `getContainerData()` för att hämta containerfältobjektet.

```
echo $fm->getContainerData($_GET['-url']);
```

I introduktionen till FileMaker Api för PHP finns flera exempel på hur man använder containerfält. Se "Introduktion till FileMaker Api för PHP" på sidan 28.

Kommentar

- Web Publishing Engine stöder progressiv nedladdning av ljudfiler (.mp3), videofiler (.mov, .mp4, och .avi rekommenderas), och PDF filer for interaktiva containrar. En webbanvändare kan till exempel börja titta på en film även om inte hela filmfilen har laddats ner ännu. Om du vill möjliggöra progressiv nedladdning kan du behöva skapa filer genom att använda alternativ som stöder direktuppspelning eller som optimerar för visning på webben. Skapa till exempel PDF-filer genom att använda alternativet för webbvisningsoptimering.
- Om alternativet **Aktivera säkra anslutningar** i FileMaker Server inte är valt, kommer anslutningarna som FileMaker Server använder för att överföra data inte vara krypterade under överföringen.
 - FileMaker-klienter ser interaktiva containerdata med liten fördröjning.
 - FileMaker Server dekrypterar containerfältdata till en cachemapp på servern när en FileMaker Pro-, FileMaker Go- eller webbklient skickar en förfrågan om data. Dessa data kan vara dekrypterade i cachemappen på servern i två timmar innan FileMaker Server regelbundet tömmer cachemappen. Dessa data cachelagras inte lokalt på klienten.
- När alternativet **Aktivera säkra anslutningar** i FileMaker Server är valt, kommer FileMaker Server använda säkra anslutningar för att överföra data. FileMaker-klienter hämtar containerdata fullständigt innan användaren kan interagera med dem. Data är lika säkra som om lösningen var en lokal databas, eftersom det inte skapas några temporära cachefiler och data krypteras under överföring.

Databasservern måste stoppas och startas om när inställningen **Aktivera säkra anslutningar** ändras för att den nya inställningen ska gälla.

Containerfält med externa filer

Om ett containerfält lagrar en filreferens kan du använda metoden `getContainerData()` för att hämta containerfältobjekten från databasen i din PHP-kod, men du kan också använda metoden `getContainerDataURL()` för att hämta en fullständig URL för containerfältobjektet.

Du måste dessutom följa dessa steg om du ska publicera de refererade filerna med Web Publishing Engine:

1. Lagra containerobjektfilerna i webbmappen inuti FileMaker Pro-mappen.
2. Starta FileMaker Pro, sätt in objekten i containerfältet och välj **Spara endast länk till filen**.
3. Kopiera eller flytta objektfilerna som refereras till i webbmappen till samma relativa sökväg i följande mapp i webbservern.
 - För IIS (Windows) via HTTP eller HTTPS:
[enhet] : \Program\FileMaker\FileMaker Server\HTTPServer\Conf där [enhet] är den enhet där FileMaker Server-distributionens komponent Web Publishing Engine ligger.
 - För Apache (OS X) via HTTP:
/Bibliotek/FileMaker Server/HTTPServer/htdocs
 - För Apache (OS X) via HTTP:
/Bibliotek/FileMaker Server/HTTPServer/htdocs/httpsRoot

Kommentar

- För att containerobjekt ska kunna lagras som filreferenser måste webbservern vara konfigurerad för att kunna hantera MIME-typerna (Multipurpose Internet Mail Extensions) för de typer av filer du vill använda, till exempel filmer. Webbservern fastställer stöd för de aktuella MIME-typer som har registrerats för Internet. Web Publishing Engine ändrar inte en webbserverns MIME-stöd. Mer information finns i dokumentationen till webbservern.
- Alla QuickTime-filmer som lagras i containerfält lagras efter referens.

Containerfält med externt lagrade data

Om ett containerfält lagrar objekt externt – det vill säga om du har markerat **Lagra containerdata externt** i dialogrutan Fältalternativ – måste PHP-koden använda metoden `getContainerDataURL()` för att hämta en fullständig URL för containerfältobjektet.

Använd FileMaker Api för PHP för att definiera databasobjektet med rätt identifieringsuppgifter (kontonamn och lösenord), och använd sedan metoden `getContainerDataURL()`.

Exempel som visar bilder med taggen HTML img

```
$fm=new FileMaker($database, $hostspec, $user, $password);
$findCommand = $fm->newFindCommand($layout);
$findCommand->addFindCriterion('type', 'png');
$result = $findCommand->execute();
$records = $result->getRecords();
foreach ($records as $record) {
    echo $record->getField('container').'<br>';
    // För bilder använder du taggen HTML img
    echo '';
    break;
}
```

Exempel som visar inbäddade data med taggen HTML embed

```
$fm=new FileMaker($database, $hostspec, $user, $password);
$findCommand = $fm->newFindCommand($layout);
$findCommand->addFindCriterion('type', 'pdf');
$result = $findCommand->execute();
$records = $result->getRecords();
foreach ($records as $record) {
    echo $record->getField('container').'<br>';
    // För filmer och PDF-filer använder du taggen HTML embed
    //echo '<embed src="'. $fm->
        getContainerDataURL($record->getField('container')) .'">';
    break;
}
```

Överföra containerfältsdata till FileMaker Server

När du överför en databas med hjälp av FileMaker Pro, överförs externt lagrade containerfältsdata till FileMaker Server som en del av processen. Information om hur du överför databasfiler till FileMaker Server finns i direkthjälpen för FileMaker Pro.

Om du manuellt överför en databas som använder ett containerfält med externt lagrade objekt, måste du följa dessa steg när du ska publicera de externt lagrade containerobjekten med Web Publishing Engine.

Så här överför du en databas manuellt:

1. Placera databasfilen på rätt plats på servern. Placera de FileMaker Pro-databasfiler som ska öppnas av FileMaker Server – eller genvägar (Windows) eller alias (OS X) till filerna – i följande mappar:
 - Windows: [enhet]:\Program\FileMaker\FileMaker Server\Data\Databases\
där [enhet] är den primära enhet där programmet startas.
 - OS X: /Bibliotek/FileMaker Server/Data/Databases/Du kan också placera dina filer i någon annan databasmapp som du har angett.
2. I den mapp där du placerade databasen skapar du en mapp med namnet RC_Data_FMS, om den inte redan finns.
3. I mappen RC_Data_FMS skapar du sedan en ny mapp med samma namn som din databas. Om din databas till exempel heter Kunder skapar du en mapp som också heter Kunder. Placera de externt lagrade objekten i den nya mappen som du just skapade.

Obs! När databaserna administreras med FileMaker Server finns det inget sätt för flera databaser att dela en gemensam mapp bestående av containerobjekt. Containerobjekten för varje databas måste finnas i en mapp som identifieras av databasens namn.
4. För filer som ska delas från OS X ändrar du dem så att de tillhör gruppen **fmsadmin**. Mer information om hur du överför databaser manuellt finns i hjälpen för FileMakerPro Server.

Kommentar

- Web Publishing Engine stöder progressiv nedladdning av ljudfiler (.mp3), videofiler (.mov, .mp4, och .avi rekommenderas), och PDF filer för interaktiva containrar. En webbanvändare kan till exempel börja titta på en film även om inte hela filmfilen har laddats ner ännu. Om du vill möjliggöra progressiv nedladdning kan du behöva skapa filer genom att använda alternativ som stöder direktuppspelning eller som optimerar för visning på webben. Skapa till exempel PDF-filer genom att använda alternativet för webbvisningsoptimering.
- Om alternativet **Aktivera säkra anslutningar** i FileMaker Server inte är valt, kommer anslutningarna som FileMaker Server använder för att överföra data inte vara krypterade under överföringen.
 - FileMaker-klienter ser interaktiva containerdata med liten fördröjning.
 - FileMaker Server dekrypterar containerfältdata till en cachemapp på servern när en FileMaker Pro-, FileMaker Go- eller webbklient skickar en förfrågan om data. Dessa data kan vara dekrypterade i cachemappen på servern i två timmar innan FileMaker Server regelbundet tömmer cachemappen. Dessa data cachelagras inte lokalt på klienten.
- När alternativet **Aktivera säkra anslutningar** i FileMaker Server är valt, kommer FileMaker Server använda säkra anslutningar för att överföra data. FileMaker-klienter hämtar containerdata fullständigt innan användaren kan interagera med dem. Data är lika säkra som om lösningen var en lokal databas, eftersom det inte skapas några temporära cachefiler och data krypteras under överföring.

Databasservern måste stoppas och startas om när inställningen **Aktivera säkra anslutningar** ändras för att den nya inställningen ska gälla.

Hur webbanvändare kan visa containerfältobjekt

När du publicerar en databas med Web Publishing Engine gäller följande begränsningar för containerfältobjekt:

- Webbanvändare kan inte ändra eller lägga till innehåll i ett containerfält. Webbanvändare kan inte använda containerfält för att ladda upp objekt till databasen.
- När det gäller databaser som använder ett containerfält som har miniatyrbilder aktiverat, laddar Web Publishing Engine ned hela filen, inte bara en miniatyrbild.

FileMaker-manus och anpassad webbpublicering

Funktionen Hantera manus i FileMaker Pro kan automatisera uppgifter som utförs ofta eller kombinera flera uppgifter. Genom att använda FileMaker-manus tillsammans med anpassad webbpublicering kan webbanvändare utföra en serie uppgifter. Med FileMaker-manus kan man även utföra uppgifter som inte stöds på något annat sätt, till exempel att använda manussteget Ändra lösenord för att ge webbanvändarna möjlighet att ändra lösenord från en webbläsare.

FileMaker stöder över 65 manussteg för anpassad webbpublicering. Om du vill se vilka manussteg som inte stöds väljer du kryssrutan **Anpassad webbpublicering** i listan **Visa kompatibilitet** i fönstret Ändra manus i FileMaker Pro. De manussteg som är nedtonade stöds inte av Anpassad webbpublicering. Information om hur du skapar manus finns i direkthjälpen för FileMaker Pro.

Tips och råd om manus

Många manussteg fungerar på samma sätt på webben men det finns några som fungerar annorlunda. Se "Funktionssätt i egna webbpubliceringslösningar" på sidan 24. Testa alla manus som ska användas från en webbläsare innan du delar din databas. Logga in med olika användarkonton så att du är säker på att de fungerar för alla klienter.

Här följer några tips och råd:

- Använd konton och behörighet när du vill styra vilka manus en webbanvändare kan utföra. Kontrollera att manusen endast innehåller webbkompatibla manussteg och ge bara tillgång till manus som ska användas från en webbläsare.
- Var också uppmärksam på eventuella biverkningar av manus som utför kombinationer av steg som styrs via behörigheten. Om ett manus exempelvis innehåller steget Radera poster och en webbanvändare inte loggar in med ett konto som ger behörighet att radera poster, utförs inte manussteget Radera poster. Manuset kan dock fortsätta att utföras, vilket kan ge oväntade resultat.
- I fönstret Ändra manus väljer du **Kör manus med fullständig behörighet** om du vill låta manuset utföra uppgifter som du inte vill låta den enskilda användaren utföra. Du kan till exempel hindra användare från att radera poster genom att begränsa deras konton och behörighet, men ändå tillåta dem att utföra ett manus som raderar vissa typer av poster under vissa förutsättningar som anges i manuset.

