Explore cURL for FileMaker

Make sure you have the latest version of the
demo file on your laptop - download from
http://bit.ly/devcon-2017-cURL

INN004

Steve Winter
Matatiro Solutions
@steveWinterNZ

© 2017 FileMaker, Inc.

FILEMAKER DEVCON 2017
JULY 24 - 27, 2017 | PHOENIX, AZ

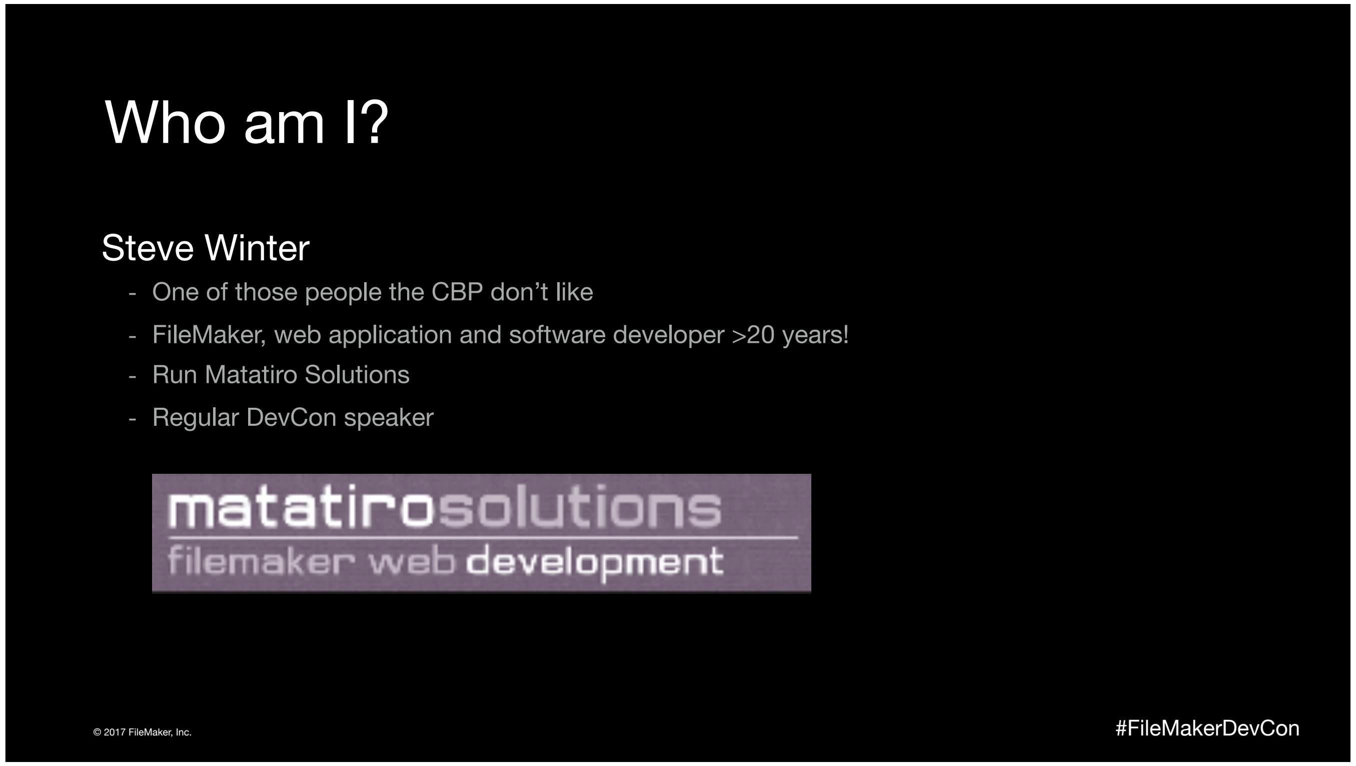
Explore cURL for FileMaker

INN004

Steve Winter
Matatiro Solutions
@steveWinterNZ

© 2017 FileMaker, Inc.

FILEMAKER DEVCON 2017
JULY 24 - 27, 2017 | PHOENIX, AZ



Who am I?

Steve Winter

- One of those people the CBP don’t like
- FileMaker, web application and software developer >20 years!
- Run Matatiro Solutions
- Regular DevCon speaker

matatirosolutions
filemaker web development

© 2017 FileMaker, Inc.

#FileMakerDevCon

Originally from New Zealand, I now have dual citizenship with the UK but live in Latvia, and have travelled rather a lot, including to countries like Libya and Iran - all of which makes the current US administration somewhat twitchy!

Company website: https://msdev.co.uk/

Who am I?

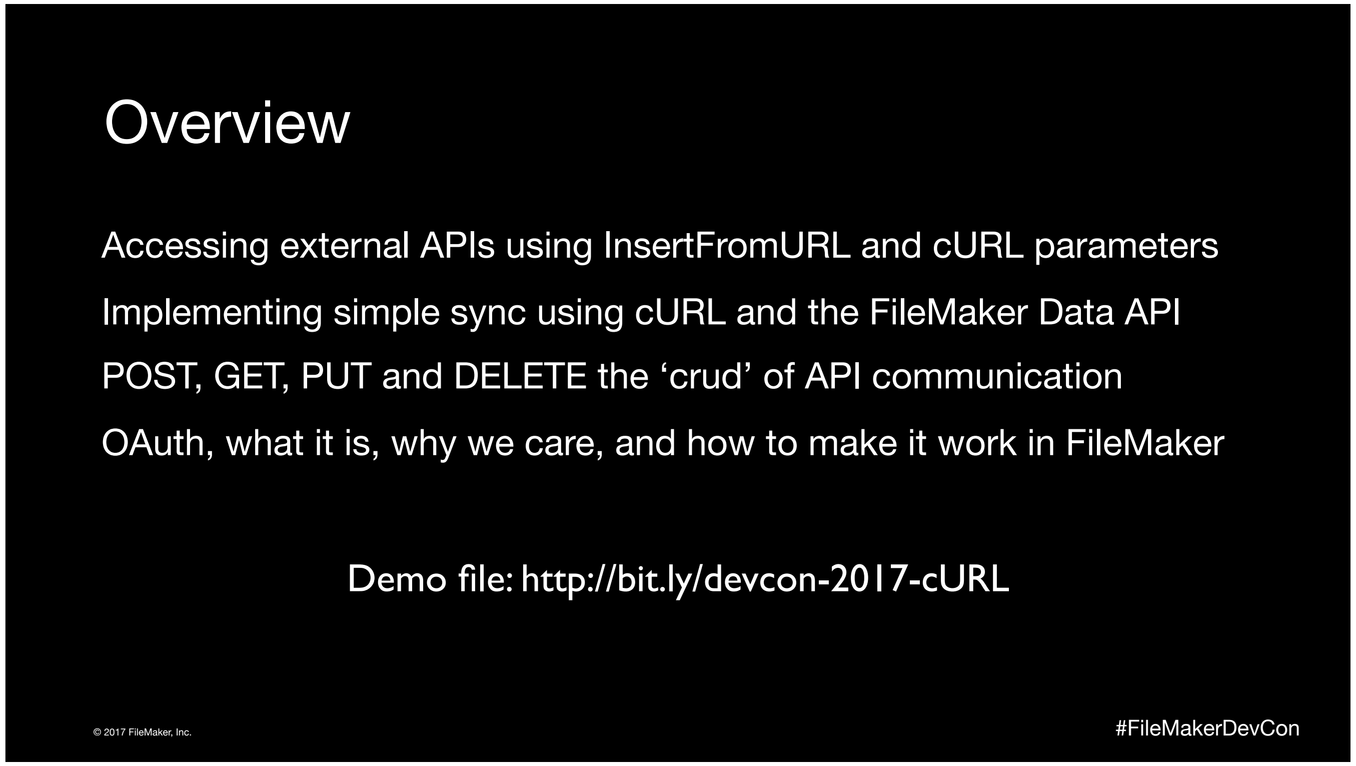
Steve Winter

© 2017 FileMaker, Inc.

#FileMakerDevCon

I didn’t attend DevCon last year as I was off on an adventure.

Trip blog: https://ride.bluecrocodile.co.nz



Overview

Accessing external APIs using InsertFromURL and cURL parameters

Implementing simple sync using cURL and the FileMaker Data API

POST, GET, PUT and DELETE the ‘crud’ of API communication

OAuth, what it is, why we care, and how to make it work in FileMaker

Demo file: http://bit.ly/devcon-2017-cURL

© 2017 FileMaker, Inc.

Wow… Did I really promise I was going to cover all that, and questions, in an hour?

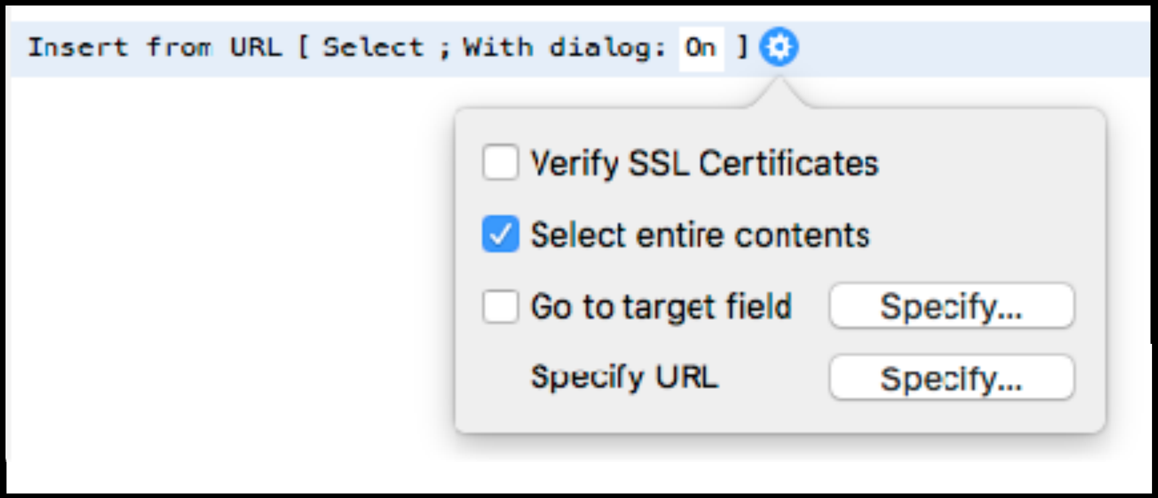
Best get started – oh, and on the topic of questions, if you could save them to the end – when I promise there will be time to ask them – then it really helps with the session recording!

In this session I’m going to do lots of demos, so please bear with me as we switch back and forth between Keynote and FM. I’ll try and show script steps and the data viewer as large as possible, but the easiest way to follow along is to have the demo file open on your laptop.

Everything that I demo today is (now) FileMaker native – no plugins, and I’ve chosen not to use any custom functions, though there are places where I would tend to abstract some repeated script steps or calcs into custom functions, I wanted this session to be only scripting. There’s one place where I’ve had to cheat a tiny bit which I’ll explain when we get there.

Right – on with the show!

