No REST API for the Wicked:

Scripting FileMaker Made Me a Better MSP

precursor.ca/slides

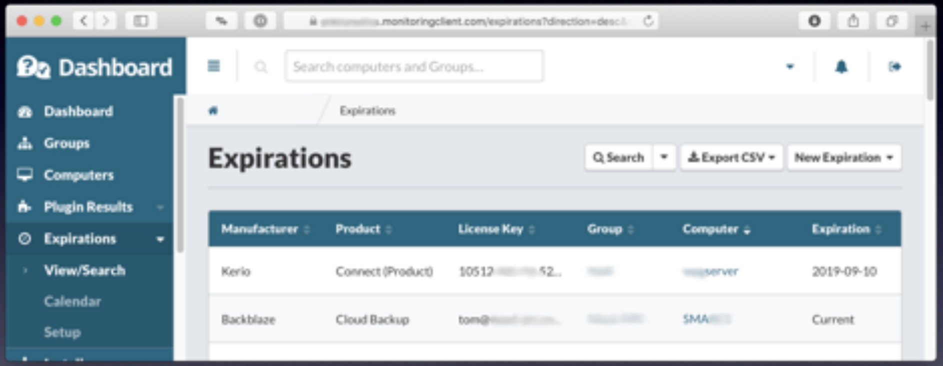
I’m Alex Narvey and today I am going to show you how I used FileMaker to make my MSP life a little easier. I’m going to race through this preso so don’t worry about writing things down. Just get the slides from precursor.ca/slides
Lets start with No REST API for the Wicked…
CLICK

Mac mini MSP Stack

precursor.ca/slides

Many MSPs use Watchman Monitoring as a basic part of their tool Stack. It is so useful.
CLICK

MAC
DEV
OPS
YVR



[redacted].monitoringclient.com/expirations?direction=desc

Expirations

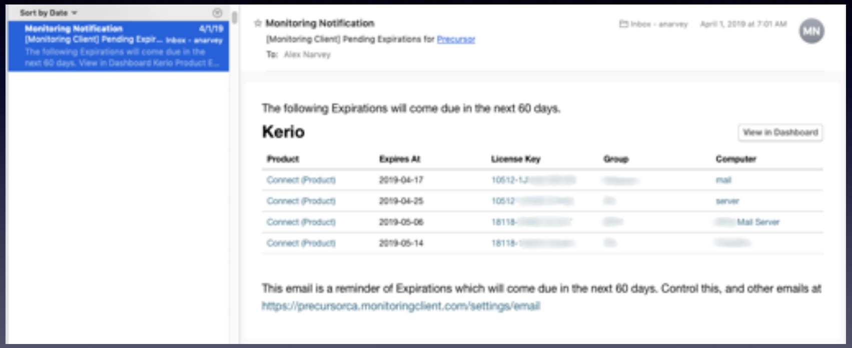
Expirations

Manufacturer	Product	License Key	Group	Computer	Expiration
Kerio	Connect (Product)	10512[redacted]52...	[redacted]	[redacted]server	2019-09-10
Backblaze	Cloud Backup	tom@[redacted]	[redacted]	SMA[redacted]	Current

precursor.ca/slides

One of my favourite things is that it automatically notifies you about expirations like for BackBlaze, CrashPlan, Kerio Connect, and other software subscriptions.
CLICK

MAC DEV OPS YVR



Sort by Date
Monitoring Notification 4/1/19
[Monitoring Client] Pending Expir... Inbox - anarvey
The following Expirations will come due in the next 60 days. View in Dashboard Kerio Product E...

☆ Monitoring Notification

[Monitoring Client] Pending Expirations for Precursor

To: Alex Narvey

Inbox - anarvey April 1, 2019 at 7:01 AM MN

The following Expirations will come due in the next 60 days.

Kerio

Product	Expires At	License Key	Group	Computer
Connect (Product)	2019-04-17	10512-1J[redacted]	[redacted]	mail
Connect (Product)	2019-04-25	10512[redacted]	[redacted]	server
Connect (Product)	2019-05-06	18118-[redacted]	[redacted]	[redacted] Mail Server
Connect (Product)	2019-05-14	18118-[redacted]	[redacted]	[redacted]

This email is a reminder of Expirations which will come due in the next 60 days. Control this, and other emails at https://precursorca.monitoringclient.com/settings/email

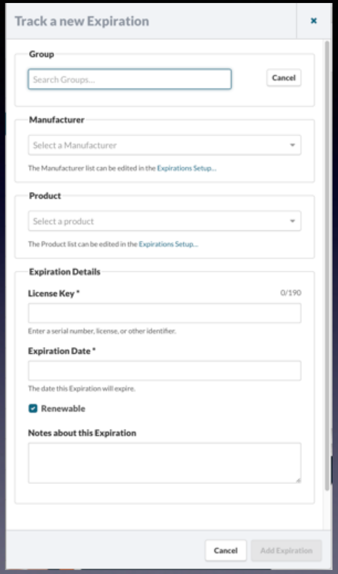
precursor.ca/slides

You don’t have to do anything, the Watchman agent just emails you these lovely notifications automatically. And Watchman can also do this for things like Domain Registrations and SSL Certificates.

CLICK

Manually enter:

- Search a Group name
- Select a Manufacturer*
- Select a Product*
- Fill in Certificate Name
- Fill in Expiration Date


Track a new Expiration

Group

Search Groups...

Manufacturer

Select a Manufacturer

The Manufacturer list can be edited in the Expirations Setup...

Product

Select a product

The Product list can be edited in the Expirations Setup...

Expiration Details

License Key * 0/190

Enter a serial number, license, or other identifier.

Expiration Date *

The date this Expiration will expire.

☑ Renewable

Notes about this Expiration

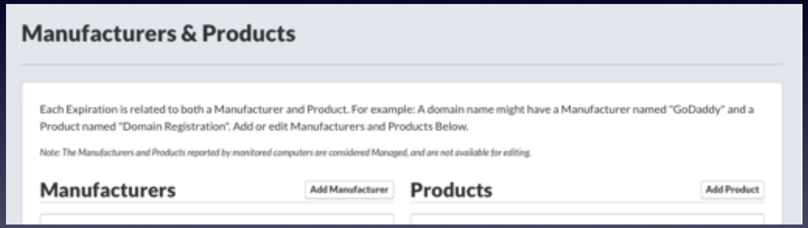
“Like an animal !”

Both for Domains and Certificates you have to enter all the information manually…

“Like an animal!”

CLICK

Registrars and *Issuers* have to be manually added to Manufacturers



Manufacturers & Products

Each Expiration is related to both a Manufacturer and Product. For example: A domain name might have a Manufacturer named "GoDaddy" and a Product named "Domain Registration". Add or edit Manufacturers and Products Below.

Note: The Manufacturers and Products reported by monitored computers are considered Managed, and are not available for editing.

precursor.ca/slides

And you can’t even select the Manufaturer without having added it first on a separte pane.
CLICK