- Om ett manus innehåller steg som inte stöds, exempelvis steg som inte är webbkompatibla, kan du använda manussteget **Tillåt användaren att avbryta** för att avgöra hur efterföljande steg ska hanteras.
 - Om manussteget **Tillåt användaren att avbryta** är aktiverat (på), stoppas manuset vid manussteg som inte stöds.
 - Om **Tillåt användaren att avbryta** är avstängt kommer manussteg som inte stöds att hoppas över och manuset att fortsätta.
 - Om det här manussteget inte ingår utförs manus som om funktionen vore aktiverad, så att manus stoppas vid manussteg som inte stöds.
- Vissa manus som fungerar med ett steg från en FileMaker Pro-klient kan kräva ett ytterligare steg Verkställ post/sökpost för att spara data på värddatorn. Eftersom webbanvändarna inte har någon direktanslutning till värddatorn får de inte veta när data ändras. Funktioner som till exempel villkorliga värdelistor är inte lika smidiga för webbanvändare eftersom data måste sparas på värddatorn innan effekterna märks i fältet med värdelistan.
- Manus som förändrar data bör innehålla steget Verkställ post/sökpost, eftersom vissa dataändringar inte syns i webbläsaren förrän data sparas eller "skickas" till servern. Detta gäller flera manussteg som Klipp ut, Kopiera, Klistra in, o.s.v. Många enkelstegsåtgärder bör konverteras till manus och innehålla steget Verkställ post/sökpost. När du skapar manus som ska utföras från en webbläsare ska du inkludera steget Verkställ post/sökpost sist i manuset, så att du är säker på att alla ändringar verkställs.
- Du kan skapa villkorsstyrda manus enligt typen av klient med funktionen Get(programversion). Om det returnerade värdet innehåller "Web Publishing Engine" vet du att den aktuella användaren använder din databas via anpassad webbpublicering. Mer information om funktioner finns i direkthjälpen för FileMaker Pro.
- När du har konverterat dina filer bör du öppna varje manus som webbanvändarna kan köra och markera **Webbpublicering** i listan Visa kompatibilitet i fönstret Ändra manus för att verifiera att manuset körs korrekt i Direkt webbpublicering.

Funktionssätt i egna webbpubliceringslösningar

Följande manussteg fungerar annorlunda på webben än i FileMaker Pro. Information om samtliga manussteg finns i direkthjälpen för FileMaker Pro.

Manussteg	Funktionssätt i egna webbpubliceringslösningar
Utför manus	Manus kan inte utföras i andra filer, såvida inte filerna ligger på FileMaker Server och anpassad webbpublicering är aktiverad i de andra filerna.
Avsluta programmet	Loggar ut webbanvändare, stänger alla fönster men avslutar inte webbläsaren.
Tillåt användaren att avbryta	Bestämmer hur manussteg som inte stöds ska hanteras. Välj På om manuset ska stoppas vid steg som inte stöds eller Av om sådana steg ska hoppas över. Närmare information finns i "Tips och råd om manus" på sidan 22. Obs! Webbanvändare kan inte avbryta manus i anpassad webbpublicering, men detta alternativ gör att manussteg som inte stöds förhindrar att manuset fortsätter.
Manuell felhantering	Detta är alltid aktiverat i anpassad webbpublicering. Webbanvändarna kan inte avbryta manus i anpassad webbpublicering.
Paus/fortsätt manus	Trots att dessa manussteg stöds i anpassad webbpublicering, bör du undvika att använda dem. När ett paussteg körs, görs en paus i manuset. Det är bara manus som innehåller ett fortsätt-manussteg som kan återuppta körningen. Om manuset förblir i pausläge tills sessionen avslutas, slutförs inte manuset.
Sortera poster	Du måste spara en sorteringsordning med manussteget Sortera poster om det ska utföras vid anpassad webbpublicering.
Öppna URL	Detta manussteg har ingen effekt i en anpassad webbpubliceringslösning.
Gå till fält	Du kan inte använda Gå till fält för att aktivera ett visst fält i webbläsaren, men du kan använda manussteget tillsammans med andra manussteg för att utföra uppgifter. Du kan till exempel gå till ett fält, kopiera innehållet, gå till ett annat fält och klistra in värdet. Se till att spara posten med manussteget Verkställ post om du vill se resultatet i webbläsaren.
Verkställ post/sökpost	Skickar posten till databasen.

Manustriggers i egna webbpubliceringslösningar

I FileMaker Pro kan både manus och användaråtgärder (som att användaren klickar i ett fält) aktivera manustrigger. I anpassad webbpublicering kan endast manus aktivera manustrigger. Om t.ex. en användare av anpassad webbpublicering klickar på ett fält som har en VidObjektStarta-manustrigger kommer utlösaren inte att aktiveras. Om ett manus gör att fokus flyttas till fältet så kommer VidObjektStarta-manustriggern att aktiveras. Mer information om dessa manustrigger finns i direkthjälpen för FileMaker Pro.

Obs! När du behöver ange att du vill att ett manus ska köras när en fil öppnas måste du använda manustriggern IFörstaFönsterÖppna. När du behöver ange att du vill att ett manus ska köras när en fil stängs måste du nu av samma anledning använda manustriggern ISistaFönsterStäng.

Kapitel 4

Översikt över anpassad webbpublicering med PHP

Med FileMaker Api för PHP kan du integrera data från FileMaker Pro-databaser med PHP-lösningar. I detta kapitel beskrivs hur PHP fungerar tillsammans med FileMaker Servers anpassade Web Publishing Engine. Mer detaljerad information om FileMaker Api för PHP finns i kapitel 5, "Använda FileMaker Api för PHP".

Hur Web Publishing Engine fungerar med PHP-lösningar

FileMaker Server består av tre delar: en webbserver, Web Publishing Engine och databasservern. (Dessa komponenter kan köras på en eller två datorer. Se *FileMaker Server startbok* för mer information.) FileMaker Server är värd för PHP-lösningen när du placerar PHP-filerna på webbservern där PHP-motorn är installerad.

- När en webbanvändare öppnar en PHP-lösning vidarebefordrar webbservern förfrågan till PHP-motorn, som behandlar PHP-koden.
- Om PHP-koden innehåller anrop till FileMaker Api för PHP tolkas de anropen och skickas som förfrågningar till Web Publishing Engine.
- Web Publishing Engine efterfrågar data från databaser som databasservern är värd för.
- Databasservern skickar efterfrågade data till Web Publishing Engine.
- Web Publishing Engine skickar data till PHP-motorn på webbservern som svar på API-anropet.
- PHP-lösningen behandlar informationen och visar den för webbanvändaren.

Allmänna steg för anpassad webbpublicering med PHP

Här följer en sammanfattning över de steg du måste utföra när du vill använda anpassad webbpublicering med PHP:

1. Kontrollera att **Aktivera PHP-publicering** är aktiverat i Admin Console. Se *FileMaker Server startbok*.
2. I Admin Console väljer du fönstret **Databaser** och ser till att varje FileMaker-databas som du vill publicera har den **fmphp**-utökade behörigheten aktiverad för anpassad webbpublicering med PHP.

Om det behövs använder du FileMaker Pro och aktiverar anpassad webbpublicering för en databas. Se kapitel 3, "Förbereda databaser för anpassad webbpublicering".

Obs! Se till att använda motsvarande FileMaker-databasbehörigheter när du utvecklar PHP-lösningar som ska användas av slutanvändaren. I annat fall kanske du har tillgång till layouter och funktioner i FileMaker-databasen som inte är tillgängliga för slutanvändaren, vilket ger inkonsekventa funktioner.

3. Skapa din PHP-lösning med PHP-redigeringsverktygen och integrera FileMaker API-funktioner i din PHP-kod för att få åtkomst till dina FileMaker-data. Se kapitel 5, "Använda FileMaker Api för PHP".

4. Kopiera eller flytta webbplatsens katalogstruktur och filer till följande mapp på webbservern.
 - För IIS (Windows) via HTTP eller HTTPS:
`[enhet] : \Program\FileMaker\FileMaker Server\HTTPServer\Config` där [enhet] är den enhet där FileMaker Server-distributionens komponent Web Publishing Engine ligger.
 - För Apache (OS X) via HTTP: `/Bibliotek/FileMaker Server/HTTPServer/htdocs`
 - För Apache (OS X) via HTTPS:
`/Bibliotek/FileMaker Server/HTTPServer/htdocs/httpsRoot`
5. Om ett containerfält i en databas lagrar en filreferens istället för en verklig fil måste det hänvisade containerobjektet lagras i FileMaker Pro-katalogen när posten skapas eller redigeras. Du måste kopiera eller flytta objektet till en katalog med samma relativa position i rotkatalogen på webbserverprogramvaran.
Se "Publicera innehållet i containerfält på webben" på sidan 18.
6. Se till att säkerhetsfunktionerna för platsen eller programmet fungerar.
7. Testa din webbplats genom att använda samma konton och behörigheter som har definierats för webbanvändare.
8. Informera om webbplatsen och gör den tillgänglig. Den URL som webbanvändarna anger har detta format:

```
http://<server>/<webbplatsens_sökväg>
```

- <server> är den maskin där FileMaker Server finns
- <webbplatsens_sökväg> är den relativa sökväg till startsidan på din webbplats, som avgörs av den katalogstruktur som du använde i steg 4 ovan.

Till exempel, om din webbserver är 192.168.123.101 och din webbplats startsida ligger på webbservern på `c:\inetpub\wwwroot\customers\index.php`, ska webbanvändaren ange denna URL:

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

Obs! PHP 5 använder Latin-1-kodning (ISO-8859-1). FileMaker Server returnerar Unicode-data (UTF-8). Använd FileMaker Server Admin Console för att ange standardteckenkodningen för din webbplats. För PHP-webbplatser kan du ange antingen UTF-8 eller ISO-8859-1, men vi rekommenderar UTF-8. Ange samma inställning för `charset`-attributet i `<HEAD>`-avsnittet i webbplatsens PHP-filer.

Mer information om hur du installerar och använder en PHP-lösning finns i kapitel 6, "Iscensätta, testa och övervaka en webbplats".

Kapitel 5

Använda FileMaker Api för PHP

I FileMaker Api för PHP ingår en PHP-klass – FileMaker class – som ger ett objektorienterat gränssnitt för FileMaker-databaser. FileMaker Api för PHP gör att man kan få åtkomst till både uppgifter och logik som har lagrats i FileMaker Pro-databaser och publicera dem på Internet, eller exportera dem till andra program.

Med FileMaker Api för PHP kan PHP-kod utföra samma sorts funktioner som du redan har tillgång till i FileMaker Pro-databaser:

- skapa, radera, redigera och duplicera poster
- utföra sökningar
- kontrollera fält och poster
- använda layouter
- köra FileMaker-manus
- visa portaler och relaterade poster
- använda värdelistor

I detta kapitel beskrivs hur man använder FileMaker class-objekt och metoder för att lägga till dessa vanliga funktioner i en PHP-lösning. Detta kapitel täcker inte hela FileMaker Api för PHP, men introducerar nyckelobjekt och metoder.

Mer information

Mer information om FileMaker Api för PHP finns på följande ställen.

Om du redan har en PHP-motor installerad och konfigurerad och du endast vill lägga till FileMaker API för PHP går du till "Installera FileMaker Api för PHP manuellt" på sidan 13.

Referens för FileMaker Api för PHP

Om du installerade FileMaker API för PHP hittar du referensinformation i webbplatskomponenten i din FileMaker Server-driftsättning.

- För IIS (Windows):
[enhet] : \Program\FileMaker\FileMaker Server\Documentation\PHP API
Documentation\index.html
där [enhet] är den enhet där webbserverkomponenten för din FileMaker Server-driftsättning finns.
- För Apache (OS X): /Library/FileMaker Server/Documentation/PHP API
Documentation/index.html

Introduktion till FileMaker Api för PHP

Om du installerade FileMaker API för PHP hittar du en introduktion i webbplatskomponenten i din FileMaker Server-driftsättning.

- För IIS (Windows): [enhet] : \Program\FileMaker\FileMaker Server\Examples\PHP\Tutorial där [enhet] är den enhet där webbserverkomponenten för din FileMaker Server-installation finns.
- För Apache (OS X): /Bibliotek/FileMaker Server/Examples/PHP/Tutorial

Om du vill ha dessa PHP-introduktionsfiler kopierar du dem till webbserverns rotkatalog.