Insert from URL



```
Insert from URL [ Select ; With dialog: On ]
```

☐ Verify SSL Certificates

☑ Select entire contents

☐ Go to target field

Specify URL

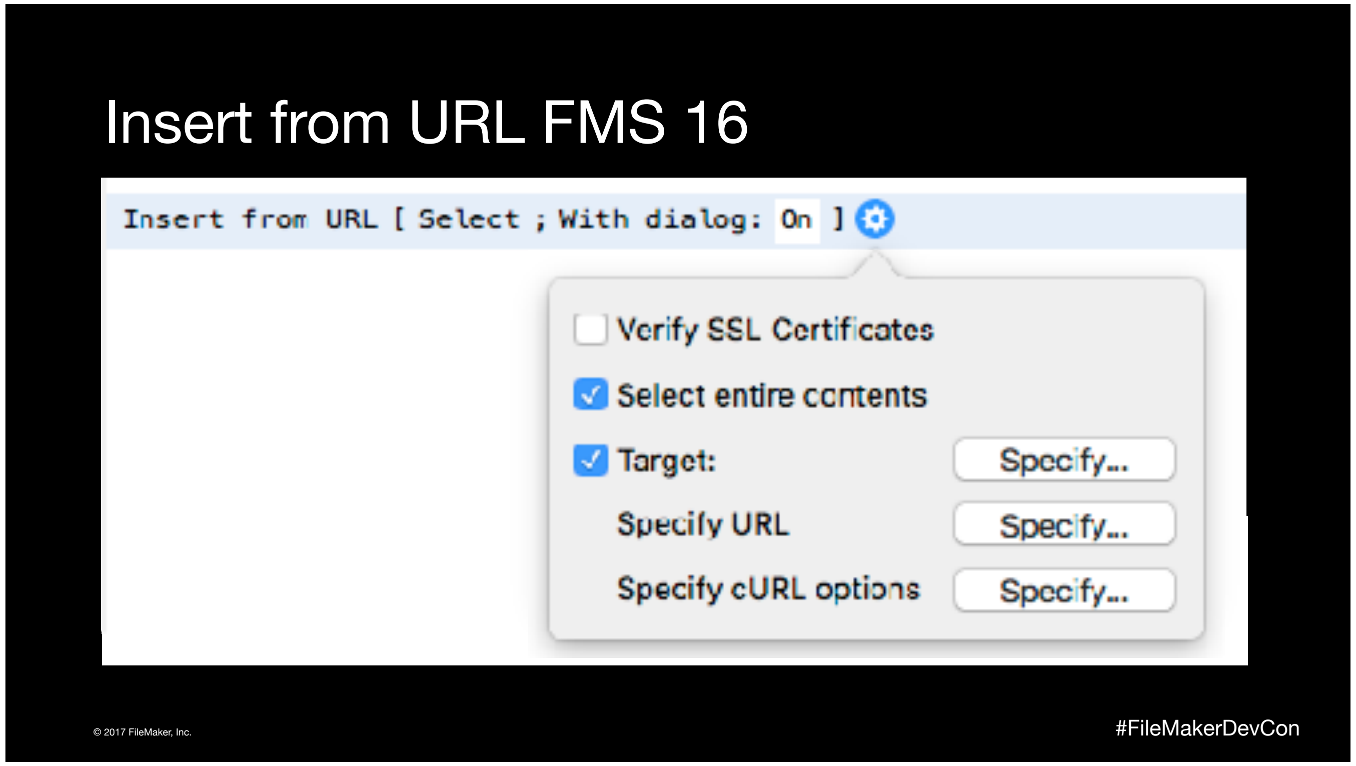
© 2017 FileMaker, Inc.

#FileMakerDevCon

I’m sure you’re all familiar with the Insert From URL config in FM < 16 where you could specify the field to save data to and URL – the URL was a calc field, so allowed for the passing of query parameters

```
https(post)://server.com/path/to/resource?param=value&param2=other+value
```

But that was it!



Insert from URL FMS 16

```
Insert from URL [ Select ; With dialog: On ]
```

□ Verify SSL Certificates

☑ Select entire contents

☑ Target: Specify…

Specify URL Specify…

Specify cURL options Specify…

© 2017 FileMaker, Inc.

#FileMakerDevCon

Target – can now Insert From URL into a variable as well as a field, great for remote API calls

Specify URL – remains the same, except that http(s)post is deprecated (though can still be used)

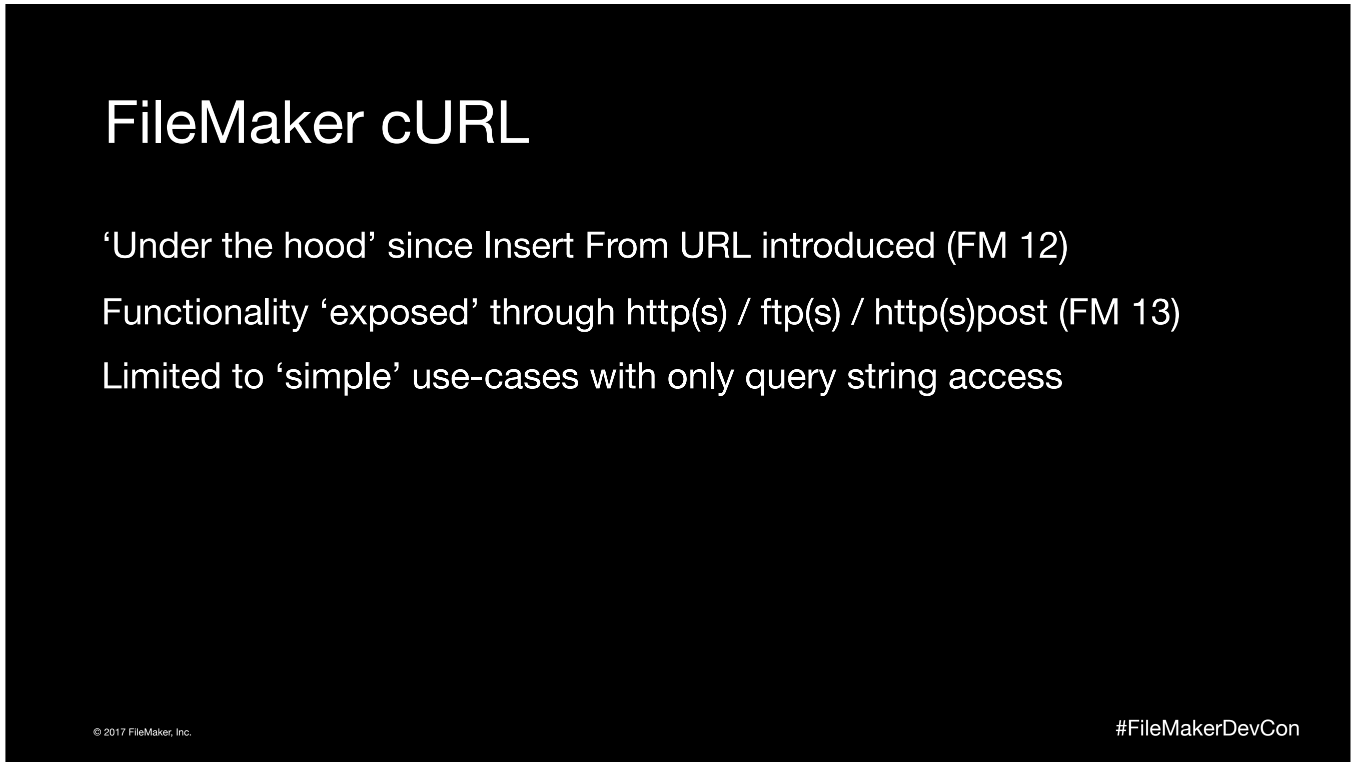
Specify cURL options – new, and allows us to pass additional parameters to the cURL request which is (already) being performed ‘behind the scenes’.

cURL

> “**cURL** (/kɝl/ or /kəːl/) is a computer software project providing a library and command-line tool for transferring data using various protocols. The cURL project produces two products, **libcurl** and **cURL**. It was first released in 1997.
>
> The name originally stood for "see URL”.”

Wikipedia

© 2017 FileMaker, Inc.



FileMaker cURL

‘Under the hood’ since Insert From URL introduced (FM 12)

Functionality ‘exposed’ through http(s) / ftp(s) / http(s)post (FM 13)

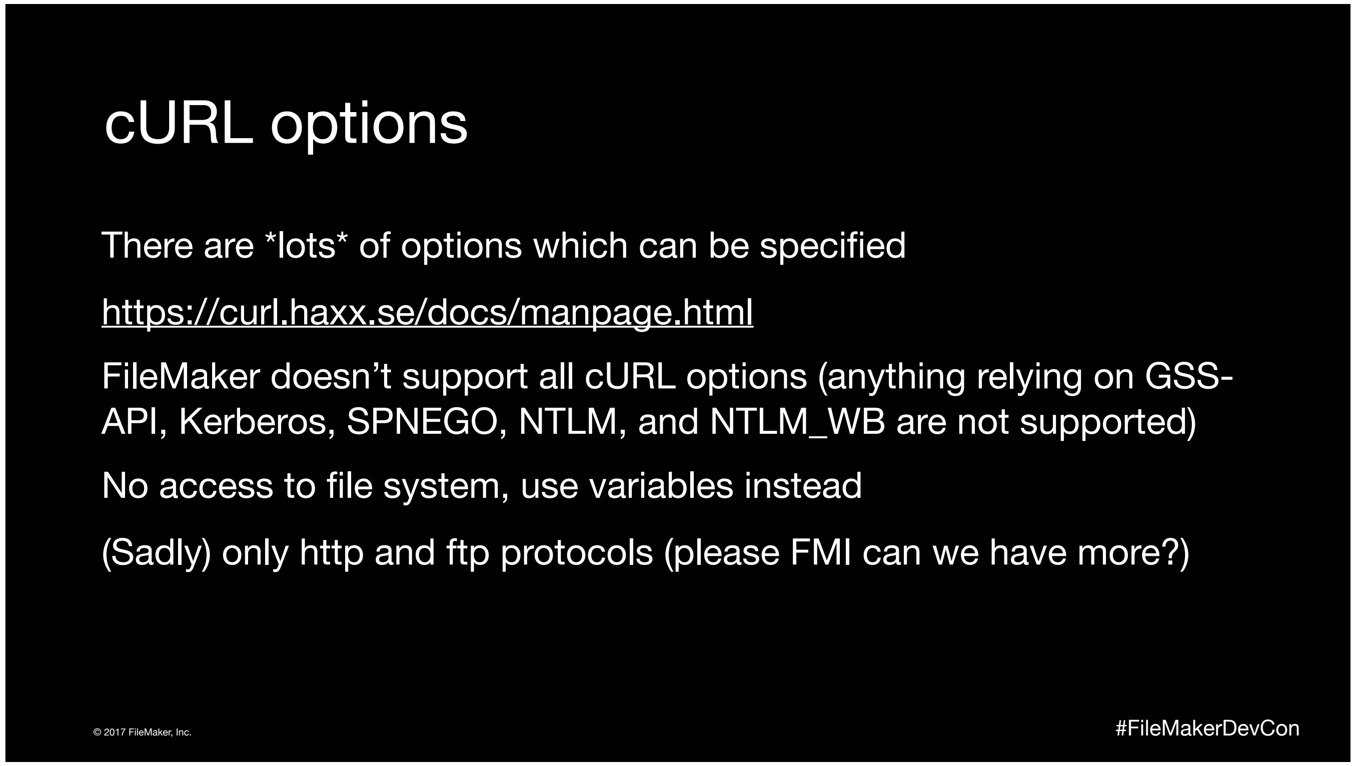
Limited to ‘simple’ use-cases with only query string access

© 2017 FileMaker, Inc.

cURL has been used ‘under the hood’ since FM introduced InsertFromURL in v12, supporting http/https, ftp/ftps and GET.