MAC
DEV
OPS
YVR

🤬

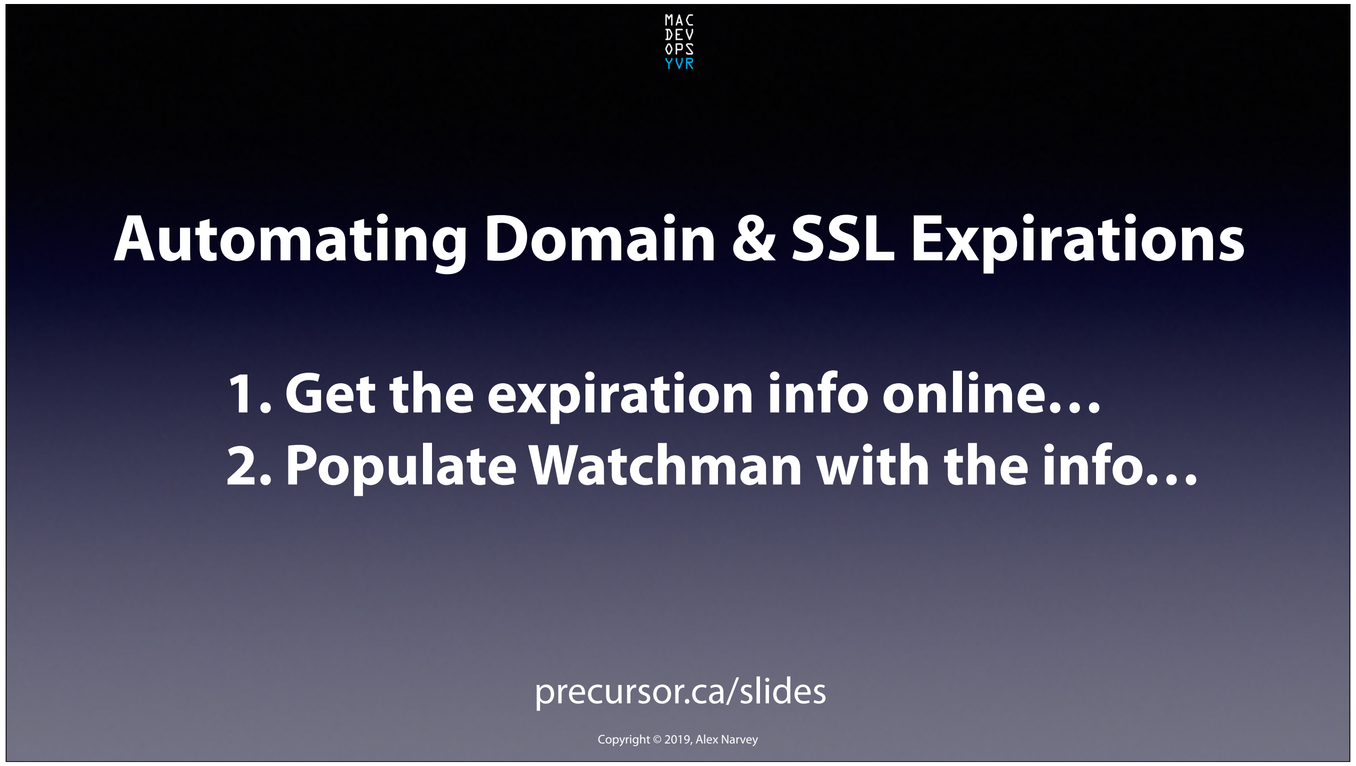
precursor.ca/slides

And that’s a pain!
CLICK

Use Watchman’s REST API & Automate!

precursor.ca/slides

So I decided to try and use some of the programming methods I have been exposed to and automate this process.
CLICK

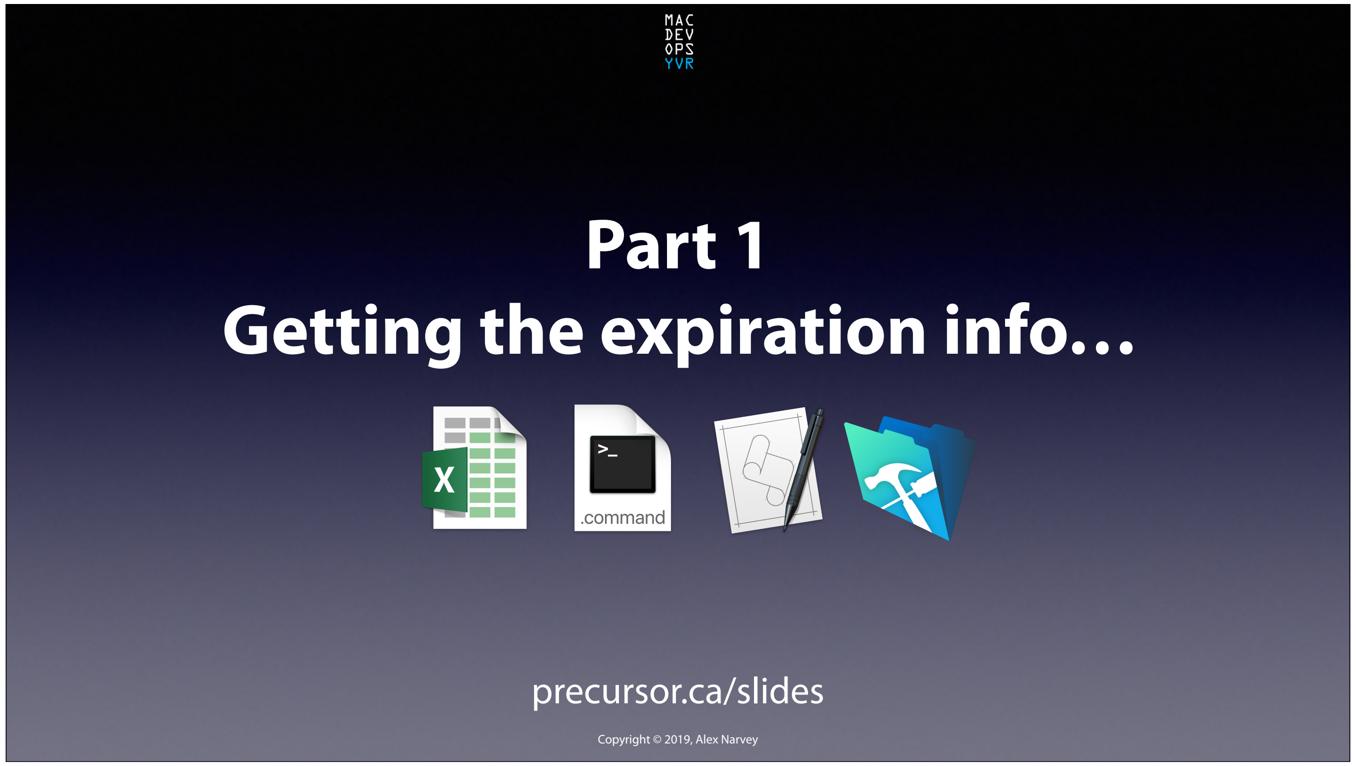


Automating Domain & SSL Expirations

1. Get the expiration info online…
2. Populate Watchman with the info…

precursor.ca/slides

Step one would be to acquire the Expiration info online
and
Step two would be to populate that info into Watchman
All programmatically.
CLICK