Exempel till FileMaker Api för PHP

Om du installerade FileMaker API för PHP hittar du ytterligare exempel i webbplatskomponenten i din FileMaker Server-driftsättning.

- För IIS (Windows): [enhet] : \Program\FileMaker\FileMaker Server\Examples\PHP\API Examples där [enhet] är den enhet där webbserverkomponenten för din FileMaker Server-driftsättning finns.
- För Apache (OS X): /Library/FileMaker Server/Examples/PHP/API Examples

Om du vill ha dessa API-exempelfiler kopierar du dem till webbserverns rotkatalog.

Använda FileMaker class

Om du vill använda FileMaker class i din PHP-lösning lägger du till följande uttryck i din PHP-kod:

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

FileMaker class-objekt

FileMaker class definierar klassobjekt som du kan använda för att hämta uppgifter från FileMaker Pro-databaser.

Klassobjekt	Använda objektet för att
FileMaker-databas	Definiera databasegenskaperna Ansluta till en FileMaker Pro-databas Skaffa information om FileMaker Api för PHP
Kommando	Skapa kommandon som lägger till poster, tar bort poster, duplicerar poster, ändrar poster, utför sökningar och utför manus.
Layout	Arbeta med databaslayouter
Post	Arbeta med postdata
Fält	Arbeta med fältdata
Relaterad uppsättning	Arbeta med portalposter
Resultat	Bearbeta posterna från en sökning
Fel	Kontrollera om ett fel har inträffat Behandla fel

FileMaker kommandoobjekt

FileMaker class definierar ett baskommandoobjekt som du kan använda för att exemplifiera ett specifikt kommando och för att ange kommandots parametrar. För att utföra kommandot måste du använda metoden `execute()`.

FileMaker class definierar följande specifika kommandon:

- Kommandot Lägg till
- Kommandot Sammanfattad sökning
- Kommandot Radera
- Kommandot Duplicera
- Kommandot Redigera
- Kommandot Utför sökning, kommandot Visa alla, kommandot Sök efter alla
- Kommandot Sökning, som läggs till i kommandot Sammanfattad sökning
- Kommandot Utför manus

Dessa kommandon beskrivs mer i detalj i följande avsnitt:

- "Arbeta med poster" på sidan 30
- "Köra FileMaker-manus" på sidan 31
- "Utföra sökningar" på sidan 38

Ansluta till en FileMaker-databas

FileMaker class definierar ett databasobjekt som du exemplifierar för att ansluta till en server eller en databas. Definiera objektgenskaperna med klasskonstruktören, eller genom att använda metoden `setProperty()`.

Exempel: Ansluta till en server för att få en lista över databaser

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

Exempel: Ansluta till en specifik databas på en server

Egenskaperna för användarnamn och lösenord avgör behörighetsuppsättningen för denna anslutning.

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

Obs! Egenskapen `hostspec` har standardvärdet `http://localhost`. Om PHP-motorn körs på samma maskin som webbserverkomponenten i FileMaker Server-lösningen behöver man inte ange `hostspec`-egenskapen. Om PHP-motorn körs på en annan maskin använder man `hostspec` för att ange platsen för webbserverkomponenten i FileMaker Server-lösningen.

Arbeta med poster

FileMaker class definierar ett postobjekt som du kan exemplifiera för att arbeta med poster. Ett exempel på ett postobjekt representerar en post från en FileMaker Pro-databas. Använd ett postobjekt med kommandona Lägg till, Radera, Duplicera och Redigera för att ändra uppgifterna i posten. Sökkommandona – Utför sökning, Visa alla, Sök efter alla och Sammanfattad sökning – returnerar en samling postobjekt.

Skapa en post

Det finns två sätt att skapa en post:

- Använd metoden `createRecord()` och ange ett layoutnamn, du kan även ange olika fältvärden. Du kan även ställa in värden individuellt i det nya postobjektet.

Metoden `createRecord()` sparar inte den nya posten i databasen. Om du vill spara posten i databasen använder du metoden `commit()`.

Till exempel:

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

- Använda kommandot Lägg till. Använd metoden `newAddCommand()` för att skapa ett `FileMaker_Command_Add`-objekt, ange layoutnamnet och postdata. Om du vill spara posten i databasen använder du metoden `execute()`.

Till exempel:

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

Duplicera en post

Du kan duplicera en befintlig post med kommandot Duplicera. Använd metoden `newDuplicateCommand()` för att skapa ett `FileMaker_Command_Duplicate`-objekt, ange layoutnamn och post-ID för den post du vill duplicera. Duplicera sedan posten genom att använda metoden `execute()`.

Exempel

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

Redigera en post

Det finns två sätt att redigera en post:

- Använda kommandot Redigera. Använd metoden `newEditCommand()` för att skapa ett `FileMaker_Command_Edit`-objekt, ange layoutnamn, post-ID för den post du vill redigera och värden som du vill uppdatera. Redigera sedan posten genom att använda metoden `execute()`.

Till exempel:

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

- Använda ett postobjekt. Hämta en post från databasen, ändra fältvärdena och redigera sedan posten genom att använda metoden `commit()`.

Till exempel:

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

Radera en post

Det finns två sätt att radera en post:

- Hämta en post från databasen och använd sedan metoden `delete()`.

Till exempel:

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

- Du kan radera en befintlig post med kommandot Radera. Använd metoden `newDeleteCommand()` för att skapa ett `FileMaker_Command_Delete`-objekt, ange layoutnamn och post-ID för den post du vill radera. Radera sedan posten genom att använda metoden `execute()`.

Till exempel:

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

Köra FileMaker-manus

Ett FileMaker-manus är en namngiven uppsättning manussteg. FileMaker class definierar olika metoder som gör att du kan arbeta med FileMaker-manus som har definierats i en FileMaker Pro-databas. Mer information om webbkompatibla manussteg (de manussteg som kan utföras i en webblösning) finns i "FileMaker-manus och anpassad webbpublicering" på sidan 22.

Hämta en lista på tillgängliga manus

Använd metoden `listScripts()` för att hämta en lista över tillgängliga manus från den aktuella anslutna databasen. Metoden `listScripts()` returnerar manus som kan utföras av det användarnamn och lösenord som angavs när databasanslutningen definierades. (Se "Ansluta till en FileMaker-databas" på sidan 29.)

Exempel

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

Köra ett FileMaker-manus

Använd metoden `newPerformScriptCommand()` för att skapa ett `FileMaker_Command_PerformScript`-objekt, ange layout, manusnamn och alla manusparametrar. Utför sedan manuset genom att använda metoden `execute()`.

Exempel

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

Köra ett manus innan ett kommando utförs

Använd metoden `setPreCommandScript()` för att ange ett manus som körs innan ett kommando körs. Följande exempel använder ett sökkommando, men du kan använda metoden `setPreCommandScript()` med vilket kommando som helst.

Exempel

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

Köra ett manus innan en resultatuppsättning sorteras

Använd metoden `setPreSortScript()` för att ange ett manus som körs efter ett sökresultat har genererats, men innan resultatuppsättningen sorteras. Mer information finns i "Använda kommandot Sök" på sidan 39.

Exempel

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

Köra ett manus efter resultatuppsättningen har genererats

Använd metoden `setScript()` för att ange ett manussteg som körs efter ett sökresultat har genererats. Mer information finns i "Använda kommandot Sök" på sidan 39.

Exempel

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


Ordning för att utföra manus

Du kan använda metoderna `setPreCommandScript()`, `setPreSortScript()` och `setScript()` tillsammans med metoderna `setResultLayout()` och `addSortRule()` för ett enda kommando. FileMaker Server och Web Publishing Engine bearbetar dessa metoder i följande ordning:

1. Kör det manus som har angetts för metoden `setPreCommandScript()` om detta har angetts.
2. Behandlar själva kommandot, exempelvis kommandona Utför sökning eller Radera post.
3. Kör det manus som har angetts för metoden `setPreSortScript()` om detta har angetts.
4. Sorterar sökresultatet, om metoden `addSortRule()` har angetts.
5. Gör att metoden `setResultLayout()` växlar till en annan layout, om detta har angetts.
6. Kör det manus som har angetts för metoden `setScript()` om detta har angetts.
7. Returnerar det slutgiltiga sökresultatet.

Om något av ovanstående steg genererar en felkod, stoppas kommandot och alla eventuella kvarvarande steg utförs inte. Eventuella tidigare steg i förfrågan utförs fortfarande.

Anta att ett kommando tar bort den aktuella posten, sorterar posterna och sedan kör ett manus. Om metoden `addSortRule()` anger ett fält som inte finns, tar förfrågan bort den aktuella posten och genererat felkod 102 ("Fältet saknas"), men manuset utförs inte.

Arbeta med FileMaker-layouter

En layout är en samordning av fält, objekt, bilder och layoutdelar som visar på vilket sätt information organiseras och presenteras när användaren bearbetar, granskar eller skriver ut poster. FileMaker class definierar olika metoder som gör att du kan arbeta med de layouter som har definierats i en FileMaker Pro-databas. Du kan få information om layouter från flera av FileMaker class-objekten.

Med detta klassobjekt	Använd dessa metoder
Databas	<ul style="list-style-type: none"> ▪ <code>listLayouts()</code> hämtar en lista över tillgängliga layoutnamn. ▪ <code>getLayout()</code> hämtar ett layoutobjekt genom att ange ett layoutnamn.
Layout	<ul style="list-style-type: none"> ▪ <code>getName()</code> hämtar layoutnamnet för ett specifikt layoutobjekt. ▪ <code>listFields()</code> hämtar alla fältnamn som används i en layout. ▪ <code>getFields()</code> hämtar ett associativt urval av namn på alla fält som nycklar, och de associativa FileMaker_Field-objekten som matrisvärden. ▪ <code>listValueLists()</code> hämtar en lista över namn på värdelistor. ▪ <code>listRelatedSets()</code> hämtar en lista över namn på relaterade uppsättningar. ▪ <code>getDatabase()</code> returnerar namnet på databasen.
Post	<ul style="list-style-type: none"> ▪ <code>getLayout()</code> returnerar det layoutobjekt som är associerat med en specifik post.
Fält	<ul style="list-style-type: none"> ▪ <code>getLayout()</code> returnerar det layoutobjekt som innehåller ett specifikt fält.
Kommando	<ul style="list-style-type: none"> ▪ <code>setResultLayout()</code> returnerar kommandots resultat i en annan layout än den aktuella layouten.

Använda portaler

En portal är en tabell som visar rader med data från en eller flera relaterade poster. FileMaker class definierar en relaterad uppsättning objekt och flera metoder som gör att du kan arbeta med portaler som har definierats i en FileMaker Pro-databas.

Ett relaterat uppsättningsobjekt är en matris med postobjekt från den relaterade portalen – varje postobjekt representerar en rad med data i portalen.

Lista de portaler som har definierats på en specifik layout

För ett specifikt layoutobjekt använder du metoden `listRelatedSets()` för att hämta en lista över tabellnamn för alla portaler som har definierats i denna layout.

Exempel

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

Hämta portalnamn för ett specifikt resultatobjekt

För ett specifikt FileMaker_Result-objekt använder du metoden `getRelatedSets()` för att hämta namnen på alla portaler i denna post.

Exempel

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

Hämta information om portaler för en specifik layout

För ett specifikt layoutobjekt använder du metoden `getRelatedSets()` för att hämta FileMaker_RelatedSet-objekt som beskriver portalerna i layouten. Den returnerade matrisen är en associativ matris med tabellnamnen som matrisnycklar och de associerade FileMaker_RelatedSet-objekten som matrisvärden.

Exempel

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

Hämta information för en specifik portal

För ett specifikt layoutobjekt använder du metoden `getRelatedSet()` för att hämta det FileMaker_RelatedSet-objekt som beskriver den specifika portalen.