FileMaker 13 introduced httppost and httpspost as a way to allow POST access in FM 13

Until FM16 we had to use plugins if we wanted to do anything more complex than send get/post with url parameters, which meant anything which needed custom headers (particularly) wasn’t possible.



cURL options

There are *lots* of options which can be specified

https://curl.haxx.se/docs/manpage.html

FileMaker doesn't support all cURL options (anything relying on GSS-API, Kerberos, SPNEGO, NTLM, and NTLM_WB are not supported)

No access to file system, use variables instead

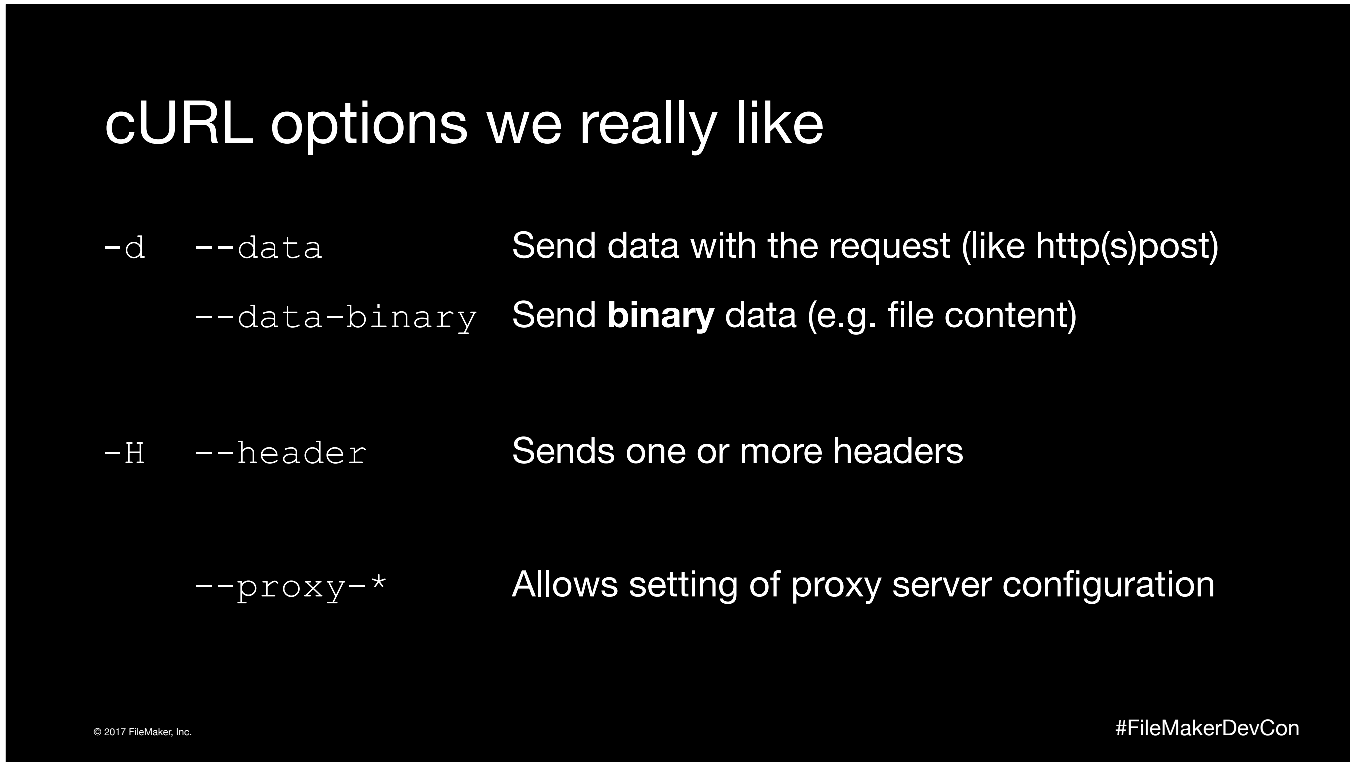
(Sadly) only http and ftp protocols (please FMI can we have more?)

© 2017 FileMaker, Inc.

All unsupported options pertain to particular security services, so for many users this won't be an issue - if you need those, then unfortunately you're still going to have to look for other ways.

No access to the file system could be an important consideration, however in pretty much every situation where you may traditionally need that, FM have provided a 'workaround' which lets us use fields or variables instead - we're going to see quite a lot of cases what that happens.

For reference:
GSS-API: The Generic Security Service Application Program Interface (GSSAPI, also GSS-API) is an application programming interface for programs to access security services.
Kerberos: Is a network authentication protocol designed to provide strong authentication for client/server applications by using secret-key cryptography
SPNEGO: a standard specification defined in The Simple and Protected GSS-API Negotiation Mechanism
NTLM: a suite of Microsoft security protocols that provides authentication, integrity, and confidentiality to users.
NTLM_WB: extension of NTLM



cURL options we really like

`-d`	`--data`	Send data with the request (like http(s)post)
	`--data-binary`	Send **binary** data (e.g. file content)
`-H`	`--header`	Sends one or more headers
	`--proxy-*`	Allows setting of proxy server configuration

© 2017 FileMaker, Inc.

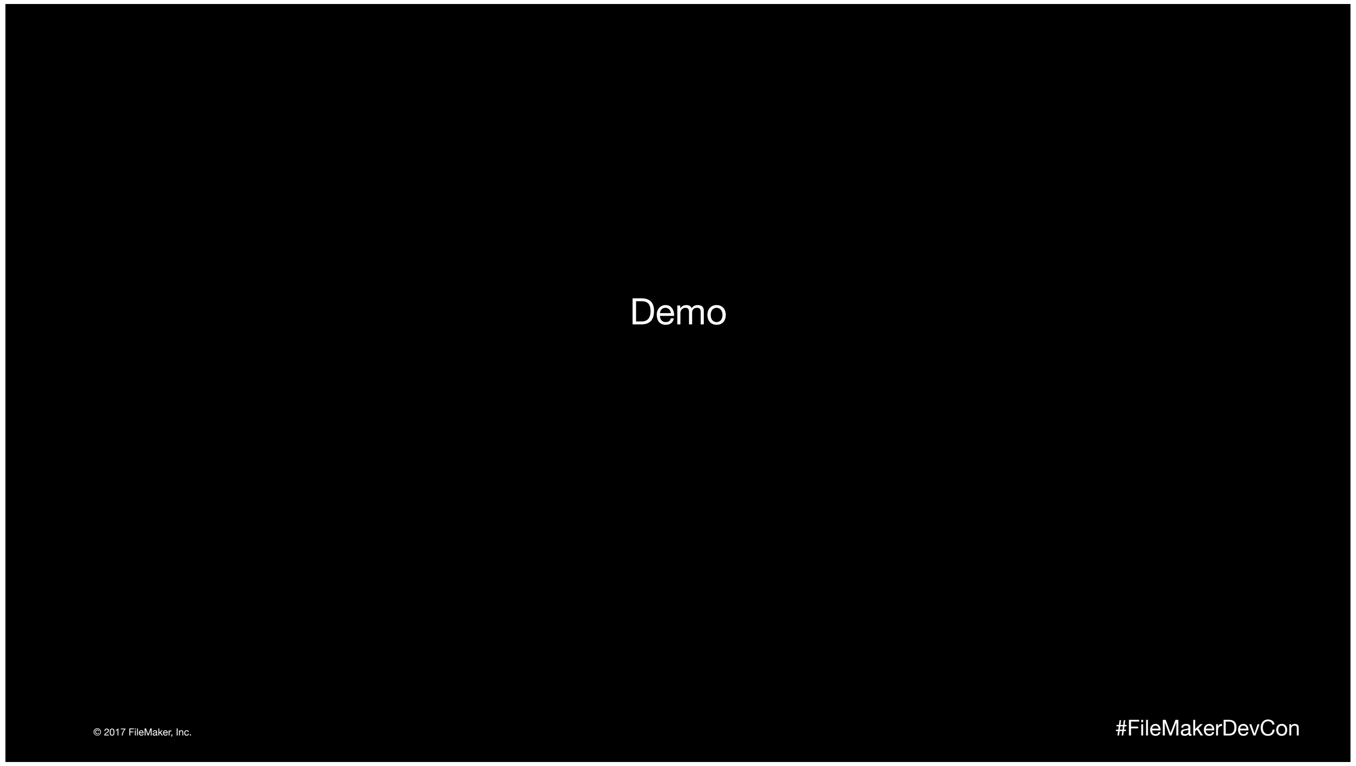
#FileMakerDevCon

Some options have shortcuts e.g. -d or -H while others are only available in their ‘full form’, e.g. —data-binary. IMPORTANT NOTE: although it may not look like it in these slides the characters before the ‘full form’ options are two hyphens.

Some options, like the —header option can be called multiple times in order to ‘repeat’ the same action, others like —data will override the previous value set

—proxy-* isn’t an actual option, but there are range of options which begin with that prefix and allow things like proxy server address, port, username, password etc to be set.

In my view just having access to the —header option is a huge win since it unlocks access to a whole range of APIs which we couldn’t previously access natively within FM because the API **requires** that one or more headers (often for authentication) be set.



Demo

© 2017 FileMaker, Inc.

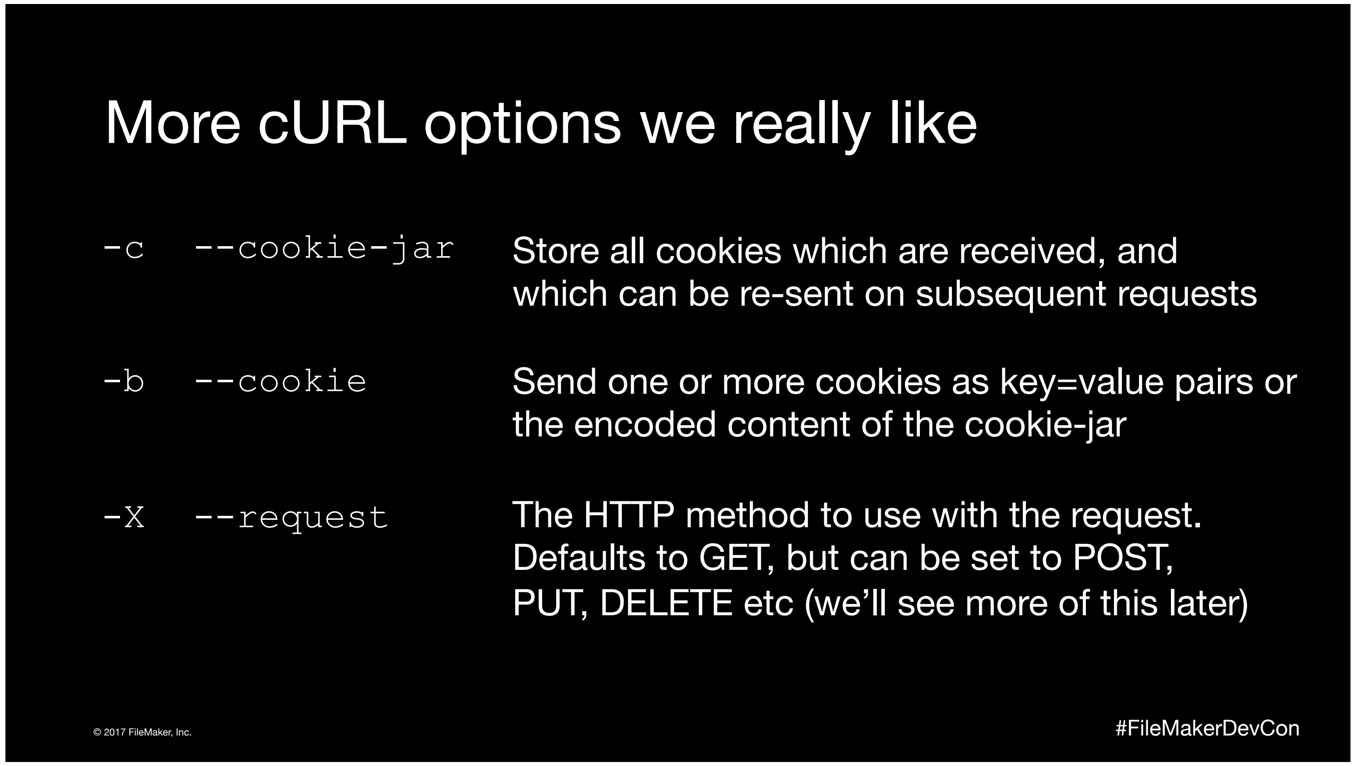
#FileMakerDevCon

I’ve tried to make my demos a bullet-proof as possible. I’m running as many of the services as possible locally, I’ve got my FMS on the MacMini here, I’ve tethered my phone to my laptop to mirror that for a couple of demos, but because this session is about APIs, some of them by nature are external - so, keep your fingers crossed when we get to the trickier ones!