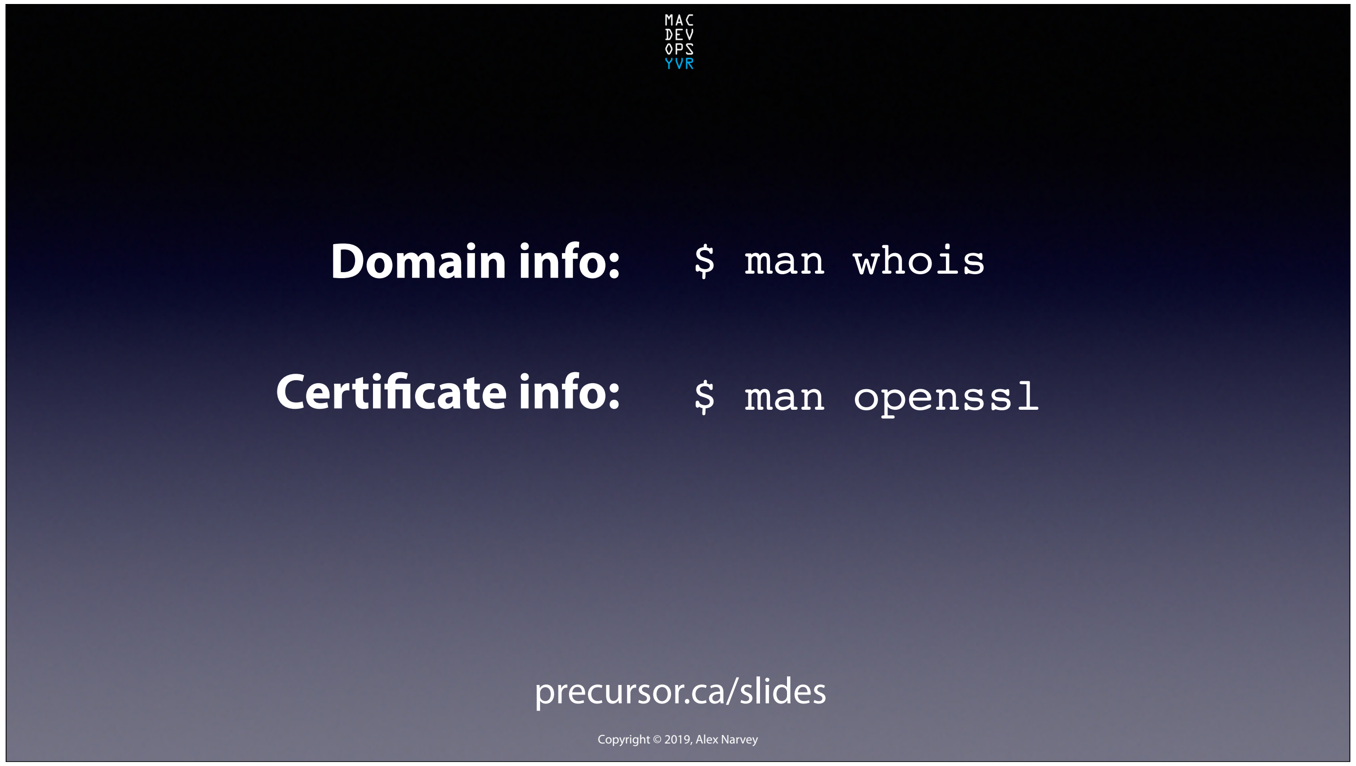


MAC DEV OPS YVR

Part 1
Getting the expiration info…

precursor.ca/slides

So lets start with Part 1: Getting the expiration info…
CLICK



MAC
DEV
OPS
YVR

Domain info:

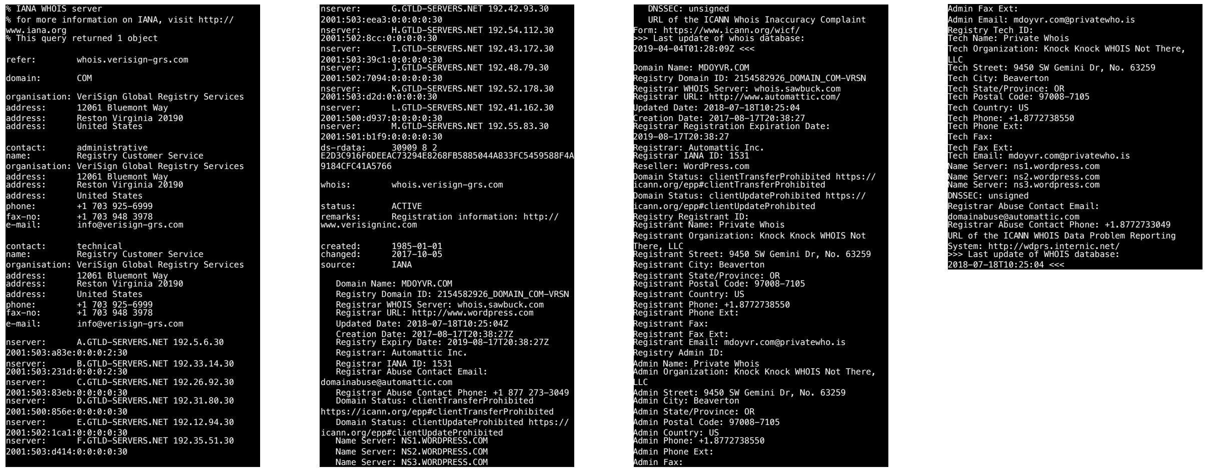
```
$ man whois
```

Certificate info:

```
$ man openssl
```

precursor.ca/slides

Fortunately, there are command line tools available to do this.
CLICK



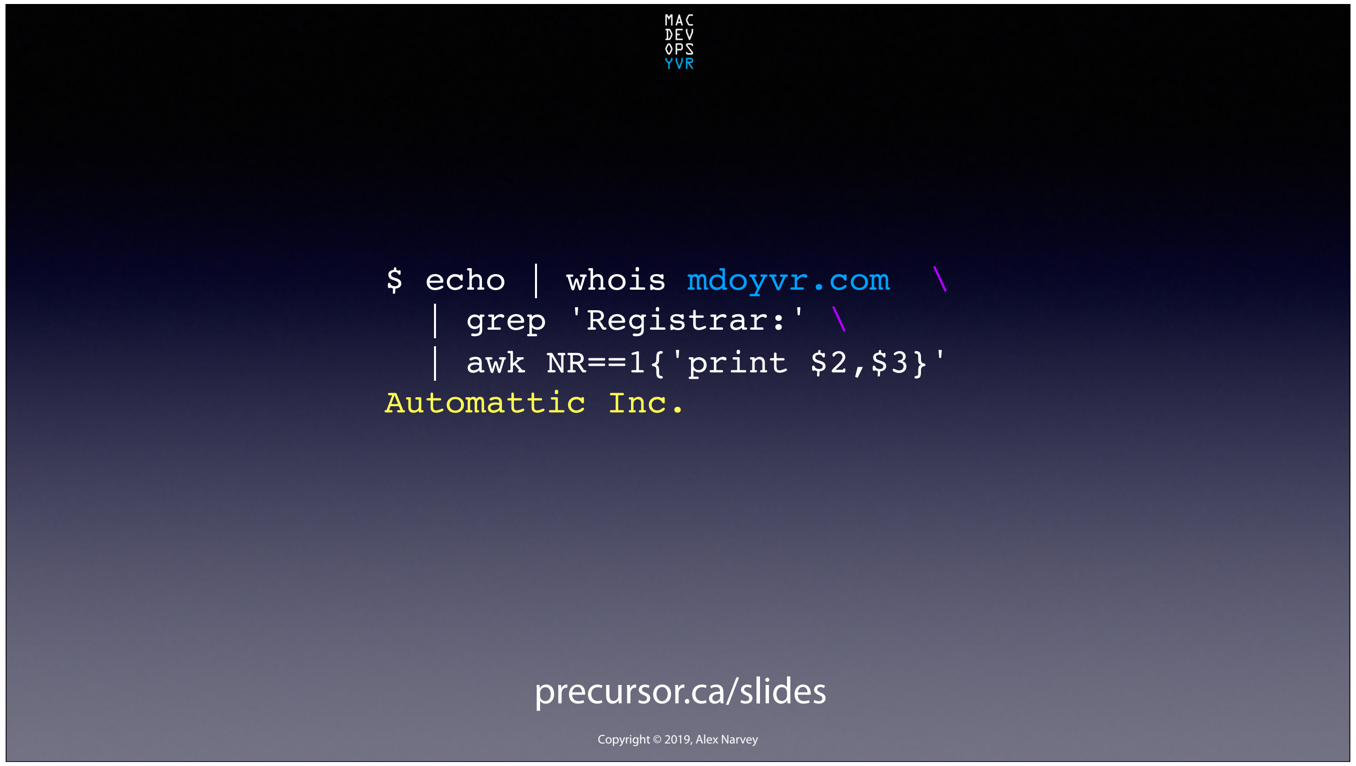
```
% IANA WHOIS server
% for more information on IANA, visit http://
www.iana.org
% This query returned 1 object

refer:        whois.verisign-grs.com

domain:       COM

organisation: VeriSign Global Registry Services
address:      12061 Bluemont Way
address:      Reston Virginia 20190
address:      United States

contact:      administrative
name:         Registry Customer Service
organisation: VeriSign Global Registry Services
address:      12061 Bluemont Way
address:      Reston Virginia 20190
address:      United States
phone:        +1 703 925-6999
fax-no:       +1 703 948 3978
e-mail:       info@verisign-grs.com

contact:      technical
name:         Registry Customer Service
organisation: VeriSign Global Registry Services
address:      12061 Bluemont Way
address:      Reston Virginia 20190
address:      United States
phone:        +1 703 925-6999
fax-no:       +1 703 948 3978
e-mail:       info@verisign-grs.com

nserver:      A.GTLD-SERVERS.NET 192.5.6.30
2001:503:a83e:0:0:0:2:30
nserver:      B.GTLD-SERVERS.NET 192.33.14.30
2001:503:231d:0:0:0:2:30
nserver:      C.GTLD-SERVERS.NET 192.26.92.30
2001:503:83eb:0:0:0:0:30
nserver:      D.GTLD-SERVERS.NET 192.31.80.30
2001:500:856e:0:0:0:0:30
nserver:      E.GTLD-SERVERS.NET 192.12.94.30
2001:502:1ca1:0:0:0:0:30
nserver:      F.GTLD-SERVERS.NET 192.35.51.30
2001:503:d414:0:0:0:0:30
```

```
nserver:      G.GTLD-SERVERS.NET 192.42.93.30
2001:503:eea3:0:0:0:0:30
nserver:      H.GTLD-SERVERS.NET 192.54.112.30
2001:502:8cc:0:0:0:0:30
nserver:      I.GTLD-SERVERS.NET 192.43.172.30
2001:503:39c1:0:0:0:0:30
nserver:      J.GTLD-SERVERS.NET 192.48.79.30
2001:502:7094:0:0:0:0:30
nserver:      K.GTLD-SERVERS.NET 192.52.178.30
2001:503:d2d:0:0:0:0:30
nserver:      L.GTLD-SERVERS.NET 192.41.162.30
2001:500:d937:0:0:0:0:30
nserver:      M.GTLD-SERVERS.NET 192.55.83.30
2001:501:b1f9:0:0:0:0:30
ds-rdata:     30909 8 2
E2D3C916F6DEEAC73294E8268FB5885044A833FC5459588F4A
9184CFC41A5766

whois:        whois.verisign-grs.com

status:       ACTIVE
remarks:      Registration information: http://
www.verisigninc.com

created:      1985-01-01
changed:      2017-10-05
source:       IANA

   Domain Name: MDOYVR.COM
   Registry Domain ID: 2154582926_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.sawbuck.com
   Registrar URL: http://www.wordpress.com
   Updated Date: 2018-07-18T10:25:04Z
   Creation Date: 2017-08-17T20:38:27Z
   Registry Expiry Date: 2019-08-17T20:38:27Z
   Registrar: Automattic Inc.
   Registrar IANA ID: 1531
   Registrar Abuse Contact Email:
domainabuse@automattic.com
   Registrar Abuse Contact Phone: +1 877 273-3049