Exempel

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

Hämta tabellnamnet för en portal

För ett relaterat uppsättningsobjekt använder du metoden `getName()` för att få tabellnamnet för portalen.

Exempel

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

Hämta portalposterna för en specifik post

För ett specifikt postobjekt använder du metoden `getRelatedSet()` för att hämta relaterade poster för en specifik portal på den posten.

Exempel

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

Skapa en ny post i en portal

Använd metoden `newRelatedRecord()` för att skapa en ny post i den angivna relaterade uppsättningen och utför ändringen i databasen genom att använda metoden `commit()`.

Exempel

```
//skapa en ny portalrad i "kundportalen"
$new_row = $currentRecord->newRelatedRecord('customer');

//ställ in fältvärden på den nya portalraden
$new_row->setField('customer::name', $newName);
$new_row->setField('customer::company', $newCompany);

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

Ta bort en post från en portal

Använd metoden `delete()` för att ta bort en post i en portal.

Exempel

```
$relatedSet = $currentRecord->getRelatedSet('customers');
/* Kör igenom alla portalrader */
foreach ($relatedSet as $nextRow) {

    $nameField = $nextRow->getField('customer::name')
    if ($nameField == $badName ) {
        $result = $nextRow->delete();
    }
}
```

Använda värdelistor

En värdelista är en uppsättning fördefinierade alternativ. FileMaker class definierar olika metoder som gör att du kan arbeta med värdelistor som har definierats i en FileMaker Pro-databas.

Hämta namnen på alla värdelistor för en specifik layout

För en specifik layout använder du metoden `listValueLists()` för att hämta en matris som innehåller namnen på värdelistorna.

Exempel

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

Hämta en matris över alla värdelistor för en specifik layout

För ett specifikt layoutobjekt använder du metoden `getValueListsTwoFields()` för att hämta en matris som innehåller värden från alla värdelistor. Den returnerade matrisen är en associativ matris. Matrisnycklarna är värdelistnamnen och matrisvärdena är associerade matriser som listar visningsnamn och deras relaterade alternativ från varje värdelista.

Exempel

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

Obs! Metoden `getValueLists()` stöds fortfarande i FileMaker API för PHP, men den kommer att avvecklas. Använd i stället metoden `getValueListsTwoFields()`.

Hämta värdena för en namngiven värdelista

För ett specifikt layoutobjekt använder du metoden `getValueListTwoFields()` för att hämta en matris som innehåller alternativen för den namngivna värdelistan. Den returnerade matrisen är en associerad matris som innehåller visade värden från det andra fältet i värdelistan som nycklar och de associerade lagrade värdena från det första fältet som matrisvärden.

Beroende på vilka alternativ du har markerat i dialogrutan Ange fält i värdelista i FileMaker-databasen returnerar metoden `getValueListTwoFields()` endast värdet i det första fältet, endast värdet i det andra fältet eller värdena i de båda fälten från en värdelista som lagrade och visade värden.

- Om du inte har markerat **Visa även värden från det andra fältet** kommer metoden `getValueListTwoFields()` att returnera värdet från det första fältet i värdelistan som både lagrat värde och visat värde.
- Om du har valt både **Visa även värden från det andra fältet** och **Visa endast värden från det andra fältet** returnerar metoden `getValueListTwoFields()` värdet från det första fältet som lagrat värde och värdet från det andra fältet som visat värde.
- Om du har markerat **Visa även värden från det andra fältet** och inte har markerat **Visa värden endast från det andra fältet** returnerar metoden `getValueListTwoFields()` värdet från det första fältet som lagrat värde och värdena från både det första och det andra fältet som visat värde.

Använd upprepning med metoden `getValueListTwoFields()` när du vill hitta visat värde och lagrat värde.

Exempel

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

Kommentar

- Även om metoden `getValueLists()` fortfarande stöds i FileMaker API för PHP kommer den att avvecklas. Använd istället metoden `getValueListsTwoFields()`.
- När du använder metoden `getValueListTwoFields()` ska du se till att använda en `foreach`-loop för varje loop i den associerade matrisen. Använd inte en `for`-loop eftersom den kan returnera oväntade resultat.

Utföra sökningar

FileMaker class definierar fyra olika sökkommandoobjekt:

- Kommandot Visa alla. Se "Använda kommandot Visa alla" på sidan 39.
- Kommandot Sök efter alla. Se "Använda kommandot Sök efter alla" på sidan 39.
- Kommandot Utför sökning. Se "Använda kommandot Sök" på sidan 39.
- Kommandot Sammanfattad sökning. Se "Använda kommandot Sammanfattad sökning" på sidan 40.

FileMaker class definierar även flera metoder som kan användas för alla fyra typer av sökkommandon:

- Använd metoden `addSortRule()` för att lägga till en regel som definierar hur sökresultatet ska sorteras. Använd metoden `clearSortRules()` för att rensa alla sorteringsregler som har definierats.
- Använd `setLogicalOperator()` för att växla mellan logiska AND-sökningar och logiska OR-sökningar.
- Använd metoden `setRange()` för att endast efterfråga en del av sökresultatet. Använd metoden `getRange()` för att hämta den aktuella intervalldefinitionen.

Genom att använda metoden `setRange()` kan man förbättra lösningens prestanda genom att minska antalet poster som returneras av sökförfrågan. Om en sökförfrågan exempelvis returnerar 100 poster kan du dela upp sökresultatet i fem grupper om 20 träffar istället för att behandla alla 100 poster på en gång.

- Du kan utföra FileMaker-manus tillsammans med sökkommandon.
 - Om du vill köra ett manus innan du utför sökkommandot använder du metoden `setPreCommandScript()`.
 - Om du vill köra ett manus innan du sorterar sökresultatet använder du metoden `setPreSortScript()`.
 - Om du vill köra ett manus efter ett sökresultat har genererats, men innan sökresultatet sorteras använder du metoden `setScript()`.

Använda kommandot Visa alla

Använd kommandot Visa alla för att hämta alla poster från en specifik layout. Använd metoden `newFindAllCommand()` och ange en specifik layout för att skapa ett `FileMaker_Command_FindAll`-objekt. Utför sedan sökförfrågan genom att använda metoden `execute()`.

Exempel

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

Obs! När du använder kommandot Visa alla ska du undvika att låta datorns minne bli överbelastat av problem genom att ange maximalt antal returnerade poster per sida.

Använda kommandot Sök efter alla

Använd kommandot Sök efter alla för att slumpmässigt hämta en post från en specifik layout. Använd metoden `newFindAnyCommand()` och ange en specifik layout för att skapa ett `FileMaker_Command_FindAny`-objekt. Utför sedan sökförfrågan genom att använda metoden `execute()`.

Exempel

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

Använda kommandot Sök

Använd metoden `newFindCommand()` och ange en specifik layout för att skapa ett `FileMaker_Command_Find`-objekt. Utför sedan sökförfrågan genom att använda metoden `execute()`.

Använd metoden `addFindCriterion()` för att lägga till villkor i sökförfrågan. Använd metoden `clearFindCriteria()` för att rensa alla sökvillkor som har definierats.

Exempel - Söka efter en post med fältnamn

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

Exempel - Lägga till en sorteringsordning

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

Använda kommandot Sammanfattad sökning

Med kommandot Sammanfattad sökning kan du kombinera flera sökobjekt i ett enda kommando.

Så här skapar du kommandot Sammanfattad sökning:

- Skapa ett `FileMaker_Command_CompoundFind`-objekt genom att använda metoden `newCompoundFindCommand()`.
- Skapa ett eller flera `FileMaker_Command_FindRequest`-objekt genom att använda metoden `newFindRequest()`.
- Använd metoden `setOmit()` för att ange poster i sökresultatet för en specifik sökförfrågan som ska uteslutas från det slutgiltiga sökresultatet.
- Använd metoden `add()` för att lägga till sökobjekten i kommandoobjektet Sammanfattad sökning.
- Utför sedan den sammanfattade sökningen genom att använda metoden `execute()`.

Exempel - Kommandot Sammanfattad sökning

```
// Skapa kommandoobjektet Sammanfattad sökning
$compoundFind = $fm->newCompoundFindCommand('Form View');

// Skapa den första sökförfrågan
$findreq1 = $fm->newFindRequest('Form View');

// Skapa den andra sökförfrågan
$findreq2 = $fm->newFindRequest('Form View');

// Skapa den tredje sökförfrågan
$findreq3 = $fm->newFindRequest('Form View');

// Ange sökvillkor för den första sökförfrågan
$findreq1->addFindCriterion('Quantity in Stock', '<100');

// Ange sökvillkor för den andra sökförfrågan
$findreq2->addFindCriterion('Quantity in Stock', '0');
$findreq2->setOmit(true);

// Ange sökvillkor för den tredje sökförfrågan
$findreq3->addFindCriterion('Cover Photo Credit', 'The London Morning News');
$findreq3->setOmit(true);

// Ange sökförfrågningar till kommandot Sammanställd sökning
$compoundFind->add(1,$findreq1);
$compoundFind->add(2,$findreq2);
$compoundFind->add(3,$findreq3);

// Ange sorteringsordning
$compoundFind->addSortRule('Title', 1, FILEMAKER_SORT_DESCEND);

// Utför kommandot Sammanfattad sökning
$result = $compoundFind->execute();

// Hämta poster från sökresultatet
$records = $result->getRecords();

// Skriv ut antalet funna poster
echo 'Found '. count($records) . " results.<br><br>";
```

Behandla posterna i ett sökresultat

- Hämta en matris med alla poster i sökresultatet genom att använda metoden `getRecords()`. Varje del i matrisen är ett `FileMaker_Record`-objekt, eller ett exempel på det klassnamn som har angetts i Api för att exemplifiera poster. Matrisen kan vara tom om sökresultatet inte innehåller några poster.
- Hämta en lista över fältnamn för alla fält i sökresultatet genom att använda metoden `getFields()`. Metoden returnerar endast fältnamnen. Om du behöver ytterligare information om fälten använder du associerade layoutobjekt.
- Hämta antalet poster i hela sökresultatet genom att använda metoden `getFoundSetCount()`.
- Hämta antalet poster i det filtrerade sökresultatet genom att använda metoden `getFetchCount()`. Om inga parametrar har ställts in för sökkommandot är detta värde lika med resultatet med metoden `getFoundSetCount()`. Det är alltid lika med värdet för `count($response->getRecords())`.
- För en specifik post använder du metoden `getField()` för att returnera innehållet i ett fält som en sträng.
- För en specifik post använder du metoden `getFieldAsTimestamp()` för att returnera innehållet i ett fält som en Unix-tidsstämpel (PHP-intern representation av ett datum).
 - Om fältet är ett datumfält anges tidsstämpeln för fältdatumet vid midnatt.
 - Om fältet är ett tidsfält anges tidsstämpeln för den tiden den 1 januari 1970.
 - Om fältet är ett tidsstämpelfält kopplas FileMaker-tidsstämpelvärdet direkt till Unix-tidsstämpeln.
 - Om det angivna fältet inte är ett datum- eller tidsfält, eller om den genererade tidsstämpeln skulle ligga utanför intervallet returnerar metoden `getFieldAsTimestamp()` ett `FileMaker_Error`-objekt.
- För en specifik post använder du metoden `getContainerData()` för att returnera ett containerfältobjekt som binära data.

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

- För en specifik post använder du metoden `getContainerDataURL()` för att returnera en fullständig URL för containerfältobjektet.