Everything that I demo also works in the demo file (which you’ve hopefully downloaded) and will work for you once you’re back home for the foreseeable future. Please keep in mind that to make this possible I’ve embedded keys, tokens, and credentials in this file, which mean that *everyone* is using the same Google account, and the same Dropbox account - so remember that this is a family show - if people are uploading inappropriate content then I’ll have to disable those tokens and keys and that would be a shame for everyone!

Note: Two-part demo, I have a pseudo-slide in the FM file to avoid having to switch back to Keynote to show a single slide.

Uses ‘Intro’ scripts



More cURL options we really like

`-c   --cookie-jar`   Store all cookies which are received, and which can be re-sent on subsequent requests

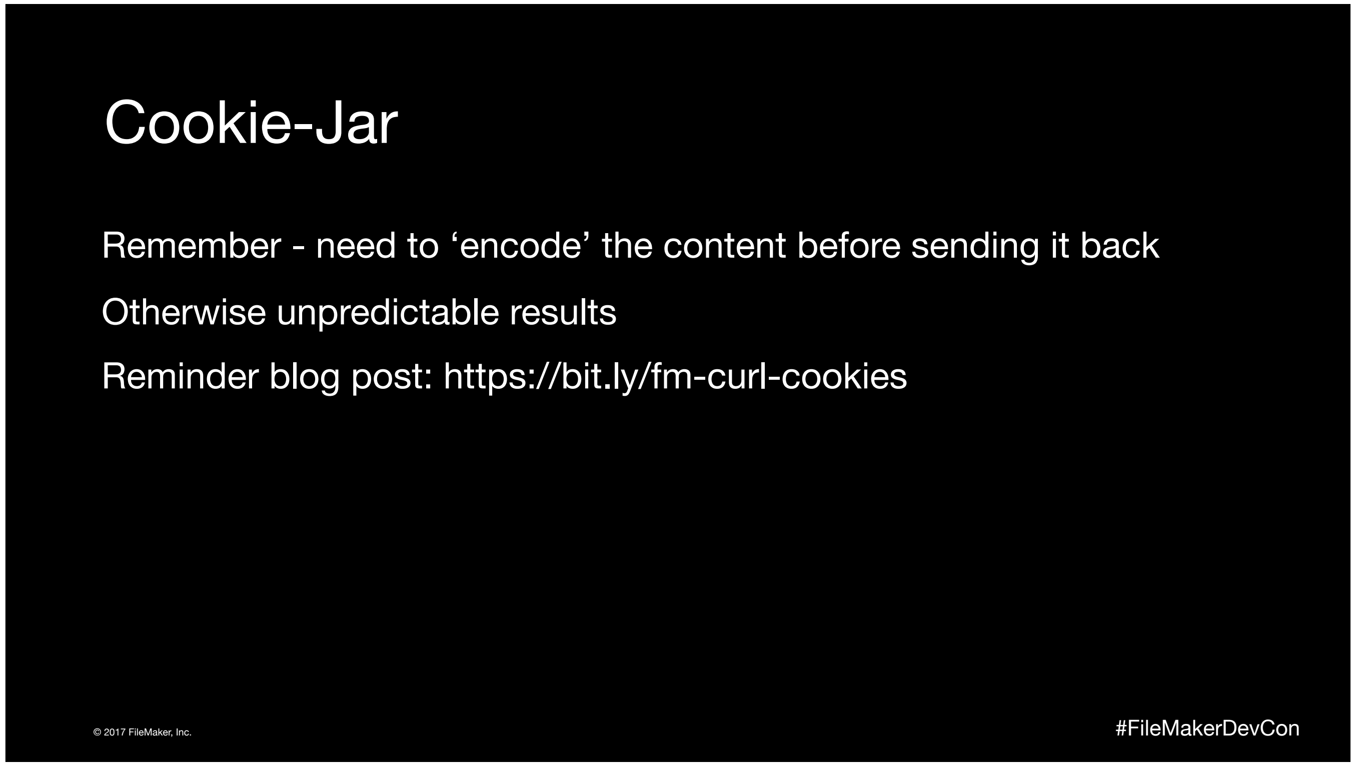
`-b   --cookie`   Send one or more cookies as key=value pairs or the encoded content of the cookie-jar

`-X   --request`   The HTTP method to use with the request. Defaults to GET, but can be set to POST, PUT, DELETE etc (we'll see more of this later)

© 2017 FileMaker, Inc.

The cookie jar is also extremely useful – some APIs will set cookies after initial authentication, and require that those cookies be sent along with subsequent queries, often for authentication purposes.

The cookie jar is one of the places where cURL would usually write a file to disk, but instead we use a field or global variable to store the received data.



Cookie-Jar

Remember - need to ‘encode’ the content before sending it back

Otherwise unpredictable results

Reminder blog post: https://bit.ly/fm-curl-cookies

© 2017 FileMaker, Inc.

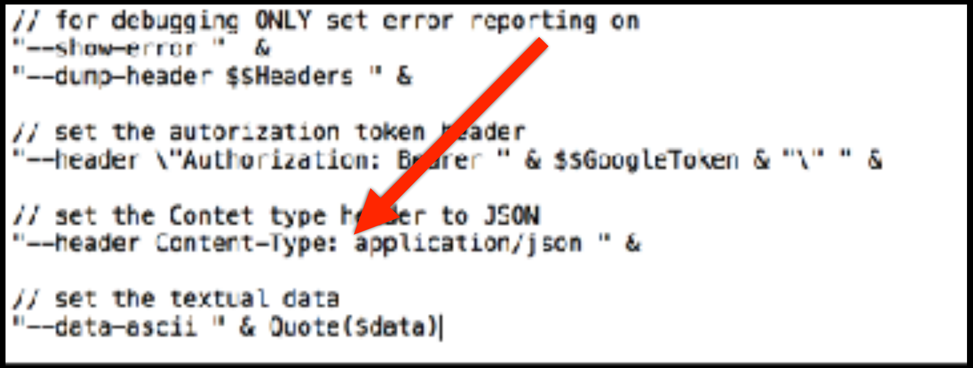
Simple substitute function

```
        Substitute ( $$CookieJar ; "¶" ; Char(10) )
```

Must be set as a variable and passed into the options string as same

```
        "--cookie $CookieString"
```

A note of caution



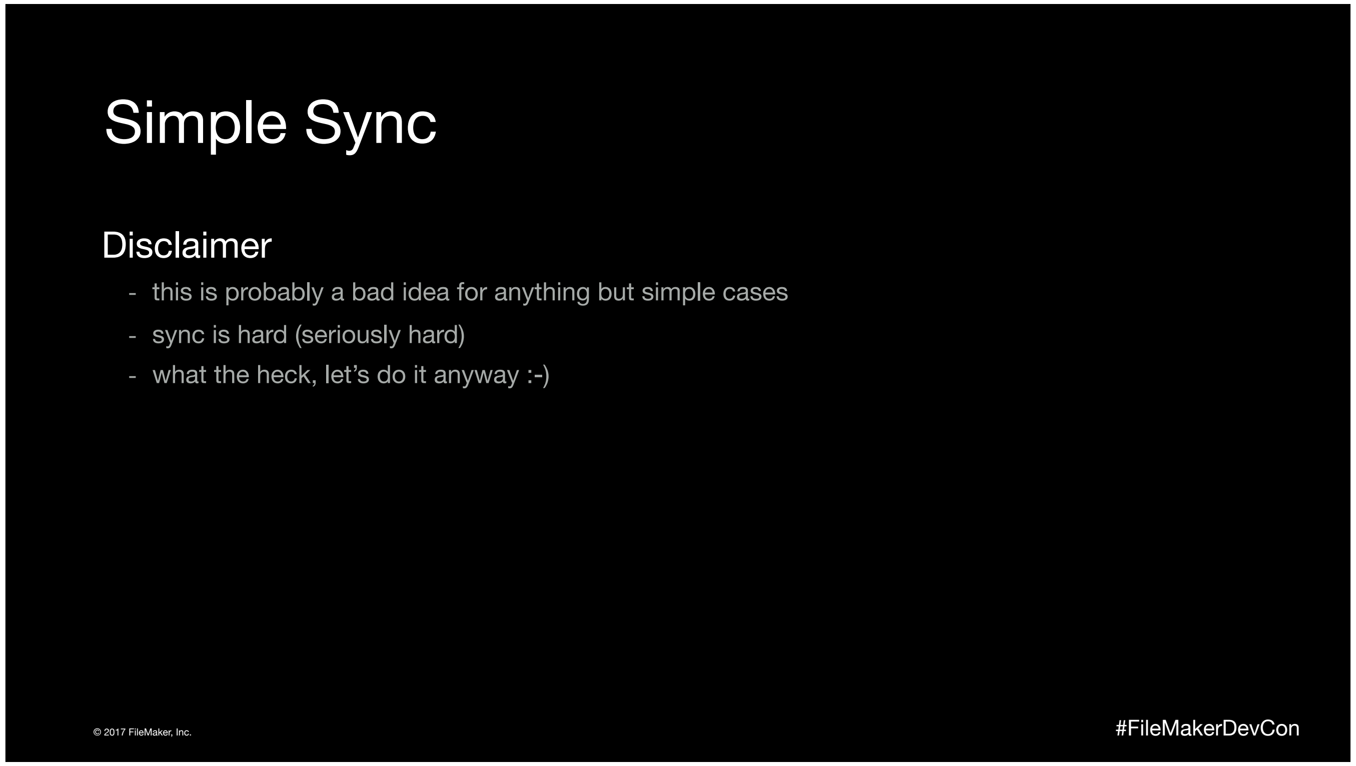
```
// for debugging ONLY set error reporting on
"--show-error "  &
"--dump-header $$Headers " &

// set the autorization token header
"--header \"Authorization: Bearer " & $$GoogleToken & "\" " &

// set the Contet type header to JSON
"--header Content-Type: application/json " &

// set the textual data
"--deta-ascii " & Quote($data)
```

“--header \”Content-Type: application/json\””

© 2017 FileMaker, Inc.

Spaces in the options string have ‘special meaning’ to the parser – they mean that another instruction is coming right up. So although these two lines look like they are identical, the space between the colon and ‘application’ result in the second line failing to work as expected. Either ensure continuous strings, or wrap your headers (and other such commands) in escaped quotes, or use the Quote() function.

And yes – I have learnt this the hard way – it’s actually much easier to see that space up there on the big screen than it is in the Script Workspace.