   Domain Status: clientTransferProhibited
https://icann.org/epp#clientTransferProhibited
   Domain Status: clientUpdateProhibited https://
icann.org/epp#clientUpdateProhibited
   Name Server: NS1.WORDPRESS.COM
   Name Server: NS2.WORDPRESS.COM
   Name Server: NS3.WORDPRESS.COM
```

```
   DNSSEC: unsigned
   URL of the ICANN Whois Inaccuracy Complaint
Form: https://www.icann.org/wicf/
>>> Last update of whois database:
2019-04-04T01:28:09Z <<<

Domain Name: MDOYVR.COM
Registry Domain ID: 2154582926_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.sawbuck.com
Registrar URL: http://www.automattic.com/
Updated Date: 2018-07-18T10:25:04
Creation Date: 2017-08-17T20:38:27
Registrar Registration Expiration Date:
2019-08-17T20:38:27
Registrar: Automattic Inc.
Registrar IANA ID: 1531
Reseller: WordPress.com
Domain Status: clientTransferProhibited https://
icann.org/epp#clientTransferProhibited
Domain Status: clientUpdateProhibited https://
icann.org/epp#clientUpdateProhibited
Registry Registrant ID:
Registrant Name: Private Whois
Registrant Organization: Knock Knock WHOIS Not
There, LLC
Registrant Street: 9450 SW Gemini Dr, No. 63259
Registrant City: Beaverton
Registrant State/Province: OR
Registrant Postal Code: 97008-7105
Registrant Country: US
Registrant Phone: +1.8772738550
Registrant Phone Ext:
Registrant Fax:
Registrant Fax Ext:
Registrant Email: mdoyvr.com@privatewho.is
Registry Admin ID:
Admin Name: Private Whois
Admin Organization: Knock Knock WHOIS Not There,
LLC
Admin Street: 9450 SW Gemini Dr, No. 63259
Admin City: Beaverton
Admin State/Province: OR
Admin Postal Code: 97008-7105
Admin Country: US
Admin Phone: +1.8772738550
Admin Phone Ext:
Admin Fax:
```

```
Admin Fax Ext:
Admin Email: mdoyvr.com@privatewho.is
Registry Tech ID:
Tech Name: Private Whois
Tech Organization: Knock Knock WHOIS Not There,
LLC
Tech Street: 9450 SW Gemini Dr, No. 63259
Tech City: Beaverton
Tech State/Province: OR
Tech Postal Code: 97008-7105
Tech Country: US
Tech Phone: +1.8772738550
Tech Phone Ext:
Tech Fax:
Tech Fax Ext:
Tech Email: mdoyvr.com@privatewho.is
Name Server: ns1.wordpress.com
Name Server: ns2.wordpress.com
Name Server: ns3.wordpress.com
DNSSEC: unsigned
Registrar Abuse Contact Email:
domainabuse@automattic.com
Registrar Abuse Contact Phone: +1.8772733049
URL of the ICANN WHOIS Data Problem Reporting
System: http://wdprs.internic.net/
>>> Last update of WHOIS database:
2018-07-18T10:25:04 <<<
```

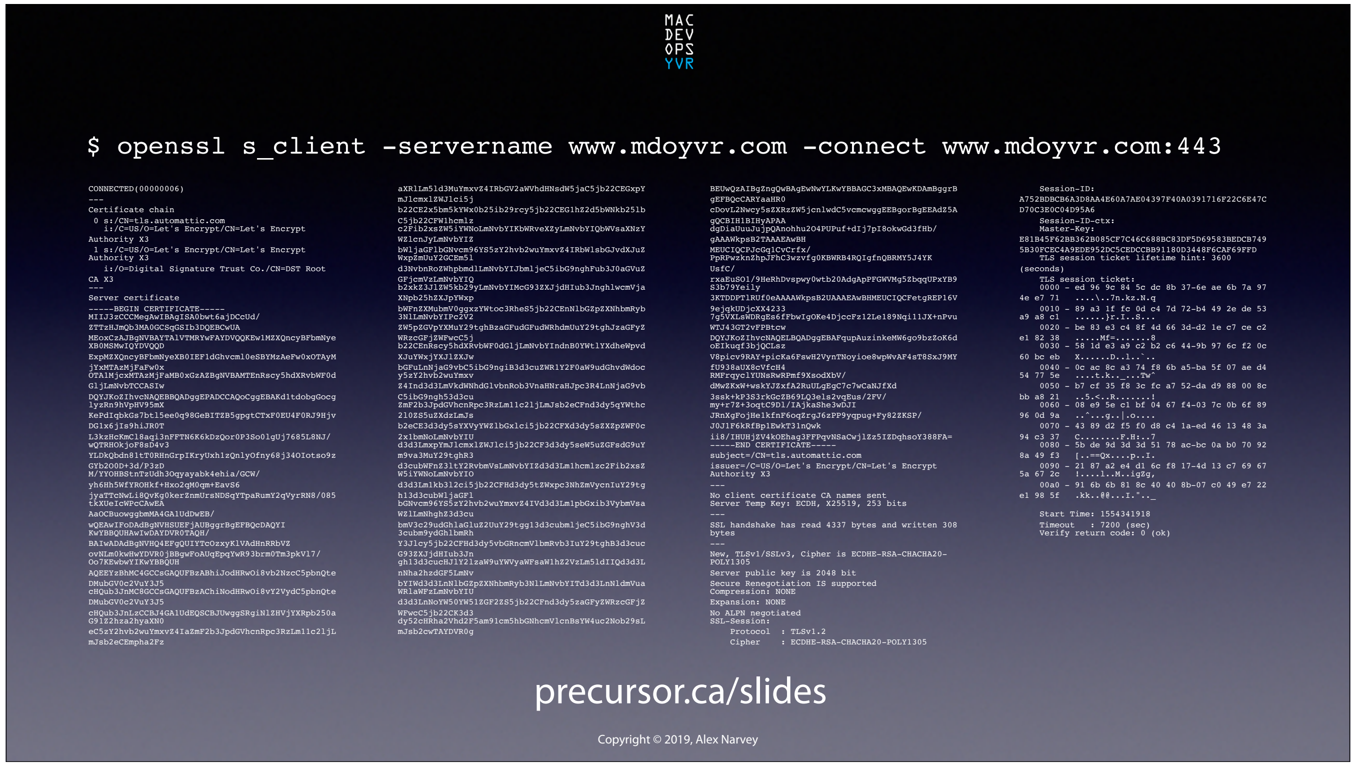
For example, if we do a **WHOIS** command on the domain mdoyvr.com we get all this!
CLICK



```
$ echo | whois mdoyvr.com  \
  | grep 'Registrar:' \
  | awk NR==1{'print $2,$3}'
Automattic Inc.
```

precursor.ca/slides

And if we pipe that output to various grep and awk routines we can suss out things like Registar's name and the expiry date.
CLICK