```
// För bilder använder du taggen HTML img
echo '';
// För filmer och PDF-filer använder du taggen HTML embed
//echo '<embed src="'. $fm->
getContainerDataURL($record->getField('container')) ."'>';
```

Filtrera portalrader som returnerats av en sökförfrågan

I en lösning som har många relaterade poster kan sökning och sortering av portalposter vara tidskrävande. För att begränsa antalet poster som visas i ett sökresultat kan du använda metoden `setRelatedSetsFilters()` för sökresultat. Metoden `setRelatedSetsFilters()` använder två argument:

- en relaterad uppsättnings filtervärde: `layout` eller `inget`.
 - Om du anger värdet `inget` returnerar Web Publishing Engine alla rader i portalen, och portalposterna sorteras inte i förväg.
 - Om du anger värdet `layout` respekteras inställningarna som har angetts i dialogrutan FileMaker Pro Portalkonfiguration. Posterna sorteras baserat på den sortering som har angetts i dialogrutan Portalkonfiguration, med postuppsättningen filtrerad för att starta med den första angivna raden.
- högsta antal portalposter som returneras: ett heltalsvärde eller `alla`.
 - Detta värde används endast om inställningen Visa vertikal rullningslist är aktiverad i dialogrutan Portalkonfiguration. Om du anger ett heltal returneras det antalet rader efter den första raden. Om du anger `alla` returnerar Web Publishing Engine alla relaterade poster.
 - Om inställningen Visa vertikal rullningslist är inaktiverad kommer inställningen Antal rader i dialogrutan Portalkonfiguration att avgöra det maximala antalet relaterade poster som returneras.

Förkontrollera kommandon, poster och fält

Med FileMaker class kan du *förkontrollera* fältuppgifter i en PHP-lösning på webbservern innan du skickar uppgifterna till databasen.

När du avgör om du ska använda förkontroll eller inte, ska du tänka över antalet datavärden som webbanvändaren anger. Om användaren uppdaterar ett litet antal fält kan du förbättra prestandan genom att inte använda förkontroll. Om användaren däremot anger data i många fält kan förkontroll förhindra att användaren blir frustrerad över att en post avvisas av databasen på grund av kontrollfel.

Med FileMaker class förkontrollerar PHP-motorn följande fältvillkor:

- `ej tomt`
Giltiga data är en icke-tomt teckensträng. Dessa data måste innehålla minst ett tecken.
- `endast numeriskt`
Giltiga data innehåller endast numeriska tecken.
- `maximalt antal tecken`
Giltiga data innehåller högst det antal tecken som angetts.
- `fysiffrigt årtal`
Giltiga data är en teckensträng som representerar ett datum med ett fysiffrigt årtal i formatet `D/M/ÅÅÅÅ`, där `M` är ett tal mellan 1 och 12, `D` är ett tal mellan 1 och 31 och `ÅÅÅÅ` är ett fysiffrigt tal mellan 0001 och 4000. Till exempel är `30/1/3030` ett giltigt fysiffrigt årtalsvärde. Strängen `31/4/2013` är dock ett ogiltigt fysiffrigt årtalsvärde eftersom april inte har 31 dagar. Datumkontroll stöder snedstreck (`/`), omvänt snedstreck (`\`) och bindestreck (`-`) som avgränsare. Strängen får dock inte innehålla en blandning av avgränsarna. Till exempel är `30\1-2013` ogiltigt.

- tidpunkt

Giltiga data är en teckensträng som representera ett 12-timmarsvärde i ett av dessa format:

- H
- T:M
- T:M:S
- T:M:S FM/EM
- T:M FM/EM

där T är ett tal mellan 1 och 12 och M och S är tal mellan 1 och 60.

Förkontrollen i PHP-motorn stöder implicit kontroll av fältdata baserat på fälttypen.

- datum

Ett fält om har definierats som ett datumfält kontrolleras enligt reglerna i "fysisiffrigt årtal", förutom att årtalsvärdet kan innehålla 0-4 siffror (årtalsvärdet kan vara tomt). Till exempel är 1/30 ett giltigt datum även om det inte har något årtal angivet.

- tid

Ett fält som har definierats som ett tidsfält kontrolleras enligt reglerna för "tidpunkt", förutom att timkomponenten (T) kan vara ett tal mellan 1 och 24 för att stödja 24-timmarsvärden.

- tidsstämpel

Ett fält som har definierats som ett tidsstämpelfält kontrolleras enligt reglerna för "tid" för tidskomponenten och enligt reglerna för "datum" för datumkomponenten.

FileMaker class kan inte förkontrollera alla fältkontrollalternativ som är tillgängliga i FileMaker Pro. Följande kontrollalternativ kan inte förkontrolleras eftersom de är beroende av status för alla data i databasen vid det tillfälle då data verkställs:

- unikt värde
- befintligt värde
- i intervall
- medlem i värdelista
- kontrollera genom beräkning

Förkontrollera poster i ett kommando

För ett kommandoobjekt använder du metoden `validate()` för att kontrollera ett fält eller hela kommandot mot förkontrollreglerna som drivs igenom av PHP-motorn. Om du godkänns av argumentet valfritt fältnamn förkontrolleras endast det fältet.

Om förkontrollen godkänns returnerar metoden `validate()` TRUE. Om förkontrollen inte godkänns returnerar metoden `validate()` ett `FileMaker_Error_Validation`-objekt som innehåller information om vad som inte kunde kontrolleras.

Förkontrollera poster

För ett postobjekt använder du metoden `validate()` för att kontrollera ett fält eller alla fält i posten mot förkontrollreglerna som drivs igenom av PHP-motorn. Om du godkänns av argumentet valfritt fältnamn förkontrolleras endast det fältet.

Om förkontrollen godkänns returnerar metoden `validate()` `TRUE`. Om förkontrollen inte godkänns returnerar metoden `validate()` ett `FileMaker_Error_Validation`-objekt som innehåller information om vad som inte kunde kontrolleras.

Förkontrollera fält

För ett fältobjekt använder du metoden `validate()` för att avgöra om ett givet värde är giltigt för ett fält.

Om förkontrollen godkänns returnerar metoden `validate()` `TRUE`. Om förkontrollen inte godkänns returnerar metoden `validate()` ett `FileMaker_Error_Validation`-objekt som innehåller information om vad som inte kunde kontrolleras.

Behandla kontrollfel

När förkontrollen inte godkänns innehåller det `FileMaker_Error_Validation`-objekt som returneras en matris med tre element för varje kontrollfel:

1. Det fältobjekt som inte kunde förkontrolleras
2. Ett konstant kontrollvärde som anger vilken kontrollregel som inte godkändes:
 - 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. Det verkliga värdet som angetts för det fält som inte kunde förkontrolleras

Du kan även använda följande metoder med `FileMaker_Error_Validation`-objekt:

- Använd metoden `isValidationError()` för att testa om felet är ett kontrollfel.
- Använd metoden `numErrors()` för att visa det antal kontrollregler som inte kunde godkännas.

Exempel

```
//Skapa en Lägg till-förfrågan
$addrequest = $fm->newAddCommand('test', array('join' => 'added', 'maxchars' =>
'abcx', 'field' => 'something' , 'numericonly' => 'abc'));

//Kontrollera alla fält
$result = $addrequest->validate();

//Om metoden validate() returnerade några fel skriver du ut namnet på fältet,
felnumret och det värde som inte kunde godkännas.
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";
    }
}
```

Utenhet