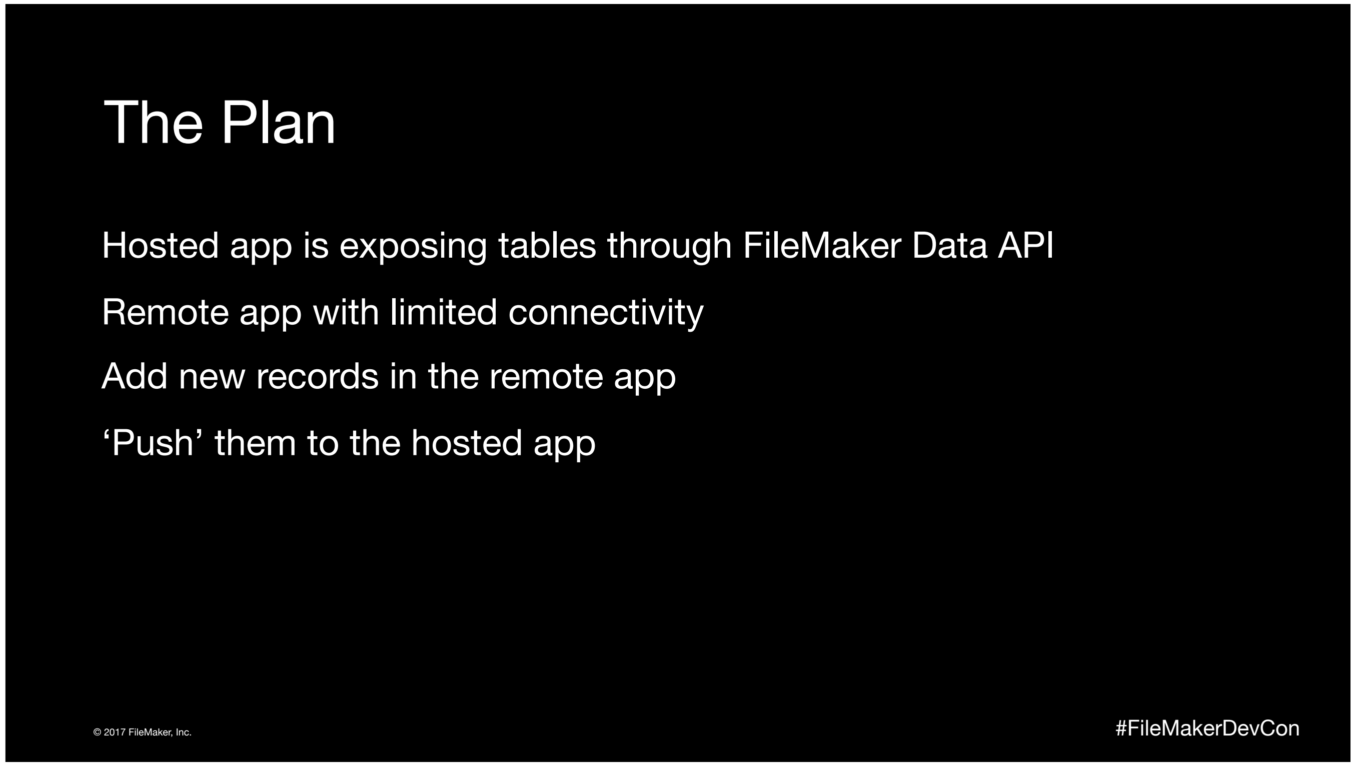
Simple Sync

Disclaimer

- this is probably a bad idea for anything but simple cases
- sync is hard (seriously hard)
- what the heck, let’s do it anyway :-)

© 2017 FileMaker, Inc.

So if I say it’s a bad idea, why am I doing a demo of it?
- it’s a relatively simple example to demonstrate
- it ‘requires’ two new features supported by cURL in InsertFromURL
- it introduces you to the FileMaker Data API, another “cool new thing” (TM)
- it introduces you to using JSON in FileMaker, even more coolness!



The Plan

Hosted app is exposing tables through FileMaker Data API

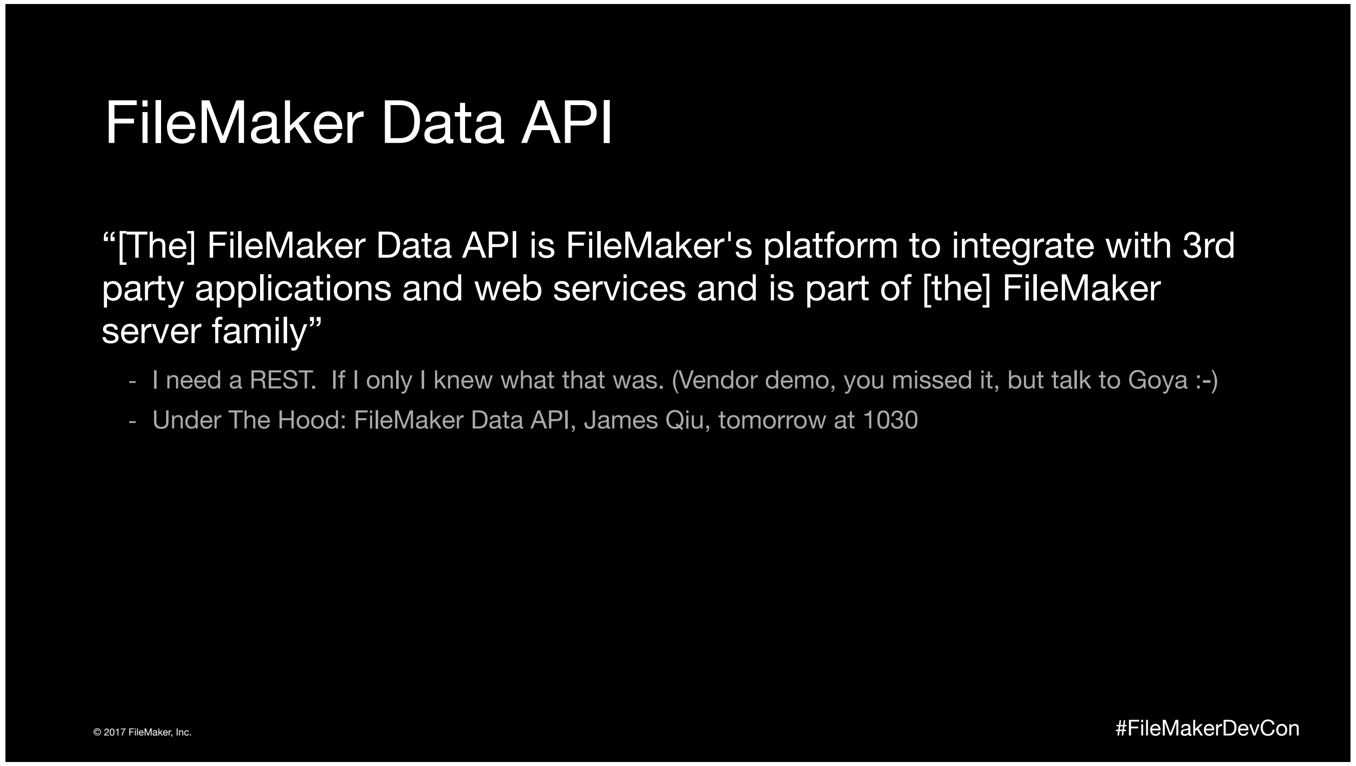
Remote app with limited connectivity

Add new records in the remote app

‘Push’ them to the hosted app

© 2017 FileMaker, Inc.

Although I said it’s not a great idea I can see real-world examples of where this might actually be a good idea
 - limited connectivity (poor data coverage), connecting to an API requires far lass bandwidth than connecting to FMS
 - it’s quick (with the ‘right’ sort of data), even with poor connectivity
 - avoids potential firewall issues on WiFi networks, e.g. if port 5003 is blocked, since uses standard https



FileMaker Data API

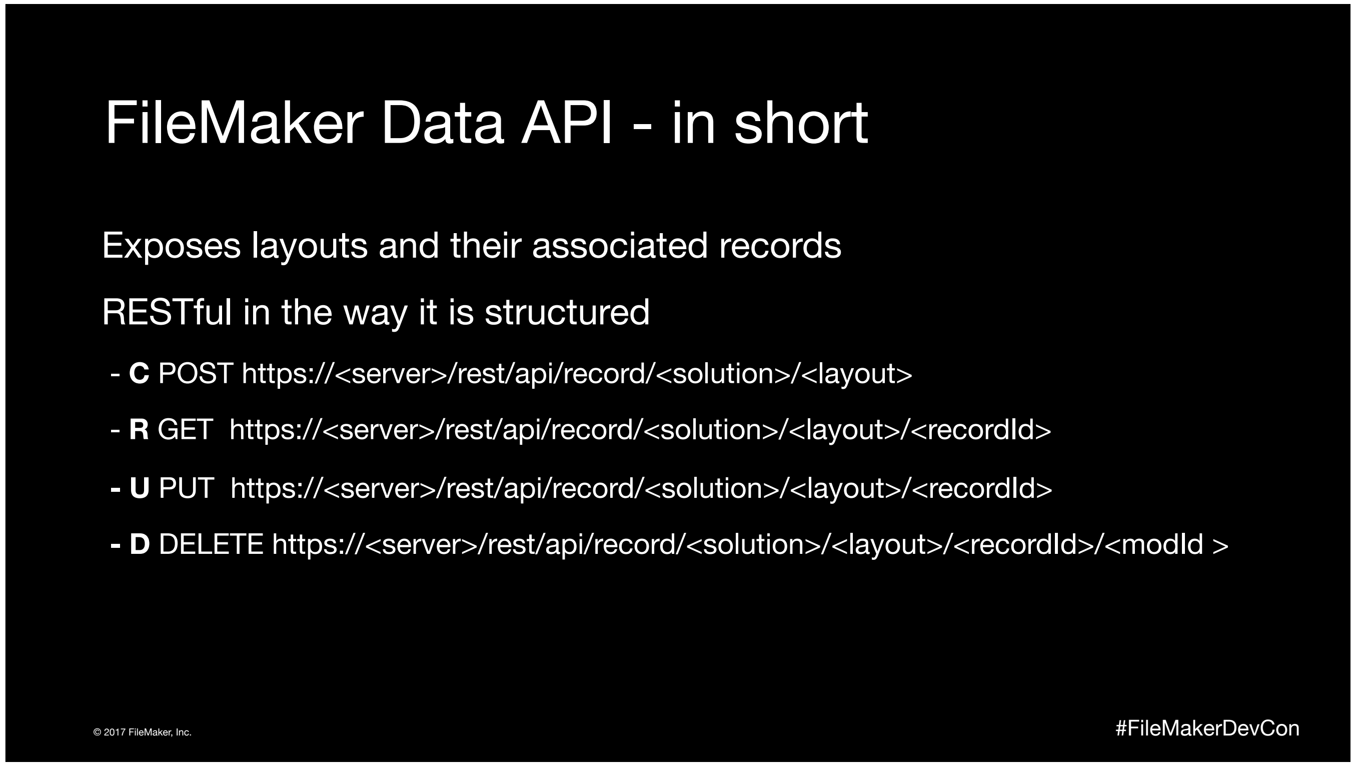
“[The] FileMaker Data API is FileMaker's platform to integrate with 3rd party applications and web services and is part of [the] FileMaker server family”

- I need a REST. If I only I knew what that was. (Vendor demo, you missed it, but talk to Goya :-)
- Under The Hood: FileMaker Data API, James Qiu, tomorrow at 1030

© 2017 FileMaker, Inc.

Overview of James’ session
The introduction of the FileMaker Data API allows customers to connect their FileMaker data to other systems using JSON and REST, enabling integration of FileMaker into their workflow.

In this session, you'll start with the architecture as well as an overview of the FileMaker Data API, then proceed to code examples that illustrate how to utilize the FileMaker Data API.