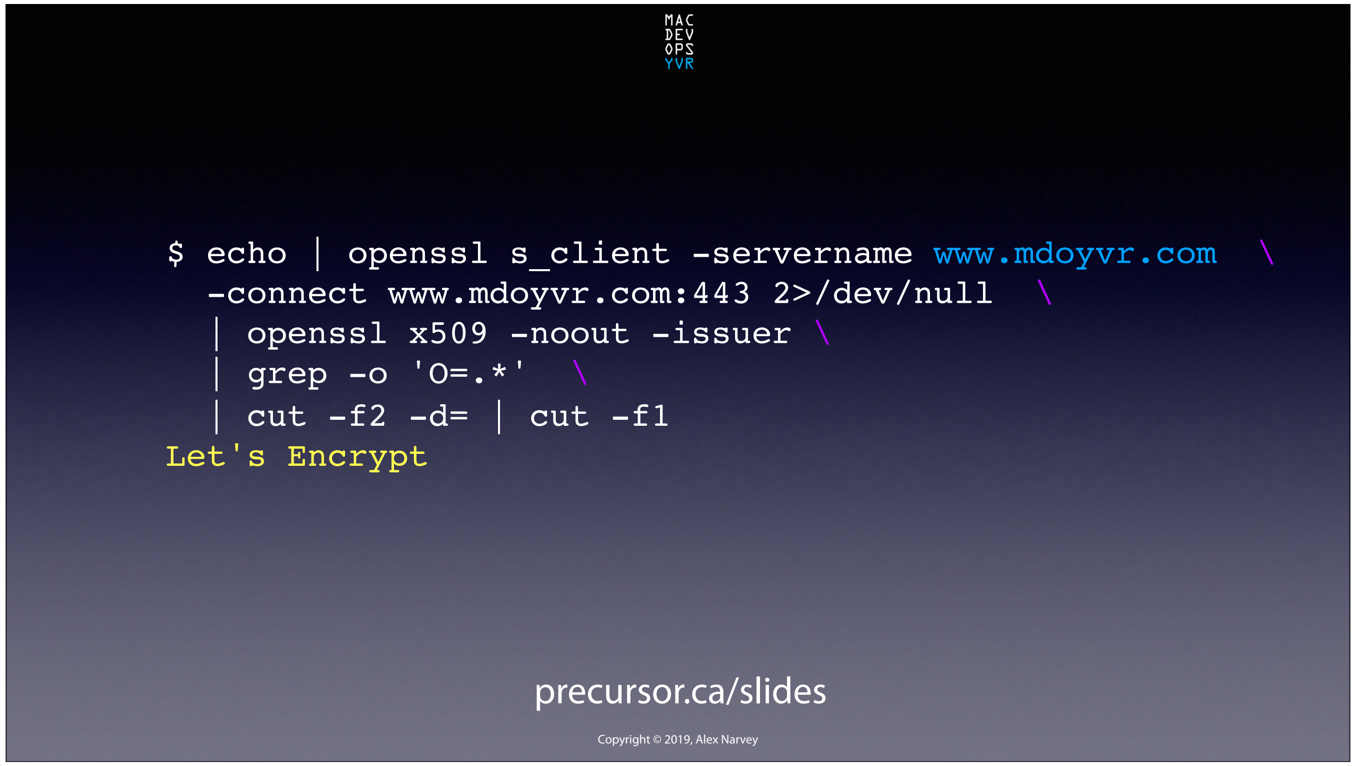
```
$ openssl s_client -servername www.mdoyvr.com -connect www.mdoyvr.com:443
```

```
CONNECTED(00000006)
---
Certificate chain
 0 s:/CN=tls.automattic.com
   i:/C=US/O=Let's Encrypt/CN=Let's Encrypt
Authority X3
 1 s:/C=US/O=Let's Encrypt/CN=Let's Encrypt
Authority X3
   i:/O=Digital Signature Trust Co./CN=DST Root
CA X3
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIJ3zCCCMegAwIBAgISA0bwt6ajDCcUd/
ZTTzHJmQb3MA0GCSqGSIb3DQEBCwUA
MEoxCzAJBgNVBAYTAlVTMRYwFAYDVQQKEw1MZXQncyBFbmNye
XB0MSMwIQYDVQQD
ExpMZXQncyBFbmNyeXB0IEF1dGhvcml0eSBYMzAeFw0xOTAyM
jYxMTAzMjFaFw0x
OTA1MjcxMTAzMjFaMB0xGzAZBgNVBAMTEnRscy5hdXRvbWF0d
GljLmNvbTCCASIw
DQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAKd1tdobgGocg
lyzRn9hVpHV95mX
KePdIqbkGs7btl5ee0q98GeBITZB5gpgtCTxF0EU4F0RJ9Hjv
DG1x6jIs9hiJR0T
L3kzHcKmCl8aqi3nFFTN6K6kDzQor0P3So0lgUj7685L8NJ/
wQTRHOkjoF8sD4v3
YLDkQbdn81tT0RHnGrpIKryUxh1zQnlyOfny68j34OIotso9z
GYb2O0D+3d/P3zD
M/YYOHBStnTzUdh3Oqyayabk4ehia/GCW/
yh6Hh5WfYROHkf+Hxo2qM0qm+EavS6
jyaTTcNwLi8QvKg0kerZnmUrsNDSqYTpaRumY2qVyrRN8/085
tkXUeIcWPcCAwEA
AaOCBuowggbmMA4GA1UdDwEB/
wQEAwIFoDAdBgNVHSUEFjAUBggrBgEFBQcDAQYI
KwYBBQUHAwIwDAYDVR0TAQH/
BAIwADAdBgNVHQ4EFgQUIYTcOzxyKlVAdHnRRbVZ
ovNLm0kwHwYDVR0jBBgwFoAUqEpqYwR93brm0Tm3pkVl7/
Oo7KEwbwYIKwYBBQUH
AQEEYzBhMC4GCCsGAQUFBzABhiJodHRwOi8vb2NzcC5pbnQte
DMubGV0c2VuY3J5
cHQub3JnMC8GCCsGAQUFBzAChiNodHRwOi8vY2VydC5pbnQte
DMubGV0c2VuY3J5
cHQub3JnLzCCBJ4GA1UdEQSCBJUwggSRgiNlZHVjYXRpb250a
G91Z2hza2hyaXN0
eC5zY2hvb2wuYmxvZ4IaZmF2b3JpdGVhcnRpc3RzLm11c2ljL
mJsb2eCEmpha2Fz
aXRlLm5ld3MuYmxvZ4IRbGV2aWVhdHNsdW5jaC5jb22CEGxpY
mJlcmxlZWJlci5j
b22CE2x5bm5kYWx0b25ib29rcy5jb22CEG1hZ2d5bWNkb25lb
C5jb22CFW1hcmlz
c2Fib2xsZW5iYWNoLmNvbYIKbWRveXZyLmNvbYIQbWVsaXNzY
WZlcnJyLmNvbYIZ
bWljaGFlbGNvcm96YS5zY2hvb2wuYmxvZ4IRbWlsbGJvdXJuZ
WxpZmUuY2GCEm5l
d3NvbnRoZWhpbmdlLmNvbYIJbmljeC5ibG9nghFub3J0aGVuZ
GFjcmVzLmNvbYIQ
b2xkZ3JlZW5kb29yLmNvbYIMcG93ZXJjdHIub3JnghlwcmVja
XNpb25hZXJpYWxp
bWFnZXMubmV0ggxzYWtoc3RheS5jb22CEnNlbGZpZXNhbmRyb
3NlLmNvbYIPc2V2
ZW5pZGVpYXMuY29tghBzaGFudGFudWRhdmUuY29tghJzaGFyZ
WRzcGFjZWFwcC5j
b22CEnRscy5hdXRvbWF0dGljLmNvbYIndnB0YWtlYXdheWpvd
XJuYWxjYXJlZXJw
bGFuLnNjaG9vbC5ibG9ngiB3d3cuZWR1Y2F0aW9udGhvdWdoc
y5zY2hvb2wuYmxv
Z4Ind3d3LmVkdWNhdGlvbnRob3VnaHNraHJpc3R4LnNjaG9vb
C5ibG9ngh53d3cu
ZmF2b3JpdGVhcnRpc3RzLm11c2ljLmJsb2eCFnd3dy5qYWthc
2l0ZS5uZXdzLmJs
b2eCE3d3dy5sYXVyYWZlbGxlci5jb22CFXd3dy5sZXZpZWF0c
2x1bmNoLmNvbYIU
d3d3LmxpYmJlcmxlZWJlci5jb22CF3d3dy5seW5uZGFsdG9uY
m9va3MuY29tghR3
d3cubWFnZ3ltY2RvbmVsLmNvbYIZd3d3Lm1hcmlzc2Fib2xsZ
W5iYWNoLmNvbYIO
d3d3Lm1kb3l2ci5jb22CFHd3dy5tZWxpc3NhZmVycnIuY29tg
h13d3cubWljaGFl
bGNvcm96YS5zY2hvb2wuYmxvZ4IVd3d3Lm1pbGxib3VybmVsa
WZlLmNhghZ3d3cu
bmV3c29udGhlaGluZ2UuY29tgg13d3cubmljeC5ibG9nghV3d
3cubm9ydGhlbmRh
Y3Jlcy5jb22CFHd3dy5vbGRncmVlbmRvb3IuY29tghB3d3cuc
G93ZXJjdHIub3Jn
gh13d3cucHJlY2lzaW9uYWVyaWFsaW1hZ2VzLm5ldIIQd3d3L
nNha2hzdGF5LmNv
bYIWd3d3LnNlbGZpZXNhbmRyb3NlLmNvbYITd3d3LnNldmVua
WRlaWFzLmNvbYIU
d3d3LnNoYW50YW51ZGF2ZS5jb22CFnd3dy5zaGFyZWRzcGFjZ
WFwcC5jb22CK3d3
dy52cHRha2Vhd2F5am91cm5hbGNhcmVlcnBsYW4uc2Nob29sL
mJsb2cwTAYDVR0g
BEUwQzAIBgZngQwBAgEwNwYLKwYBBAGC3xMBAQEwKDAmBggrB
gEFBQcCARYaaHR0
cDovL2Nwcy5sZXRzZW5jcnlwdC5vcmcwggEEBgorBgEEAdZ5A
gQCBIH1BIHyAPAA
dgDiaUuuJujpQAnohhu2O4PUPuf+dIj7pI8okwGd3fHb/
gAAAWkpsB2TAAAEAwBH
MEUCIQCPJcGq1CvCrfx/
PpRPwzknZhpJFhC3wzvfg0KBWRB4RQIgfnQBRMY5J4YK
UsfC/
rxaEuSO1/9HeRhDvspwy0wtb20AdgApPFGWVMg5ZbqqUPxYB9
S3b79Yeily
3KTDDPTlRUf0eAAAAWkpsB2UAAAEAwBHMEUCIQCFetgREP16V
9ejqkUDjcXX4233
7g5VXLsWDRgEs6fFbwIgOKe4DjccFz12Le189Nqi11JX+nPvu
WTJ43GT2vFPBtcw
DQYJKoZIhvcNAQELBQADggEBAFqupAuzinkeMW6go9bzZoK6d
oEIkuqf3bjQCLsz
V8picv9RAY+picKa6FswH2VynTNoyioe8wpWvAF4sT8SxJ9MY
fU938aUX8cVfcH4
RMFrqyclYUNsRwRPmf9XsodXbV/
dMwZKxW+wskYJZxfA2RuULgEgC7c7wCaNJfXd
3ssk+kP3S3rkGcZB69LQ3els2vqEus/2FV/
my+r7Z+3oqtC9Dl/IAjkaShe3wDJI
JRnXgFojHe1kfnF6oqZrgJ6zPP9yqpug+Fy82ZKSP/
J0J1F6kRfBp1EwkT31nQwk
ii8/IHUHjZV4kOEhag3FFPqvNSaCwjlZz5IZDqhsoY388FA=
-----END CERTIFICATE-----
subject=/CN=tls.automattic.com
issuer=/C=US/O=Let's Encrypt/CN=Let's Encrypt
Authority X3
---
No client certificate CA names sent
Server Temp Key: ECDH, X25519, 253 bits
---
SSL handshake has read 4337 bytes and written 308
bytes
---
New, TLSv1/SSLv3, Cipher is ECDHE-RSA-CHACHA20-
POLY1305
Server public key is 2048 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
SSL-Session:
    Protocol  : TLSv1.2
    Cipher    : ECDHE-RSA-CHACHA20-POLY1305
    Session-ID:
A752BDBCB6A3D8AA4E60A7AE04397F40A0391716F22C6E47C
D70C3E0C04D95A6
    Session-ID-ctx:
    Master-Key:
E81B45F62BB362B085CF7C46C688BC83DF5D69583BEDCB749
5B30FCEC4A9EDE952DC5CEDCCBB91180D3448F6CAF69FFD
    TLS session ticket lifetime hint: 3600
(seconds)
    TLS session ticket:
    0000 - ed 96 9c 84 5c dc 8b 37-6e ae 6b 7a 97
4e e7 71   ....\..7n.kz.N.q
    0010 - 89 a3 1f fc 0d c4 7d 72-b4 49 2e de 53
a9 a8 c1   ......}r.I..S...
    0020 - be 83 e3 c4 8f 4d 66 3d-d2 1e c7 ce c2
e1 82 38   .....Mf=.......8
    0030 - 58 1d e3 a9 c2 b2 c6 44-9b 97 6c f2 0c
60 bc eb   X......D..l..`..
    0040 - 0c ac 8c a3 74 f8 6b a5-ba 5f 07 ae d4
54 77 5e   ....t.k.._...Tw^
    0050 - b7 cf 35 f8 3c fc a7 52-da d9 88 00 8c
bb a8 21   ..5.<..R.......!
    0060 - 08 e9 5e c1 bf 04 67 f4-03 7c 0b 6f 89
96 0d 9a   ..^...g..|.o....
    0070 - 43 89 d2 f5 f0 d8 c4 1a-ed 46 13 48 3a
94 c3 37   C........F.H:..7
    0080 - 5b de 9d 3d 3d 51 78 ac-bc 0a b0 70 92
8a 49 f3   [..==Qx....p..I.
    0090 - 21 87 a2 e4 d1 6c f8 17-4d 13 c7 69 67
5a 67 2c   !....l..M..igZg,
    00a0 - 91 6b 6b 81 8c 40 40 8b-07 c0 49 e7 22
e1 98 5f   .kk..@@...I.".._

    Start Time: 1554341918
    Timeout   : 7200 (sec)
    Verify return code: 0 (ok)
```