```
Kontrollen misslyckades:
Fältnamn: numericonly
Felkod: 2
Värde: abc
Fältnamn: maxchars
Felkod: 3
Värde: abcx
```

Hantera fel

FileMaker class definierar FileMaker_Error-objekt för att hjälpa dig att hantera fel som uppstår i en PHP-lösning.

Ett fel kan uppstå när ett kommando körs. Om ett fel uppstår returnerar kommandot ett FileMaker_Error-objekt. Det är bra att kontrollera det fel som returnerar när ett kommando körs.

Använd följande metoder för att lära dig mer om felet som anges i FileMaker_Error-objektet.

- Testa om en variabel är ett FileMaker_Error-objekt genom att använda metoden `isError()`.
- Hämta antalet fel som uppstått genom att använda metoden `numErrors()`.
- Hämta en matris med matriser som beskriver de fel som uppstått genom att använda metoden `getErrors()`.
- Visa ett felmeddelande genom att använda metoden `getMessage()`.

Exempel

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

Mer information om de felkoder som returneras med FileMaker_Error-objektet finns i bilaga A, "Felkoder för anpassad webbpublicering med PHP".

Kapitel 6

Iscensätta, testa och övervaka en webbplats

I detta kapitel finns instruktioner för att iscensätta och testa en webbplats med anpassad webbpublicering innan den körs i en produktionsmiljö. Instruktioner finns även för att använda loggfiler för att övervaka sidan vid testning och efter den har körts igång.

Iscensätta en anpassad webbpubliceringsplats

Innan du kan testa din webbplats ordentligt ska du kopiera och flytta de filer som behövs till korrekt plats på servern eller serverna där du ska iscensätta webbplatsen.

Så här iscensätter du och förbereder du webbplatsen för testning:

1. Genomför alla steg i kapitel 3, "Förbereda databaser för anpassad webbpublicering".
2. Kontrollera att Anpassad webbpublicering med PHP har aktiverats och konfigurerats på rätt sätt i FileMaker Server Admin Console.

Obs! Anvisningar finns i hjälpen för FileMaker Server.

3. Kontrollera att webbservern och Web Publishing Engine är i gång.
4. Kopiera eller flytta dina webbplatsfiler till webbserverkomponenten i din FileMaker-driftsättning.

Kopiera eller flytta dina webbplatsfiler till följande katalog på webbserverdatorn:

- IIS (Windows) via HTTP eller HTTPS:
[enhet]:\Program\FileMaker\FileMaker Server\HTTPServer\Conf där [enhet] är den enhet där FileMaker Server-distributionens komponent Web Publishing Engine ligger.
- Apache (OS X) via HTTP: /Bibliotek/FileMaker Server/HTTPServer/htdocs
- Apache (OS X) via HTTPS:
/Bibliotek/FileMaker Server/HTTPServer/htdocs/httpsRoot

5. Om du inte redan har gjort det ska du kopiera eller flytta alla hänvisade containerobjekt till rätt katalog på webbserverdatorn.
 - Om containerfälten innehåller de faktiska filerna i FileMaker-databasen behöver du inte flytta innehållet i containerfältet, under förutsättning att FileMaker Server fungerar som värd för databasfilen och att filen är tillgänglig på FileMaker Server.
 - Om ett containerfält i en databas lagrar en filreferens istället för en verklig fil måste det containerobjektet som referensen leder till lagras i FileMaker Pro Web-katalogen när posten skapas eller redigeras. Om du vill iscensätta din webbplats måste du kopiera eller flytta de hänvisade containrarna till en mapp med samma relativa position i rotkatalogen på webbserverprogramvaran.
 - När du använder FileMaker Pro för att överföra en databas med containerfält som har lagrat objekt externt, överförs externt lagrade containerfältdata till FileMaker Server som en del av processen. Information om hur du överför databasfiler till FileMaker Server finns i direkthjälpen för FileMaker Pro.
 - När du manuellt överför en databas som använder ett containerfält med externt lagrade objekt, måste du kopiera eller flytta de externa objekten till en undermapp till mappen RC_Data_FMS, enligt beskrivningen i "Containerfält med externt lagrade data" på sidan 19. Se "Publicera innehållet i containerfält på webben" på sidan 18.
6. Börja testet av webbplatsen.

Testa en anpassad webbpubliceringsplats

Innan du meddelar användarna att din anpassade webbpubliceringsplats är tillgänglig ska du kontrollera att den visas och fungerar som du har tänkt dig.

- Testa funktioner som sökning och att lägga till, radera och sortera poster med olika konton och behörighet.
- Kontrollera att behörighetsinställningarna fungerar som väntat genom att logga in med olika konton. Kontrollera att inga obehöriga användare kan komma åt och ändra dina data.
- Kontrollera alla manus så att du vet att resultatet blir det väntade. Information om hur du skapar manus som fungerar väl på webben finns i "FileMaker-manus och anpassad webbpublicering" på sidan 22.
- Testa webbplatsen med olika operativsystem och webbläsare.
- När du skapar lösningar som använder sig av FileMaker API för PHP rekommenderar vi att du bygger lösningar som har stöd för cookies aktiverat. FileMaker-API:t för PHP har bättre svarstider när cookies är aktiverade. Cookies är inte ett krav om du ska använda funktionerna i Anpassad webbpublicering, men om du använder cookies kan Web Publishing Engine cachelagra information om sessionen.

Obs! Om du har installerat webbservern, Web Publishing Engine och databasservern i en endatorslösning kan du visa och testa din webbplats utan att använda en nätverksanslutning. Flytta dina webbplatsfiler till rätt katalog på den datorn, och ange följande URL i din webbläsare:

`http://127.0.0.1/<sökväg>`

där <sökväg> är den relativa sökvägen till startsidan på din webbplats.

Övervaka webbplatsen

Du kan använda följande typer av loggfiler när du vill övervaka den egna webbpubliceringsplatsen och hämta information om de webbanvändare som besöker webbplatsen:

- Webbserveråtkomst och felloggar
- Web Publishing Engine-loggen
- Fellogg för webbservermodulen
- Tomcat-loggar

Använda webbserveråtkomst och felloggar

- IIS (Windows): Webbservern Microsoft IIS genererar en åtkomstloggfil och visar felen i Loggboken (Windows) i stället för att skriva dem till en loggfil. Åtkomstloggfilen, som är i formatet W3C Extended Log File Format som standard, är en lista med alla inkommande HTTP-förfrågningar till webbservern. Du kan även använda formatet W3C Common Logfile Format för åtkomstloggen. Mer information finns i dokumentationen till Microsoft IIS-webbservern.
- Apache (endast OS X): Webbservern Apache genererar en åtkomstloggfil och en felloggfil. Åtkomstloggfilen på Apache, som är i formatet W3C Common Logfile Format som standard, är en post över alla inkommande HTTP-förfrågningar till webbservern. Felloggen på Apache är en lista med problem som rör bearbetningen av HTTP-förfrågningar. Mer information om dessa loggfiler finns i dokumentationen för Apache-webbservern.

Obs! Mer information om formaten W3C Common Logfile Format och W3C Extended Log File Format finns på webbplatsen <http://www.w3.org>.

Använda Web Publishing Engine-loggen

Som standard genererar Web Publishing Engine en loggfil med namnet wpe.log som innehåller en förteckning över alla Web Publishing Engine-fel som har inträffat, inklusive applikationsfel, användningsfel och systemfel. Du kan även låta Web Publishing Engine ta med information som hör till Anpassad webbpublicering, såsom XML-förfrågningar från slutanvändare om att generera webbpubliceringsmaterial eller ändringar av inställningarna för Anpassad webbpublicering.

Eftersom FileMaker API för PHP använder HTTP POST för att få åtkomst till Web Publishing Engine registrerar loggfilen wpe.log inte uppgifter om PHP-förfrågningarna. Du kan använda wpe.log för att se när användarna har skickat PHP-förfrågningarna genom att titta på de XML-förfrågningar som har loggats.

Filen wpe.log finns i följande katalog på Web Publishing Engine-komponenten i FileMakerServer-installationen:

- IIS (Windows): [enhet] : \Program\FileMaker\FileMaker Server\HTTPServer\Logs\wpe.log
där [enhet] är den primära enhet där programmet startas.
- Apache (OS X): /Bibliotek/FileMaker Server/HTTPServer/Logs/wpe.log

Inställningar för Web Publishing Engine-loggen

Filen wpe.log genereras om alternativet **Aktivera loggning för webbpublicering** har aktiverats i Admin Console.

Loggningsalternativ aktiverat	Information som registreras i wpe.log
Felnivåmeddelanden	Alla fel i Web Publishing Engine som har inträffat, inklusive applikationsfel, användningsfel och systemfel.
Informations- och felnivåmeddelanden	Alla fel som beskrivs ovan samt information om åtkomst till Web Publishing Engine. Den innehåller en förteckning över alla XML-förfrågningar från slutanvändare om att generera utdata från Anpassad webbpublicering.

Inställningen **Felnivåmeddelanden** är aktiverad som standard. Mer information om hur du ställer in dessa alternativ med Admin Console finns i hjälpen till FileMaker Server.

Viktigt! Med tiden kan filen wpe.log bli väldigt stor. I Admin Console kan du ange en maximal storlek för filen wpe.log. När filen wpe.log uppnår sin maxstorlek skapar Web Publishing Engine en säkerhetskopia av den med namnet wpe.log.1, samtidigt som den skapar en ny version av filen wpe.log. Om du vill behålla fler säkerhetskopior kan du kopiera filen wpe.log.1 med jämna mellanrum till ett eget arkiv.

Formatet i Web Publishing Engine-loggen

Filen wpe.log använder följande format i varje post:

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

där:

- [TIMESTAMP_GMT] är datum och tidpunkt för posten enligt Greenwich Mean Time (GMT).
- [WPC_HOSTNAME] är namnet på den dator där Web Publishing Engine är installerat.
- [CLIENT_IP:PORT] är IP-adressen och porten på den klient som XML-förfrågan kom ifrån.
- [ACCOUNT_NAME] är kontonamnet som används för att logga in på den värdbaseade FileMaker-databasen.
- [MODULE_TYPE] är antingen: XML, för förfrågningar med Anpassad webbpublicering med XML, eller PHP, för förfrågningar med Anpassad webbpublicering med PHP.
- [SEVERITY] är antingen INFO, som visar att det är ett informerande meddelande, eller ERROR, som visar att det är ett felmeddelande.
- [FM_ERROR_CODE] är felnumret som returneras för ett felmeddelande. Felnumret kan vara en felkod för FileMaker-databaser (se "Felkodsnummer för FileMaker-databaser" på sidan 54). Dessutom kan felnumret vara ett HTTP-felnummer, som föregås av "HTTP:".
- [RETURN_BYTES] är det antal byte som returneras av förfrågan.
- [MESSAGE] ger ytterligare information om loggposten.

Exempel på meddelanden i Web Publishing Engine-loggen

Följande exempel visar vilka typer av meddelanden som kan ingå i filen wpe.log:

- När Web Publishing Engine startas och stoppas

```
2013-06-02 15:15:31 -0700 - - - - INFO - - FileMaker Server
Web Publishing Engine started.
2013-06-02 15:46:52 -0700 - - - - INFO - - FileMaker Server
Web Publishing Engine stopped.
```

- Lyckade eller misslyckade XML-förfrågningar

```
2013-06-02 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"
2013-06-02 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"
```

- Manusfel

```
2013-06-02 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"
```

- Ändringar i inställningarna till Anpassad webbpublicering

```
2013-06-09 10:59:49 -0700 WPC_SERVER 192.168.100.101:0 jdoe - INFO
- - XML Web Publishing Engine is enabled.
```

- Systemfel