FileMaker Data API - in short

Exposes layouts and their associated records

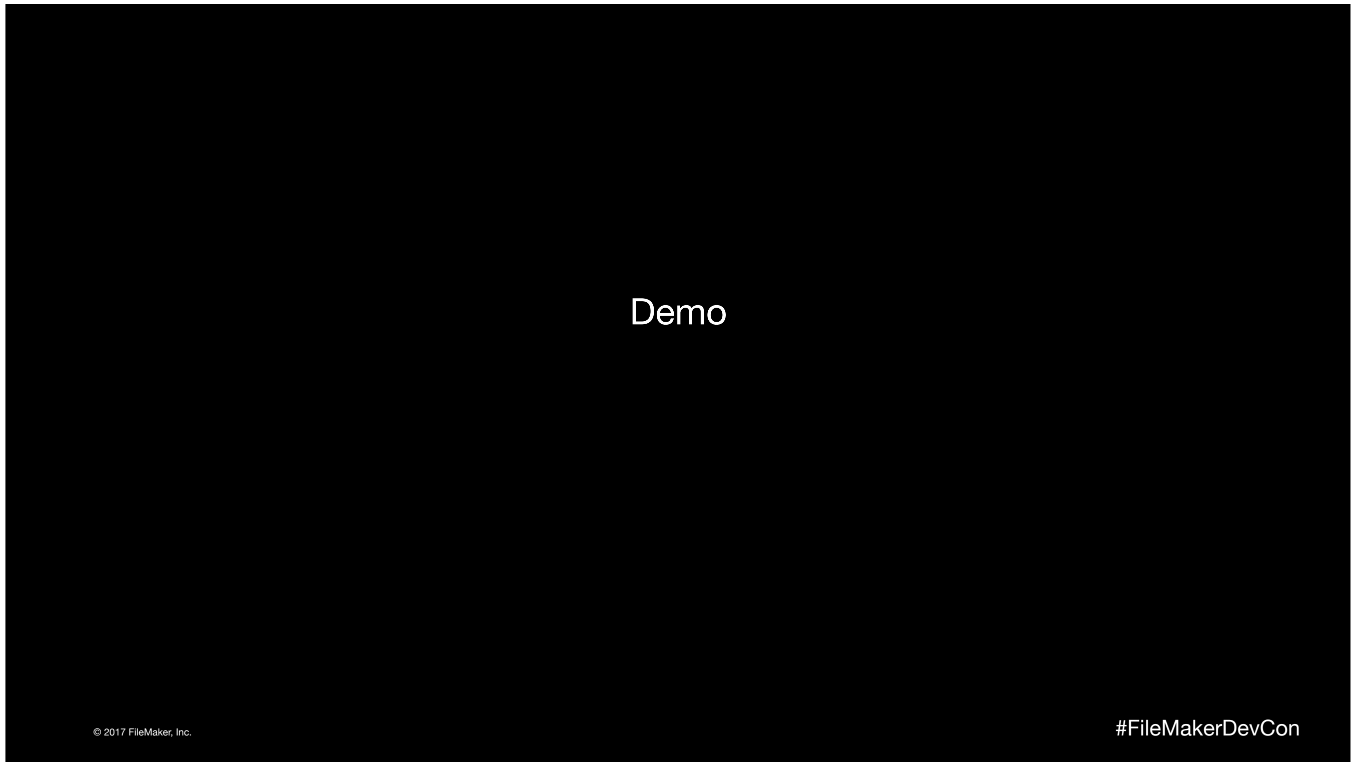
RESTful in the way it is structured

- **C** POST https://<server>/rest/api/record/<solution>/<layout>
- **R** GET https://<server>/rest/api/record/<solution>/<layout>/<recordId>
- **U** PUT https://<server>/rest/api/record/<solution>/<layout>/<recordId>
- **D** DELETE https://<server>/rest/api/record/<solution>/<layout>/<recordId>/<modId >

© 2017 FileMaker, Inc.

Note that Create and Read use familiar http methods
 - Get as used in a browser to load a page, or with the original InsertFromURL script step
 - Post like in a browser when submitting a form, or using (now deprecated) httpspost in InsertFromURL

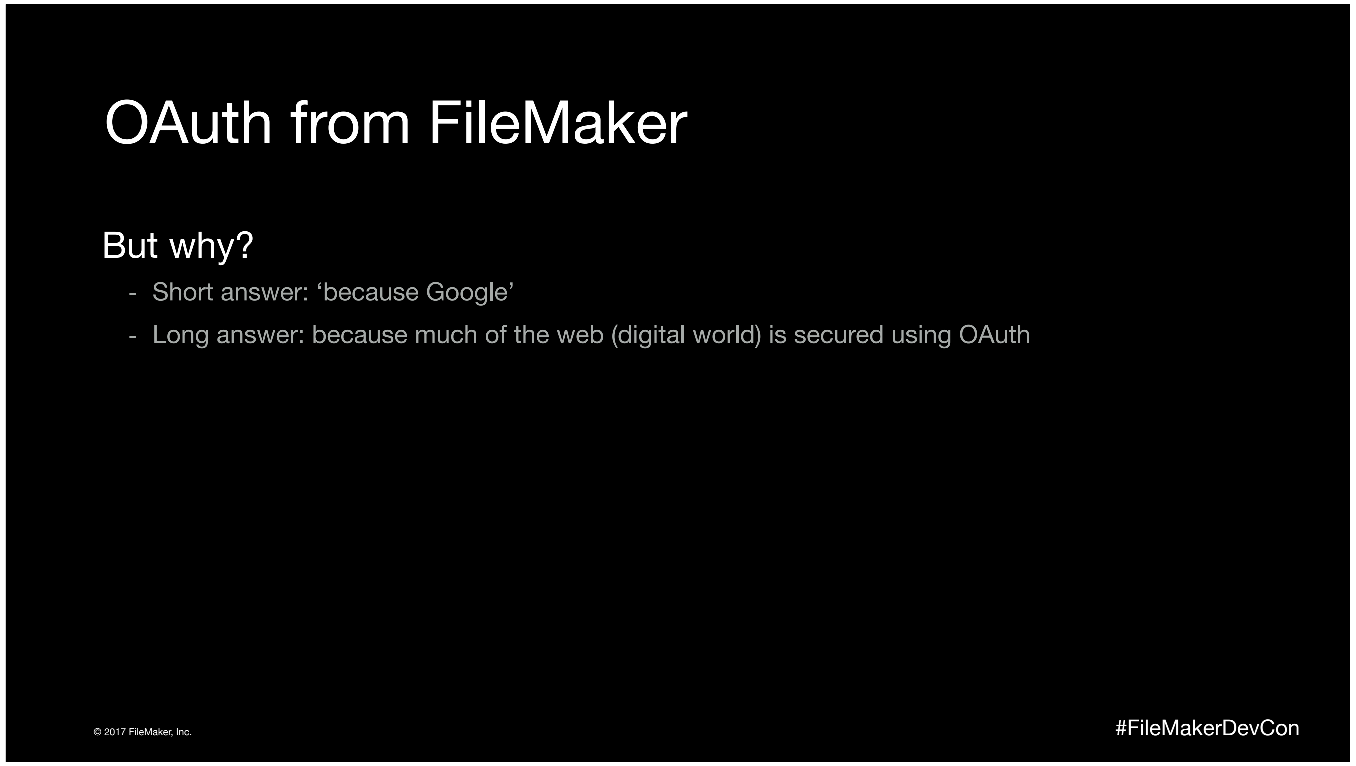
Update and delete require alternative methods which are newly available to us in FM16



Demo

© 2017 FileMaker, Inc.

Data API scripts



OAuth from FileMaker

But why?

- Short answer: ‘because Google’
- Long answer: because much of the web (digital world) is secured using OAuth

© 2017 FileMaker, Inc.

As an aside, FileMaker 16 now supports OAuth for access to apps. Unfortunately there’s no real way to ‘hook into’ that functionality for anything other than app access security which means that there are some situations where we still can’t make full use of APIs requiring OAuth because there’s no way to receive a ‘callback’ from the OAuth provider which many require.

So what is OAuth?

> “OAuth is an open standard for access delegation, commonly used as a way for Internet users to grant websites or applications access to their information on other websites but without giving them the passwords.”

Wikipedia

© 2017 FileMaker, Inc.

Adding events to Google Calendar

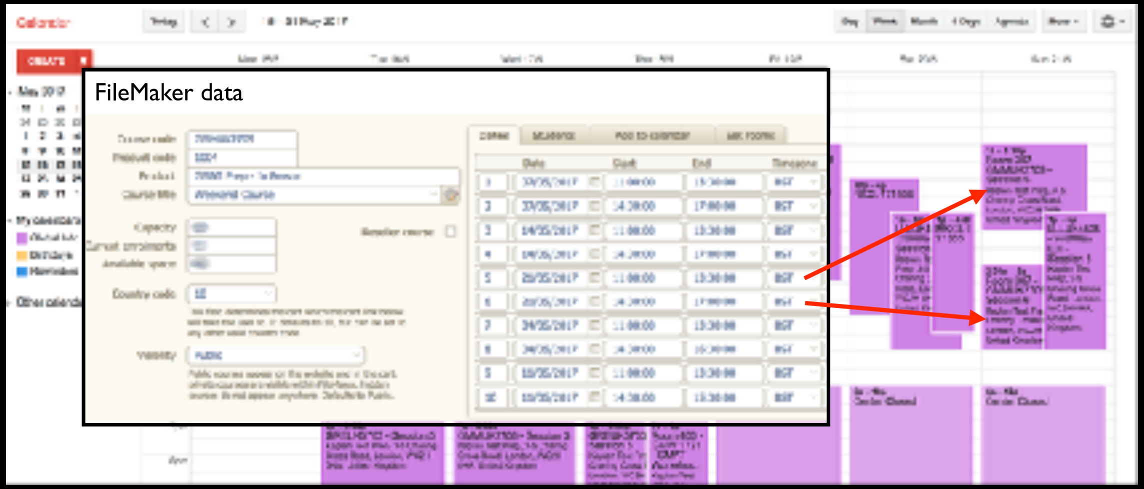
FileMaker solution for course management

Shared Google Calendar

Wanted course sessions in the shared calendar

© 2017 FileMaker, Inc.

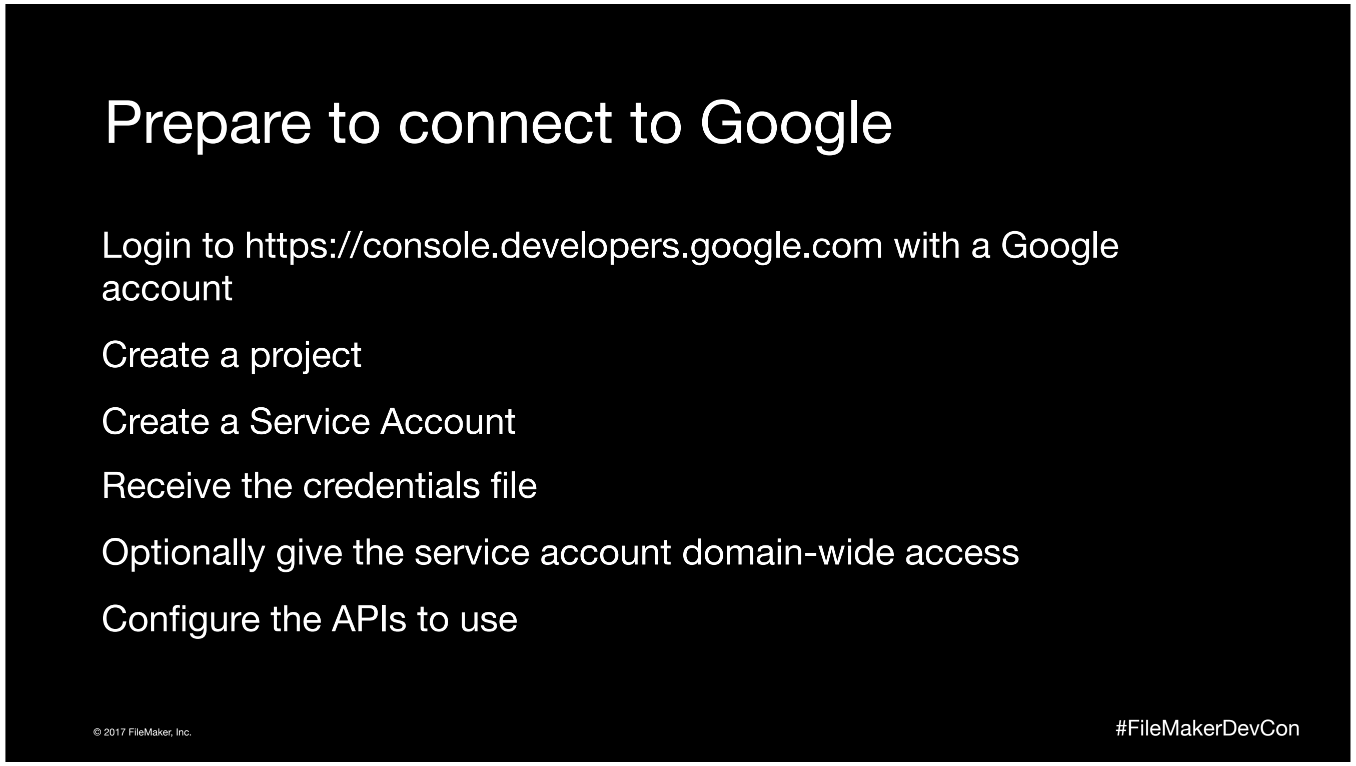
Adding events to Google Calendar



© 2017 FileMaker, Inc.