precursor.ca/slides

Here’s what we get if we do an OPENSSL on the certificate for www.mdoyvr.com!

CLICK

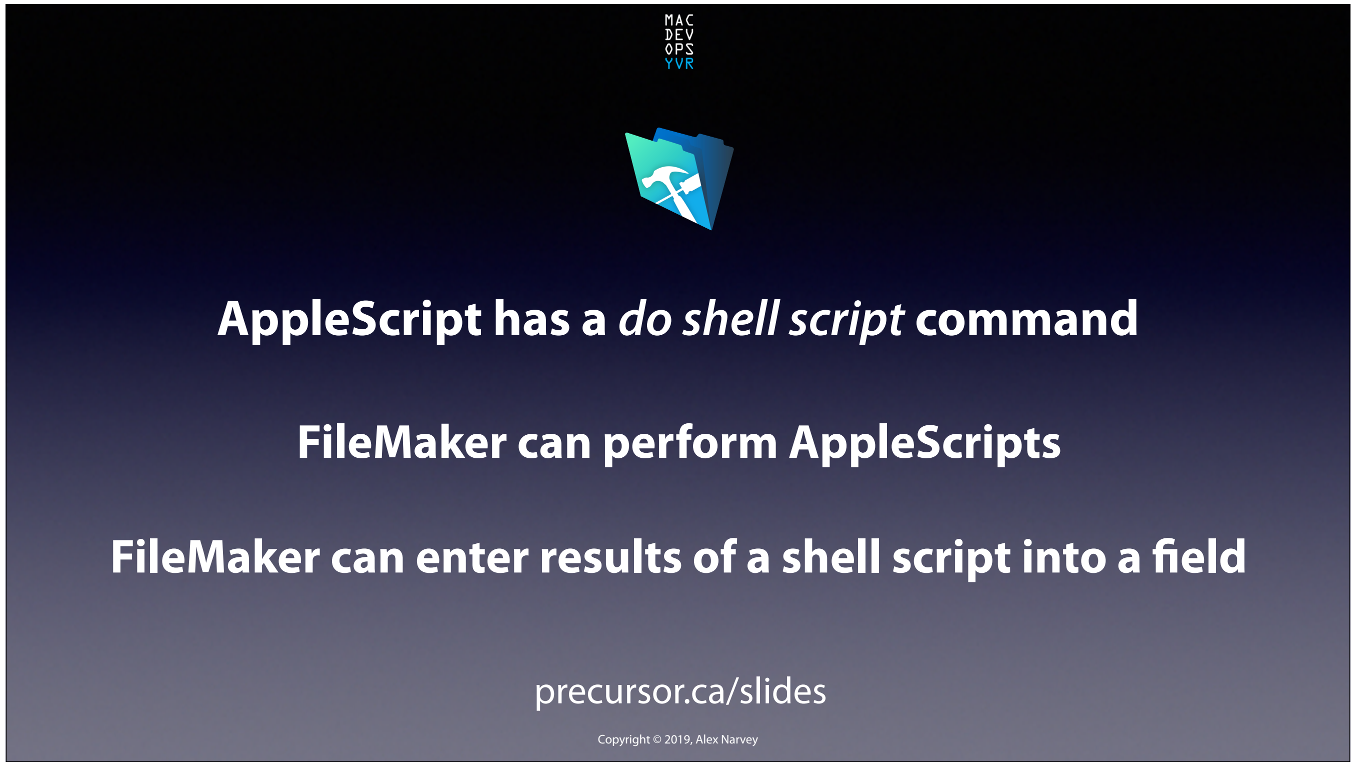


MAC
DEV
OPS
YVR

```
$ echo | openssl s_client -servername www.mdoyvr.com  \
  -connect www.mdoyvr.com:443 2>/dev/null  \
  | openssl x509 -noout -issuer \
  | grep -o 'O=.*'  \
  | cut -f2 -d= | cut -f1
Let's Encrypt
```

precursor.ca/slides

And again if we pipe that output through various grep and cut commands we can suss out the name of the Certificate’s issuer and the expiry date.
CLICK



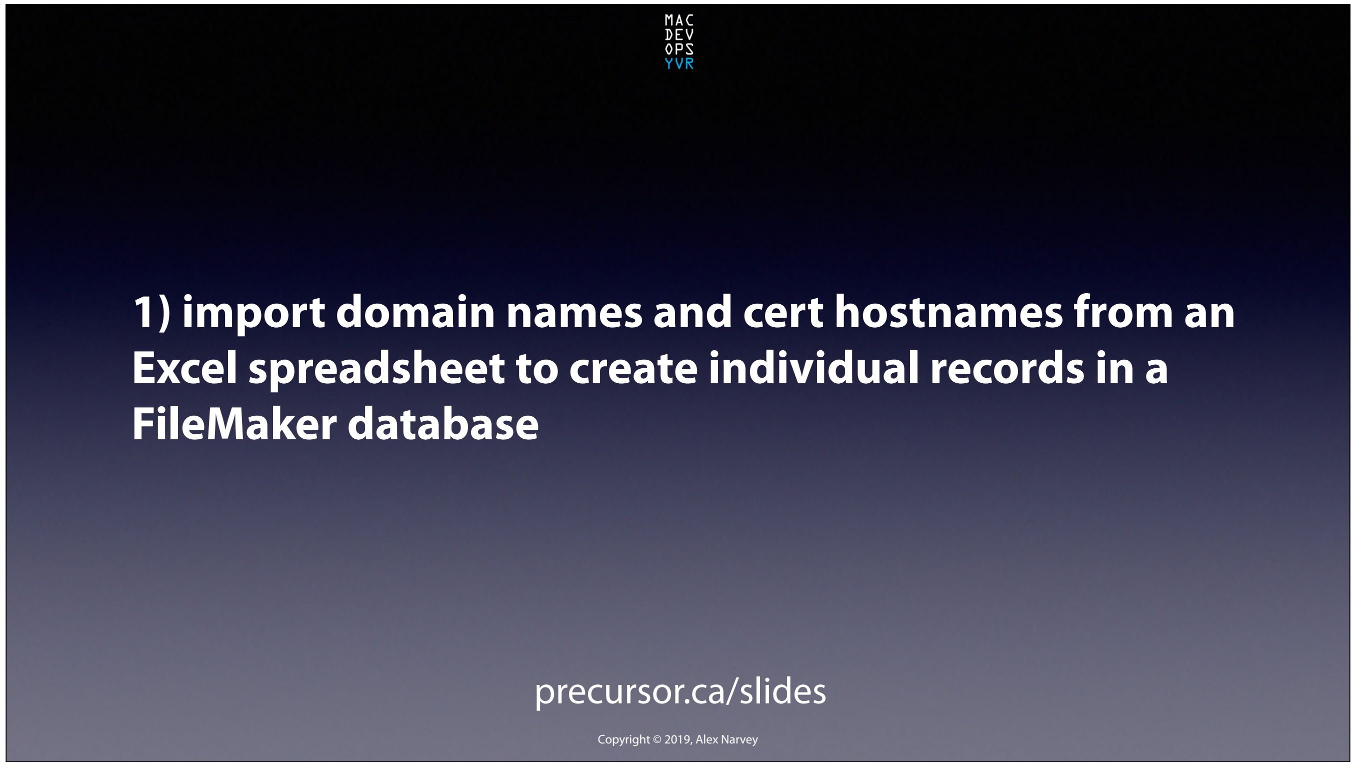
AppleScript has a *do shell script* command

FileMaker can perform AppleScripts

FileMaker can enter results of a shell script into a field

precursor.ca/slides

Now to automate these shell scripts I recalled that AppleScript has a “do shell script” command, and that FileMaker can perform AppleScripts and enter the results into a database field.
CLICK



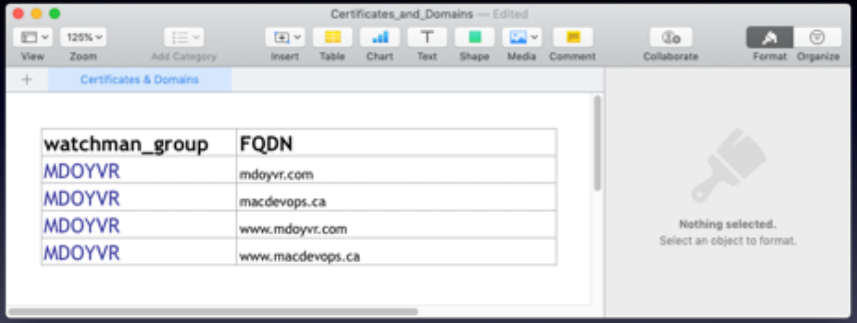
MAC
DEV
OPS
YVR

1) import domain names and cert hostnames from an Excel spreadsheet to create individual records in a FileMaker database

precursor.ca/slides

So all we need to do is import a list of domain names and certificate host names in to FileMaker and have it run these scripts for us.
CLICK