```
2013-06-02 15:30:42 -0700 WPC_SERVER 192.168.100.101:0 jdoe XML
ERROR - - Communication failed
```

Använda felloggen för webbservern

Om webbservern inte kan ansluta till Web Publishing Engine skapar webbservermodulen en loggfil över eventuella fel som uppstår vid bearbetningen. Loggfilen har namnet web_server_module_log.txt och finns på webbpubliceringskomponenten i FileMaker Server-installationen:

- IIS (Windows): [enhet] : \Program\FileMaker\FileMaker Server\Logs\web_server_module_log.txt
där [enhet] är den primära enhet där programmet startas.
- Apache (OS X): /Library/FileMaker Server/Logs/web_server_module_log.txt

Använda Tomcat-loggar

När det har uppstått ett problem i FileMaker Server som orsakas av ett internt webbserverfel kan det hjälpa att ta en titt i Tomcat-loggarna. Tomcat-loggarna finns på webbserverkomponenten i FileMakerServer-installationen:

- IIS (Windows): [enhet] : \Program\FileMaker\FileMaker Server\Admin\admin-master-tomcat\logs/
där [enhet] är den primära enhet där programmet startas.
- Apache (OS X): /Bibliotek/FileMaker Server/Admin/admin-master-tomcat/logs/

Felsöka webbplatsen

Om du har problem med att visa eller använda webbplatsen ska du kontrollera följande:

- Den utökade behörighetsuppsättningen i databasen är inställd för anpassad webbpublicering med PHP och kopplad till ett användarkonto. Se ”Aktivera anpassad webbpublicering med PHP för databaser” på sidan 15.
- Databasen finns på FileMaker Server och är öppen. Se direkthjälpen för FileMaker Server.
- Kontonamnet och lösenordet till databasen är korrekta.
- Webbservern och Web Publishing Engine måste vara igång.
- PHP-publicering är aktiverad i Web Publishing Engine.
 - Öppna sidan FileMaker Server tekniktest i en webbläsare:
`http://<server>:16000/test`
där <server> är den dator där FileMaker Server finns.
 - Klicka på länken **Testa PHP anpassad webbpublicering** för att öppna en PHP-sida som har åtkomst till FMServer_Sample testdatabasen.

Mer information finns i *FileMaker Server startbok* och FileMaker Server Hjälp.

Bilaga A

Felkoder för anpassad webbpublicering med PHP

Web Publishing Engine stöder två typer av felkoder för anpassad webbpublicering:

- Databas- och dataförfrågningsfel. Web Publishing Engine genererar en felkod när data efterfrågas från en publicerad databas. FileMaker Api för PHP returnerar denna felkod som ett FileMaker_Error-objekt. Se nästa avsnitt, "Felkodsnummer för FileMaker-databaser".
- PHP-fel. Dessa fel genereras och returneras av PHP-komponenter, inklusive cURL-modulen. Se "Felkodsnummer för PHP-komponenter" på sidan 61.

Felkodsnummer för FileMaker-databaser

Som utvecklare av en anpassad webbpubliceringslösning måste du kontrollera värdet för den returnerade felkoden och hantera det på rätt sätt. Web Publishing Engine hanterar inte databasfel.

Felnummer	Beskrivning
-1	Okänt fel
0	Inget fel
1	Användaren har avbrutit åtgärden
2	Minnesfel
3	Kommandot är inte tillgängligt (exempelvis fel operativsystem, fel läge osv)
4	Okänt kommando
5	Ogiltigt kommando (ett Tilldela fält-steg har exempelvis ingen beräkning angiven)
6	Filen är skrivskyddad
7	Minnet räcker inte
8	Tomt resultat
9	Otillräcklig behörighet
10	Begärda data saknas
11	Namnet är ogiltigt
12	Namnet finns redan
13	Filen eller objektet används
14	Utanför det tillåtna intervallet
15	Går inte att dela med noll
16	Åtgärden misslyckades, begär nytt försök (t.ex. en användarfråga)
17	Försök att konvertera främmande teckenuppsättning till UTF-16 misslyckades
18	Klienten måste ange kontoinformation för att fortsätta
19	Strängen innehåller andra tecken än A-Z, a-z, 0-9 (Ascii)
20	Kommando eller åtgärd avbröts av utlöst manus

Felnummer	Beskrivning
21	Begäran stöds inte (t.ex. när man skapa en hård länk på ett filsystem som inte stöder hårda länkar)
100	Fil saknas
101	Post saknas
102	Fält saknas
103	Relation saknas
104	Manus saknas
105	Layout saknas
106	Tabell saknas
107	Index saknas
108	Värdelista saknas
109	Behörighetsuppsättning saknas
110	Relaterade tabeller saknas
111	Fältrepetitionen är ogiltig
112	Fönstret saknas
113	Funktionen saknas
114	Filreferensen saknas
115	Menyuppsättningen saknas
116	Layoutobjektet saknas
117	Datakällan saknas
118	Tema saknas
130	Filerna är skadade eller saknas och måste installeras om
131	Språkpaketfiler saknas (t.ex. Startlösningar)
200	Tillgång till posten nekad
201	Det går inte att ändra fältet
202	Tillgång till fältet nekad
203	Det finns inga poster i filen att skriva ut, eller så saknar lösenordet behörighet för utskrift
204	Ingen tillgång till fält i sorteringsordning
205	Användaren har inte behörighet att skapa nya poster; importen kommer att skriva över befintliga data
206	Användaren har inte behörighet att ändra lösenord eller så kan filen inte ändras
207	Användaren har inte tillräcklig behörighet för att ändra databasschemat eller så kan filen inte ändras
208	Lösenordet innehåller inte tillräckligt många tecken
209	Det nya lösenordet måste skilja sig från det befintliga
210	Användarkontot är inte aktivt
211	Lösenordet har upphört att gälla
212	Ogiltigt användarkonto och/eller lösenord. Försök igen
213	Användarkontot och/eller lösenordet finns inte

Felnummer	Beskrivning
214	För många inloggningsförsök
215	Administratörsbehörighet kan inte dupliceras
216	Gästkontot kan inte dupliceras
217	Användaren har inte tillräcklig behörighet för att ändra administratörskontot
218	Lösenordet och bekräftelsen av lösenordet stämmer inte överens
300	Filen är låst eller används
301	Posten används av en annan användare
302	Tabellen används av en annan användare
303	Databasschemat används av en annan användare
304	Layouten används av en annan användare
306	Postens ändrings-ID matchar inte
307	Transaktionen gick inte att låsa på grund av ett kommunikationsfel med värden
308	Temat är låst och används av en annan användare
400	Sökvillkoren är tomma
401	Inga poster matchar sökposten
402	Det valda fältet är inte ett matchande fält för en länk
403	Maxpostgränsen för testversionen av FileMakerPro har överskridits
404	Sorteringsordningen är ogiltig
405	Det angivna antalet poster överskrider antalet poster som kan utelämnas
406	Ogiltiga villkor för ersättning/ändring av löpnummer
407	Ett av eller båda de matchande fälten saknas (ogiltig relation)
408	Det angivna fältet har fel datatyp för den här åtgärden
409	Importordningen är ogiltig
410	Exportordningen är ogiltig
412	Fel version av FileMakerPro används för att reparera filen
413	Det angivna fältet är av fel typ
414	Resultatet kan inte visas i layouten
415	En eller flera nödvändiga relaterade poster är inte tillgängliga
416	En primär nyckel krävs från datakällstabellen
417	Databasen är inte en datakälla som stöds
418	Internt fel vid INSERT-operation i ett fält
500	Datumvärdet stämmer inte med kontrolltillvalen
501	Tidsvärdet stämmer inte med kontrolltillvalen
502	Det numeriska värdet stämmer inte med kontrolltillvalen
503	Värdet i fältet stämmer inte med intervallet som angetts i kontrolltillvalen
504	Värdet i fältet är inte unikt, vilket angetts som ett krav i kontrolltillvalen
505	Värdet i fältet är inte ett befintligt värde i databasfilen, vilket angetts som ett krav i kontrolltillvalen
506	Värdet i fältet finns inte med i värdelistan som angetts i kontrolltillvalet

Felnummer	Beskrivning
507	Värdet i fältet klarade inte beräkningstestet för kontrolltillvalet
508	Ett ogiltigt värde har angetts i sökläget
509	Fältet kräver ett giltigt värde
510	Det relaterade värdet är tomt eller inte tillgängligt
511	Värdet i fältet överskrider maximal fältstorlek
512	Posten har redan ändrats av en annan användare
513	Ingen verifiering angavs, men data får inte plats i fältet
600	Utskriftsfel
601	Sidhuvudet och sidfoten ryms inte tillsammans på en sida
602	Huvuddelen ryms inte på en sida med den aktuella kolumninställningen
603	Ingen skrivaranslutning
700	Fel filformat för import
706	EPSF-filen saknar granskningsbild
707	Det går inte att hitta grafikkonverteraren
708	Det går inte att importera filen eller, så krävs stöd för färgskärm för att importera filen
709	Import av QuickTime-film misslyckades
710	Det går inte att uppdatera QuickTime-filmpreferensen eftersom databasfilen är skrivskyddad
711	Det går inte att hitta importkonverteraren
714	Lösenordet saknar behörighet för den här åtgärden
715	Det angivna Excel-kalkylbladet eller intervallet saknas
716	En SQL-fråga med DELETE, INSERT eller UPDATE är inte tillåten vid ODBC-import
717	Det finns inte tillräcklig XML/XSL-information för att importen eller exporten ska kunna fortsätta
718	Ett fel inträffade vid tolkningen av XML-filen (från Xerces)
719	Ett fel inträffade vid konverteringen av XML med hjälp av XSL (från Xalan)
720	Ett fel inträffade vid exporten, det angivna formatet har inte stöd för repeterade fält
721	Ett okänt fel inträffade vid tolkningen eller konverteringen
722	Det går inte att importera data till en fil utan fält
723	Du har inte behörighet att lägga till eller ändra poster i måltabellen
724	Du har inte behörighet att lägga till poster i måltabellen
725	Du har inte behörighet att ändra poster i måltabellen
726	Importfilen innehåller fler poster än måltabellen. Alla poster importerades inte
727	Måltabellen innehåller fler poster än importfilen. Alla poster uppdaterades inte
729	Fel inträffade vid importen. Posterna kunde inte importeras
730	Excel-versionen stöds inte (konvertera filen till Excel 2007/2008 eller senare version som stöds och försök igen)
731	Filen som du importerar från innehåller inga data
732	Den här filen kan inte sättas in eftersom den innehåller andra filer
733	En tabell kan inte importeras till sig själv

Felnummer	Beskrivning
734	Den här filtypen kan inte visas som en bild
735	Den här filtypen kan inte visas som en bild. Den kommer att sättas in och visas som en fil.
736	Det finns för många data för att exportera till detta format. Data kommer att förkortas.
737	Den Bento-tabell du försöker importera finns inte
738	Temat du importerar finns redan
800	Det går inte att skapa filen på disken
801	Det går inte att skapa någon tillfällig fil på systemdisken
802	Det går inte att öppna filen. Detta fel kan bero på en eller flera av följande orsaker: <ul style="list-style-type: none">▪ Ogiltigt databasnamn▪ Filen är stängd i FileMakerServer▪ Ogiltig behörighet
803	Filen är en enanvändarfil eller så går det inte att hitta värden
804	Filen går inte att öppna som skrivskyddad i aktuellt tillstånd
805	Filen är skadad; använd kommandot Reparera
806	Det går inte att öppna filen i den här versionen av FileMaker Pro
807	Filen är inte en FileMakerPro-fil eller så är den allvarligt skadad
808	Det går inte att öppna filen eftersom behörigheten är skadad
809	Disken/volymen är full
810	Disken/volymen är låst
811	Det går inte att öppna den tillfälliga filen som en FileMaker Pro-fil
812	Värdens kapacitet överskreds
813	Postsynkroniseringsfel i nätverket
814	Det går inte att öppna filen eftersom maximala antalet tillåtna filer redan är öppna
815	Det gick inte att öppna den länkade filen
816	Det går inte att konvertera filen
817	Det går inte att öppna filen eftersom den inte hör till denna lösning
819	Det går inte att spara en lokal kopia av en fjärrfil
820	Filen håller på att stängas
821	Frånkoppling framtvingad av värden
822	FMI-filer hittades inte; installera om saknade filer
823	Det går inte att göra filen till en enanvändarfil eftersom gäster är anslutna
824	Filen är skadad eller så är den inte en FileMaker-fil
825	Filen har inte behörighet att referera till den skyddade filen
826	Den angivna sökvägen är ingen giltig sökväg
850	Sökvägen är inte giltig i detta operativsystem
851	Det gick inte att radera en extern fil från disken
852	Det gick inte att spara filen på den externa lagringsresursen
853	En eller flera containrar överfördes inte

Felnummer	Beskrivning
900	Allmänt fel i stavningsmotorn
901	Standardordlistan är inte installerad
902	Det gick inte att starta hjälpsystemet
903	Detta kommando kan inte användas i en delad fil
905	Inget aktivt fält har valts och kommandot kan bara användas om det finns ett aktivt fält
906	Den aktuella filen är inte delad. Kommandot kan endast användas för delade filer
920	Det går inte att initiera stavningsmotorn
921	Det går inte att öppna den egna ordlistan för redigering
922	Det går inte att hitta den egna ordlistan
923	Den egna ordlistan är skrivskyddad
951	Ett oväntat fel har inträffat
954	XML-syntax stöds inte
955	Inget databasnamn