#FileMakerDevCon

So this is what we wanted to achieve, with the calendar being updated whenever a change was made in FM. At this stage it's not bidirectional – so changes made in Google Calendar aren't reflected back into FM.



Prepare to connect to Google

Login to https://console.developers.google.com with a Google account

Create a project

Create a Service Account

Receive the credentials file

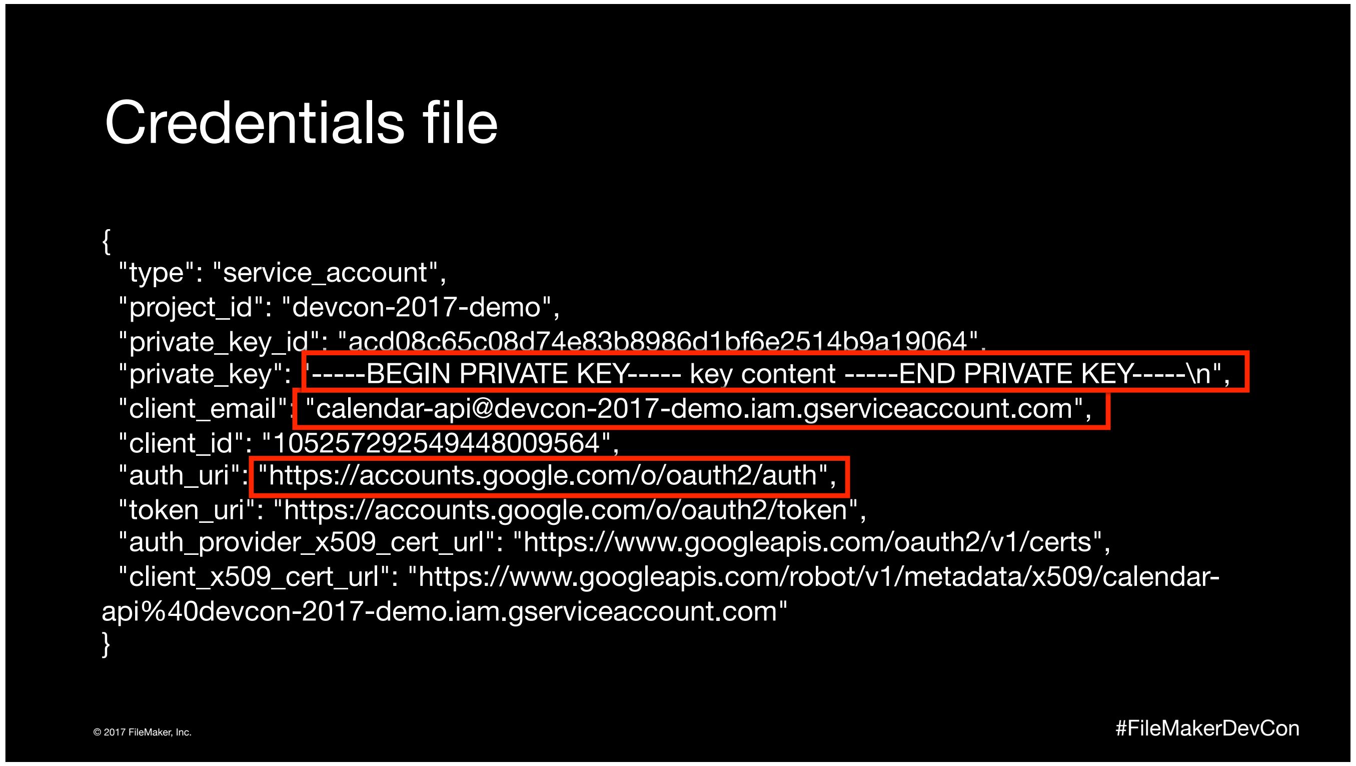
Optionally give the service account domain-wide access

Configure the APIs to use

© 2017 FileMaker, Inc.

This is a summary of the steps from https://developers.google.com/identity/protocols/OAuth2ServiceAccount

If you are using Google for domain services, e.g. for you organisations calendaring, email, etc then you can grant the service account access to all accounts in your domain - this can be useful if you want to be able to manage the individual calendars, or other Google content of people in your domain.



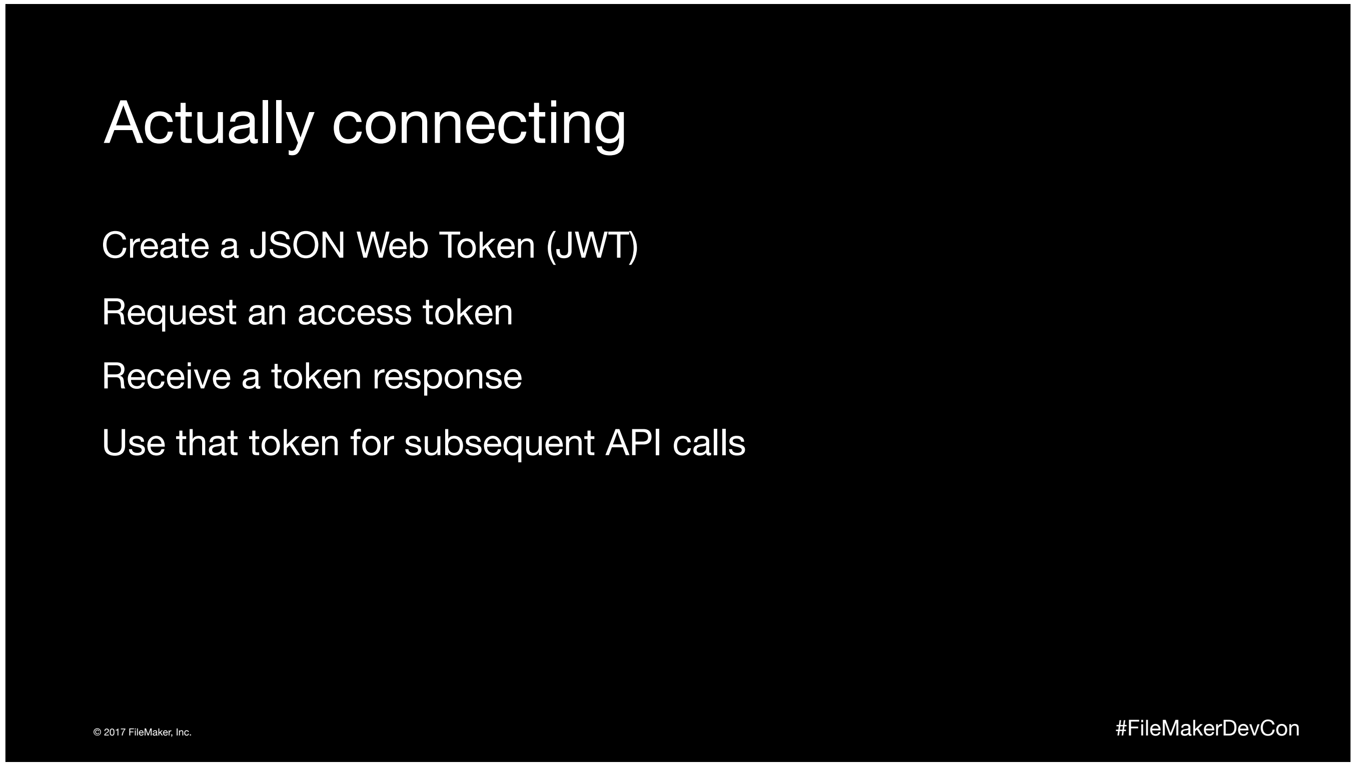
Credentials file

```
{
  "type": "service_account",
  "project_id": "devcon-2017-demo",
  "private_key_id": "acd08c65c08d74e83b8986d1bf6e2514b9a19064",
  "private_key": "-----BEGIN PRIVATE KEY----- key content -----END PRIVATE KEY-----\n",
  "client_email": "calendar-api@devcon-2017-demo.iam.gserviceaccount.com",
  "client_id": "105257292549448009564",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://accounts.google.com/o/oauth2/token",
  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
  "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/calendar-
api%40devcon-2017-demo.iam.gserviceaccount.com"
}
```

© 2017 FileMaker, Inc.

As you can see, it’s (another) chunk of JSON - aren’t we glad we got native JSON parsing in FM 16!

As well as the the username, there’s the private key which is needed for request signing, and the URLs which authentication requests get made to.