956	Det maximala antalet databassessioner har överskridits
957	Motstridiga kommandon
958	Parameter saknas i fråga
959	Tekniken för anpassad webbpublicering är inaktiverad
960	Parametern är ogiltig
1200	Allmänt beräkningsfel
1201	För få parametrar i funktionen
1202	För många parametrar i funktionen
1203	Oväntat slut på beräkningen
1204	Ett numeriskt värde, en textkonstant, ett fältnamn eller "(" krävs
1205	Kommentaren avslutas inte med "*/"
1206	Textkonstanten måste avslutas med ett citationstecken
1207	Udda parentes
1208	Operator saknas, funktionen hittades inte eller "(" förväntades inte
1209	Ett namn (t.ex. ett fältnamn eller ett layoutnamn) saknas
1210	Plugin-funktionen har redan registrerats
1211	Det är inte tillåtet att använda listor i den här funktionen
1212	En operator (t.ex. +, -, *) behövs här
1213	Den här variabeln har redan definierats i Let-funktionen
1214	AVERAGE, COUNT, EXTEND, GETREPETITION, MAX, MIN, NPV, STDEV, SUM och GETSUMMARY: uttryck hittades där det behövs ett ensamt fält
1215	Den här parametern är en ogiltig Get-funktionsparameter
1216	Första argumentet i en GETSUMMARY måste vara ett statistikfält
1217	Brytfältet är ogiltigt
1218	Det går inte att utvärdera talet
1219	Ett fält kan inte användas i sin egen formel

Felnummer	Beskrivning
1220	Fälttypen måste vara Normal eller Beräkning
1221	Datatypen måste vara numerisk, datum, tid eller tidsstämpel
1222	Beräkningen får inte vara lagrad
1223	Funktionen som det hänvisas till har inte implementerats än
1224	Funktionen du refererar till finns inte
1225	Funktionen som det hänvisas till stöds inte i denna kontext
1300	Det angivna namnet kan inte användas
1301	En av parametrarna i funktionen som importeras eller klistras in har samma namn som en funktion som redan finns i filen
1400	Det gick inte att initiera ODBC-klientdrivrutinen. Kontrollera att ODBC-klientdrivrutinerna är korrekt installerade
1401	Det gick inte att ställa in miljön (ODBC)
1402	Det gick inte att frigöra miljön (ODBC)
1403	Det gick inte att koppla från (ODBC)
1404	Det gick inte att tilldela en anslutning (ODBC)
1405	Det gick inte att frigöra en anslutning (ODBC)
1406	Kontroll av SQL-API misslyckades (ODBC)
1407	Det gick inte att tilldela en instruktion (ODBC)
1408	Utökat fel (ODBC)
1409	Utökat fel (ODBC)
1410	Utökat fel (ODBC)
1411	Utökat fel (ODBC)
1412	Utökat fel (ODBC)
1413	Utökat fel (ODBC)
1414	SQL-uttrycket är för långt
1450	Åtgärden kräver PHP-behörighetstillägg
1451	Åtgärden kräver att aktuell fil är en fjärrfil
1501	SMTP-verifieringen misslyckades
1502	Anslutningen avvisades av SMTP-servern
1503	Fel med SSL
1504	SMTP-servern kräver att anslutningen är krypterad
1505	Angiven verifiering stöds inte av SMTP-servern
1506	E-post kunde inte skickas
1507	Det går inte att logga in på SMTP-servern
1550	Det gick inte att läsa in plugin-programmet, eller också är det inte ett giltigt plugin-program
1551	Det gick inte att installera plugin-programmet. Det gick inte att radera ett befintligt plugin-program, eller också gick det inte att spara i mappen eller på disken
1626	Protokollet stöds inte
1627	Autentiseringen misslyckades

Felnummer	Beskrivning
1628	Ett fel uppstod med SSL
1629	Ingen anslutning (timeout). Tidsgränsen är 60 sekunder
1630	Felaktigt URL-format
1631	Anslutningen misslyckades

Felkodsnummer för PHP-komponenter

FileMaker Api för PHP använder flera PHP-komponenter. Dessa PHP-komponenter kan returnera flera felkoder som inte har angetts i ovanstående lista.

Till exempel, om webbpubliceringskärnan eller FileMaker Server inte körs, kan du få ett cURL-modulfelet `CURLE_GOT_NOTHING` (52).

Mer information om PHP-relaterade felkoder finns på PHP-webbplatsen på <http://php.net>.

Index

A

- Admin Console 14, 15
- aktivera anpassad webbpublicering i databas 15
- anpassad webbpublicering
 - aktivera i databas 15
- Anpassad webbpublicering
 - aktivera i Web Publishing Engine 16
 - använda manus 22
 - begränsa åtkomst av IP-adresser på webbserver 16
 - definition 7
 - manus 24
 - med PHP 9
 - med XML 9
 - utökad behörighet för 15
- ansluta
 - till en FileMaker Server 29
 - till en FileMaker-databas 29
- använda
 - layouter 33
 - manus 31
 - portaler 34
 - poster 30
 - värdelistor 36
- användarnamn
 - ange för anpassad webbpublicering 15

B

- behandla en webbpubliceringsförfrågan 8
- behandla ett sökresultat 42
- behörighet 17
- behörighetsuppsättning, tilldelat för anpassad webbpublicering 15
- beständiga databassessioner 15, 17

C

- client URL library 12
- containerfält
 - hur webbanvändare använder data 22
 - med externa filer 19
 - med externt lagrade data 19
 - publicera innehåll i 18
- cURL 12
- cURL-modulfel 61

D

- databas, felkodsnummer 54
- databas, skydda publicerad 16
- databasobjekt 29
- databassessioner, beständiga 15, 17
- datumfält 44
- datumrepresentation 42
- Dialogrutan Portalkonfiguration 43

- direktuppspelning. Se progressiv nedladdning
- dokumentation 6
- dokumentation online 6
- duplicera en post 30
- dynamisk IP-adress 12

E

- ej tomt fält 43
- elektronisk dokumentation 6
- endast numeriskt fält 43
- exempel till FileMaker Api för PHP 28
- extern SQL-datakälla 15

F

- fel
 - databas, felkodsnummer 54
 - hantera 47
 - loggfiler för webbserver 50
- felsöka webbplatser
 - Anpassad webbpublicering, webbplatser 49
 - kontrollera konfiguration 53
- FileMaker API för PHP
 - självstudier 28
 - definition 9
 - manuell installation 13
- FileMaker class 28
- FileMaker command-objekt
 - Duplicera 30
 - Lägg till 30
 - Radera 31
 - Redigera 30
- FileMaker Server
 - dokumentation 6
 - installera 6
- FileMaker Server Admin
 - se Admin Console
- FileMaker WebDirect
 - definition 7
 - dokumentation 6
- FileMaker-klassobjekt
 - databas 29
 - definition 29
 - post 30
 - relaterad uppsättning 34
- FileMaker-kommandoobjekt
 - Kommandot Sammanfattad sökning 40
 - Kommandot Sök 38, 39
 - Kommandot Sök efter alla 39
 - Kommandot Visa alla 39

fält

- datum 44
- ej tomt 43
- endast numeriskt 43
- fysiffrigt årtal 43
- maximalt antal tecken 43
- tid 44
- tidpunkt 44
- tidsstämpel 44
- fält med fysiffrigt årtal 43
- fält med maximalt antal tecken 43
- fält med tidpunkt 44
- fördelar med XML 9
- förkontrollera
 - datum 44
 - ej tomt 43
 - endast numeriskt 43
 - fysiffrigt årtal 43
 - fält 45
 - kommandon 43
 - maximalt antal tecken 43
 - poster 45
 - tid 44
 - tidpunkt 44
 - tidsstämpel 44

G

- Gäst, konto
 - aktivera 17
 - inaktivera 17

H

- hantera fel 47

I

- installation av FileMaker Api för PHP 13
- installationsdokumentation 6
- introduktion till FileMaker Api för PHP 28
- iscensätta webbplatser 48

J

- JDBC-dokumentation 6

K

- Kommandot
 - Sammanfattad sökning 40
- Kommandot Duplicera 30
- Kommandot Lägg till 30
- Kommandot Radera 31
- Kommandot Redigera 30
- Kommandot Sök 39
- Kommandot Sök efter alla 39
- Kommandot Visa alla 39

konton och behörighet

- aktivera för anpassad webbpublicering 15
- Gäst, konto 17
- manus 22
- kontroll av befintligt värde 44
- kontroll av medlem i värdelista 44
- kontroll av unikt värde 44
- kontroll av värden i intervall 44
- kontrollera
 - datum 44
 - ej tomt 43
 - endast numeriskt 43
 - fysiffrigt årtal 43
 - fält 45
 - kommandon 43
 - maximalt antal tecken 43
 - poster 45
 - tid 44
 - tidpunkt 44
 - tidsstämpel 44
- kontrollera genom beräkning 44

L

- Latin-1-kodning 26
- layouter 33
- loggfiler
 - beskrivning 50
 - Tomcat 52
 - web_server_module_log.txt 52
 - webbserveråtkomst 50
 - wpe.log 51

lösenord

- ange för anpassad webbpublicering 15
- inget lösenord för inloggning 17
- Ändra lösenord, manussteg 17

M

- manuell installation av FileMaker Api för PHP 13
- manus
 - i anpassad webbpublicering 22
 - konton och behörighet 22
 - Logga in igen 17
 - tips och råd 22
 - utlösare 24
 - Ändra lösenord 17
- Manuset Logga in igen 17
- metoden add() 40
- metoden addSortRule() 38
- metoden clearSortRules() 38
- metoden commit() 30
- metoden createRecord() 30
- metoden delete() 31, 36
- metoden getContainerData() 18, 42
- metoden getContainerDataURL() 19, 42
- metoden getDatabase() 33
- metoden getErrors() 47
- metoden getFetchCount() 42

metoden getField() 42
 metoden getFieldAsTimestamp() 42
 metoden getFields() 33, 42
 metoden getFoundSetCount() 42
 metoden getLayout() 33
 metoden getMessage() 47
 metoden getName() 33, 35
 metoden getRange() 38
 metoden getRecords() 42
 metoden getRelatedSet() 35
 metoden getRelatedSets() 35
 metoden getValueListsTwoFields() 37
 metoden getValueListTwoFields() 37
 metoden isError() 47
 metoden isValidationError() 45
 metoden listFields() 33
 metoden listLayouts() 33
 metoden listRelatedSets() 33, 34
 metoden listScripts() 31
 metoden listValueLists() 33, 36
 metoden newAddCommand() 30
 metoden newCompoundFindCommand() 40
 metoden newDeleteCommand() 31
 metoden newDuplicateCommand() 30
 metoden newEditCommand() 30
 metoden newFindAllCommand() 39
 metoden newFindAnyCommand() 39
 metoden newFindCommand() 39
 metoden newFindRequest() 40
 metoden newPerformScriptCommand() 32
 metoden newRelatedRecord() 36
 metoden numErrors() 45, 47
 metoden setLogicalOperator() 38
 metoden setOmit() 40
 metoden setPreCommandScript() 32, 38
 metoden setPreSortScript() 32, 38
 metoden setProperty() 29
 metoden setRange() 38
 metoden setRelatedSetsFilters() 43
 metoden setResultsLayout() 33
 metoden setScript() 32, 38
 metoden validate() 44
 metoder
 add() 40
 addSortRule() 38
 clearSortRules() 38
 commit() 30
 createRecord() 30
 delete() 31, 36
 getContainerData() 18, 42
 getContainerDataURL() 19, 42
 getDatabase() 33
 getErrors() 47
 getFetchCount() 42
 getField() 42
 getFieldAsTimestamp() 42
 getFields() 33, 42

getFoundSetCount() 42
 getLayout() 33
 getMessage() 47
 getName() 33, 35
 getRange() 38
 getRecords() 42
 getRelatedSet() 35
 getRelatedSets() 35
 getValueListsTwoFields() 37
 getValueListTwoFields() 37
 isError() 47
 isValidationError() 45
 listFields() 33
 listLayouts() 33
 listRelatedSets() 33, 34
 listScripts() 31
 listValueLists() 33, 36
 newAddCommand() 30
 newCompoundFindCommand() 40
 newDeleteCommand() 31
 newDuplicateCommand() 30
 newEditCommand() 30
 newFindAllCommand() 39
 newFindAnyCommand() 39
 newFindCommand() 39
 newFindRequest() 40
 newPerformScriptCommand() 32
 newRelatedRecord() 36
 numErrors() 45, 47
 setLogicalOperator() 38
 setOmit() 40
 setPreCommandScript() 32, 38
 setPreSortScript() 32, 38
 setProperty() 29
 setRange() 38
 setRelatedSetsFilters() 43
 setResultsLayout() 33
 setScript() 32, 38
 validate() 44

N

nummer för felkoder i databaser 54

O

ODBC-begränsningar 15
 ODBC-dokumentation 6
 OS X Server Admin 12

P

PHP

aktivera i databas 15
 fel 61
 felsökning 53
 fördelar 9
 sammanfattning av steg för publicering 25
 version som stöds 13
 webbplatstestning 49

PHP 5 12

PHP Api for anpassad webbpublicering 9
 portaler 34
 poster 30
 postobjekt 30
 programlogg 50
 progressiv nedladdning 18, 21
 publicera på webben
 använda PHP 25
 containerfältobjekt 18
 databaser, felkoder 54
 QuickTime-filmer 19
 skydda databaser 16

Q

QuickTime-filmer, publicera på webben 19

R

radera en post 31
 redigera en post 30
 Referens för FileMaker Api för PHP 27
 referensinformation 27
 relaterat uppsättningsobjekt 34

S

Sammanfattad sökning
 exempel 41
 SAT
 se Admin Console
 Server Admin-verktyg
 Se OS X Server Admin
 serverkrav 11
 skapa en post 30
 skydda publicerade databaser 16
 SSL-kryptering 16
 statisk IP-adress 12
 statisk publicering, definition 7
 säkerhet
 begränsa åtkomst från IP-adresser 16
 dokumentation 8
 konton och lösenord 16
 riktlinjer för att skydda publicerade databaser 16
 Sökkommandoobjekt 38
 sökresultat 42

T

tekniktest 53
 testa PHP-publicering 53
 testa webbplatser 49
 tidsfält 44
 tidsstämpelfält 42, 44
 Tomcat-loggar 52

U

Unicode 26
 Unix-tidsstämpel 42
 UTF-8-kodning 26
 utföra sökningar 38
 utlösare 24
 utökad behörighet för anpassad webbpublicering 15

V

värdelistor 36

W

Web Publishing Engine
 behandling av förfrågan 8
 beskrivning 8
 genererade felkoder 54
 programlogg 50
 web_server_module_log.txt, loggfil 52
 webbansvändare
 använda containerfältdata 22
 webbmap, kopiera containerfältobjekt 19
 webbplatser
 felsökning 53
 FileMakers supportsidor 6
 iscensätta 48
 testa 49
 övervaka 50
 webbserver
 loggfiler 50
 wpe.log, loggfil 51

X

XML anpassad webbpublicering 9

Å

åtkomstloggfiler för webbserver, beskrivning 50

Ä

Ändra lösenord, manussteg 17

Ö

översikt över PHP-publicering 25
 övervaka webbplatser 50