Actually connecting

Create a JSON Web Token (JWT)

Request an access token

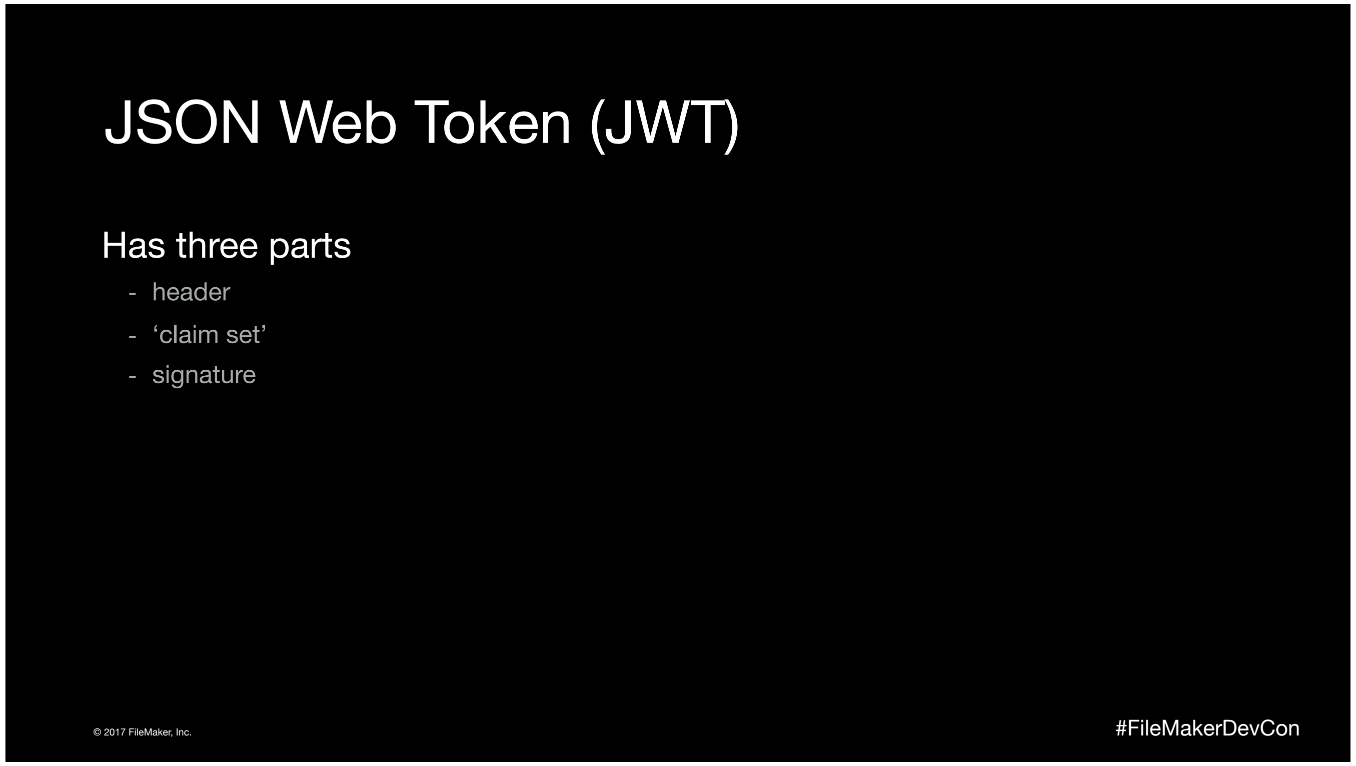
Receive a token response

Use that token for subsequent API calls

© 2017 FileMaker, Inc.

JSON Web Token (pronounced ‘Jot’) is a mechanism for securely accessing content without needing to transmit a username or password in what is essentially clear text (though it may not look like it) - more on that in a moment

Once the JWT is accepted, a token is returned (much like what happens with the FileMaker Data API), which must then be sent on subsequent requests in a header.



JSON Web Token (JWT)

Has three parts

- header
- ‘claim set’
- signature

© 2017 FileMaker, Inc.

The header is the same every time - it specifies the encryption which is being used in the signature, but as theres only one option, SHA256, it’s always that :-)

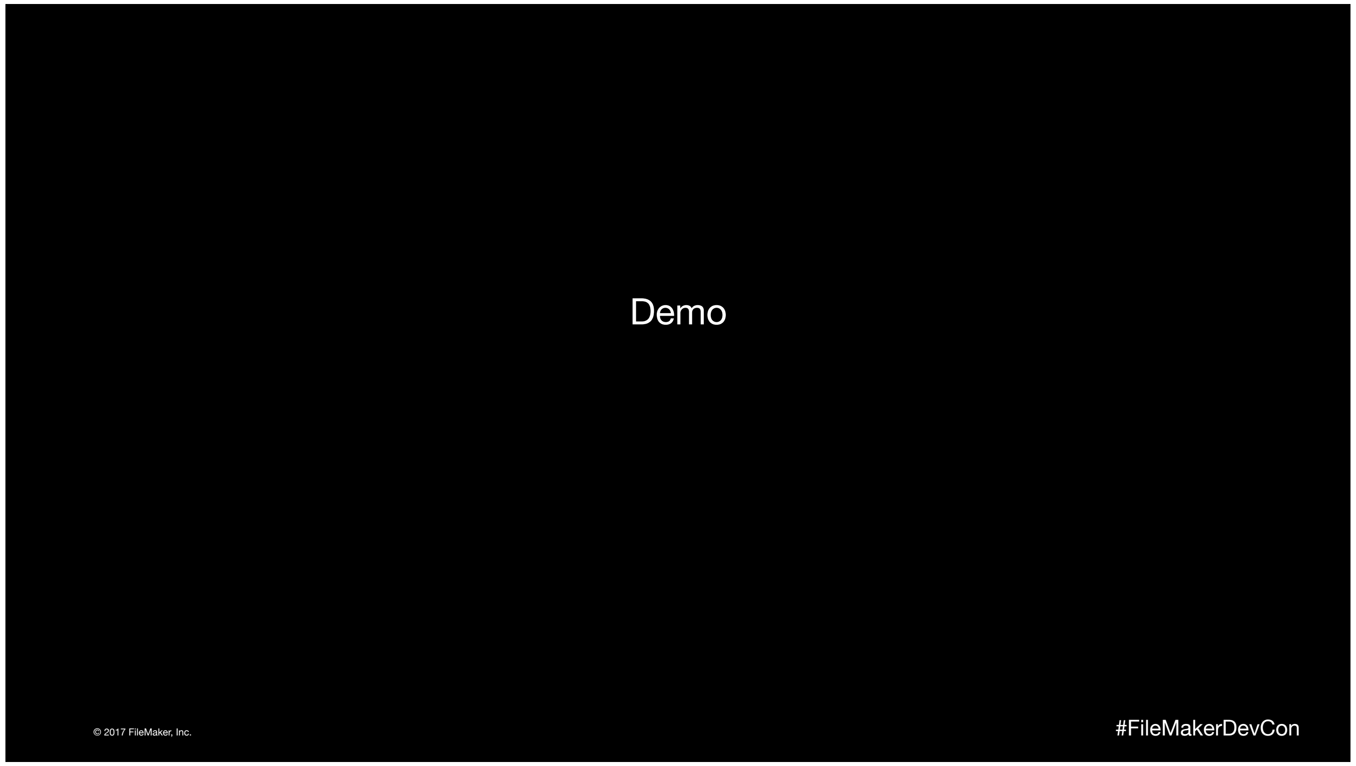
The claim set includes the account name (of that service worker account you created), as well as the ‘scope’ (i.e. the APIs that you wish to make use of, which you have enabled), two timestamps (the request time, and the expiry time of the returned token, max 1 hour from request time), and some other miscellaneous (static) data.

The signature is a combination of the base64 encoded header and claim set, which is then signed with an RSA private key.

And this is where I had to cheat - although FM 16 is capable of performing a SHA256 hash, using the new CryptHash calculation function, it can’t also RSA sign that, so we needed something else to do that - since this session is all about API integration and cURL parameters, I constructed a simple API which will do that for us. All very ‘meta’ an API that is used to sign credentials to allow you to access an API.

[
In order for this API to sign the data string it needs your private key. Is sending your private key to an API a good idea? probably not if you don’t own that API, since the owner of the API could be capturing your private key when you submit it, and then do something nefarious with it. If you ‘own’ the API, it’s on your network, you only communicate with it via https - it’s probably not the end of the world as far as security concerns go (though I imagine Stephen Blackwell may disagree with me on that ;-)

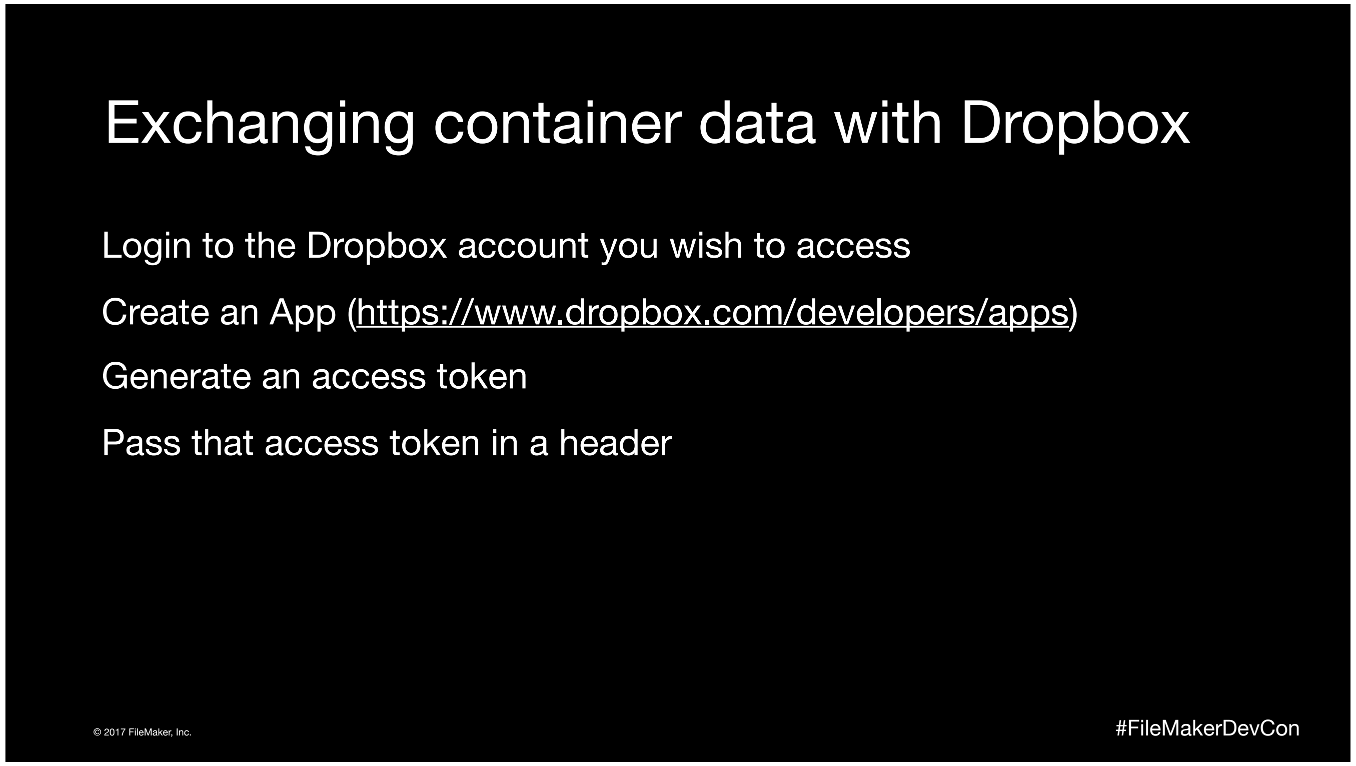
The code for the API is attached to the blog post with all the other resources for this session if you’re interested in hosting it yourself. https://bit.ly/devcon-2017-cURL
]



Demo

© 2017 FileMaker, Inc.

Uses the Google scripts
 - Perform authentication to fetch a token (this is where the really interesting stuff happens)
 - Create a calendar entry (once authenticated, this is trivial ;-)



Exchanging container data with Dropbox

Login to the Dropbox account you wish to access

Create an App (https://www.dropbox.com/developers/apps)

Generate an access token

Pass that access token in a header

© 2017 FileMaker, Inc.

#FileMakerDevCon

On the surface of it this appears as though this process is exactly the same as the other APIs which we've worked with, but there are a couple of complexities, and a couple more cURL options which are needed for this to work.

Of note is the use of JSON in headers, and the sending of binary data using the —data-binary cURL option.

[This slide is included in the demo file to prevent switching back to Keynote to show a single slide]

Demo

© 2017 FileMaker, Inc.

Questions?

© 2017 FileMaker, Inc.

Session Updates

This session **does** have updates.

www.filemaker.com/devcon/speaker_updates

(This is also listed in DevCon2Go)

© 2017 FileMaker, Inc.

Thank You!

Please remember to fill out the session evaluation at:
www.filemaker.com/devcon/evaluations

© 2017 FileMaker, Inc.

FILEMAKER DEVCON 2017
JULY 24 - 27, 2017 | PHOENIX, AZ

#FileMakerDevCon

Please do not delete or change this slide.