MAC
DEV
OPS
YVR



watchman_group	FQDN
MDOYVR	mdoyvr.com
MDOYVR	macdevops.ca
MDOYVR	www.mdoyvr.com
MDOYVR	www.macdevops.ca

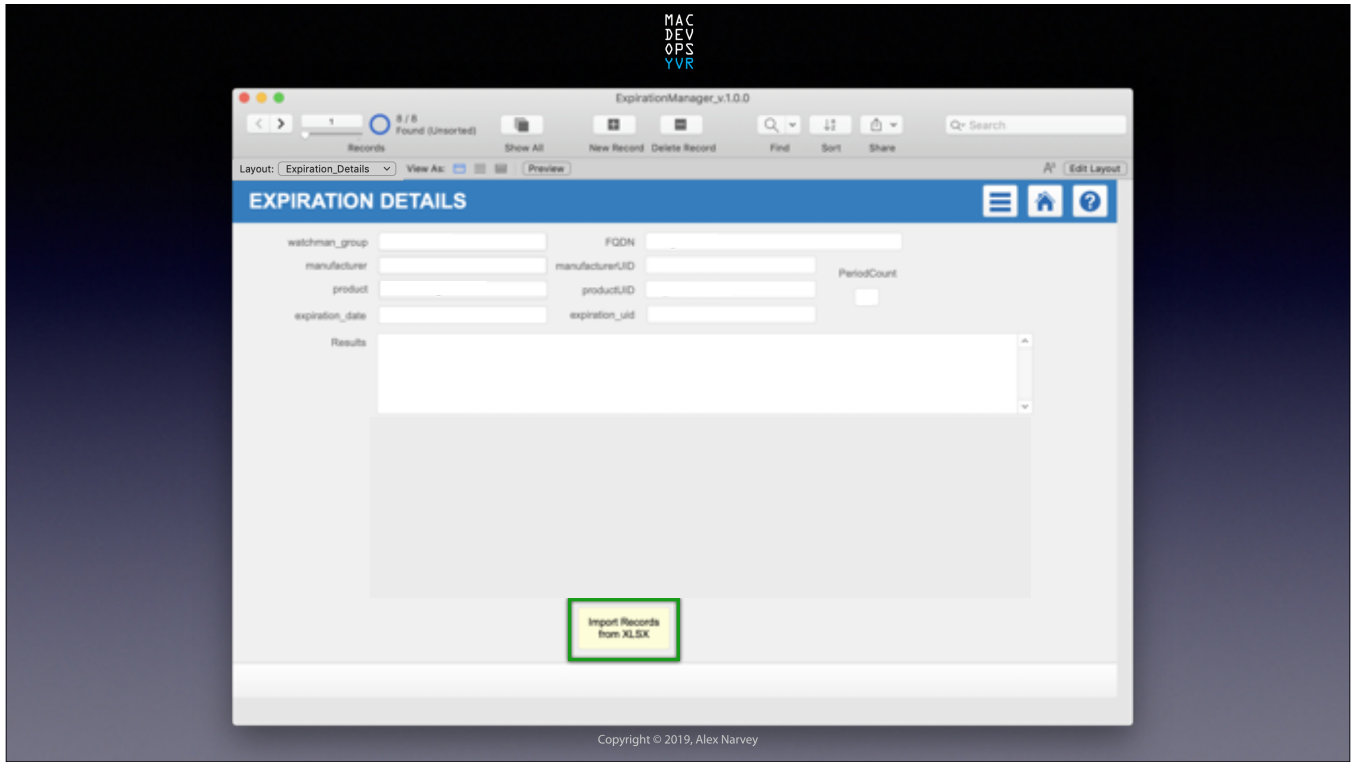
Nothing selected.
Select an object to format.

precursor.ca/slides

Here’s an example spreadsheet. For the purpose of this exercise I am unofficially the MSP for MacDevOps: YVR and I have created a Group in my Watchman Dashboard called “MDOYVR”.

I am going to track Mat X’s new domain *mdoyvr.com* and the old one too: *macdevops.ca* and the security certifcates for both.

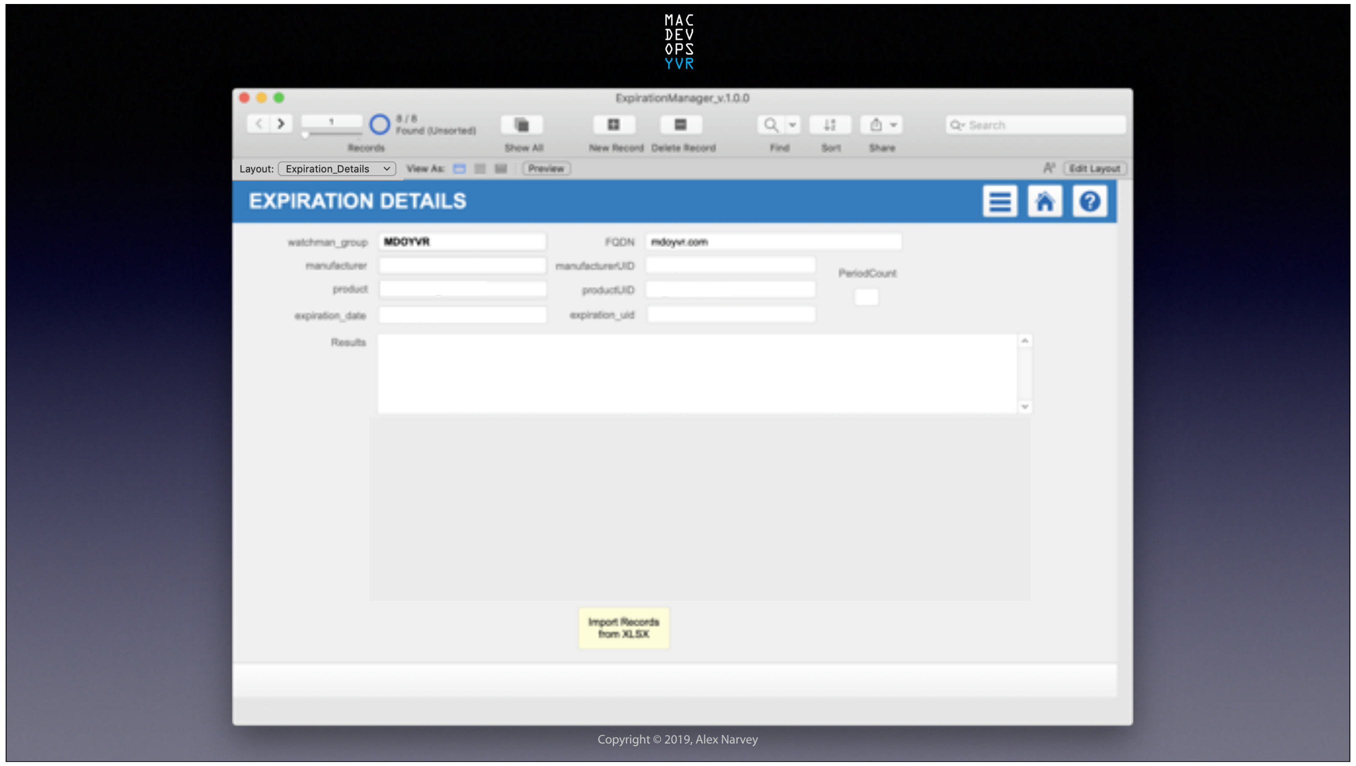
CLICK



So here’s a look at the FileMaker Expiration Details layout.

We click the button to run the import script and…

CLICK

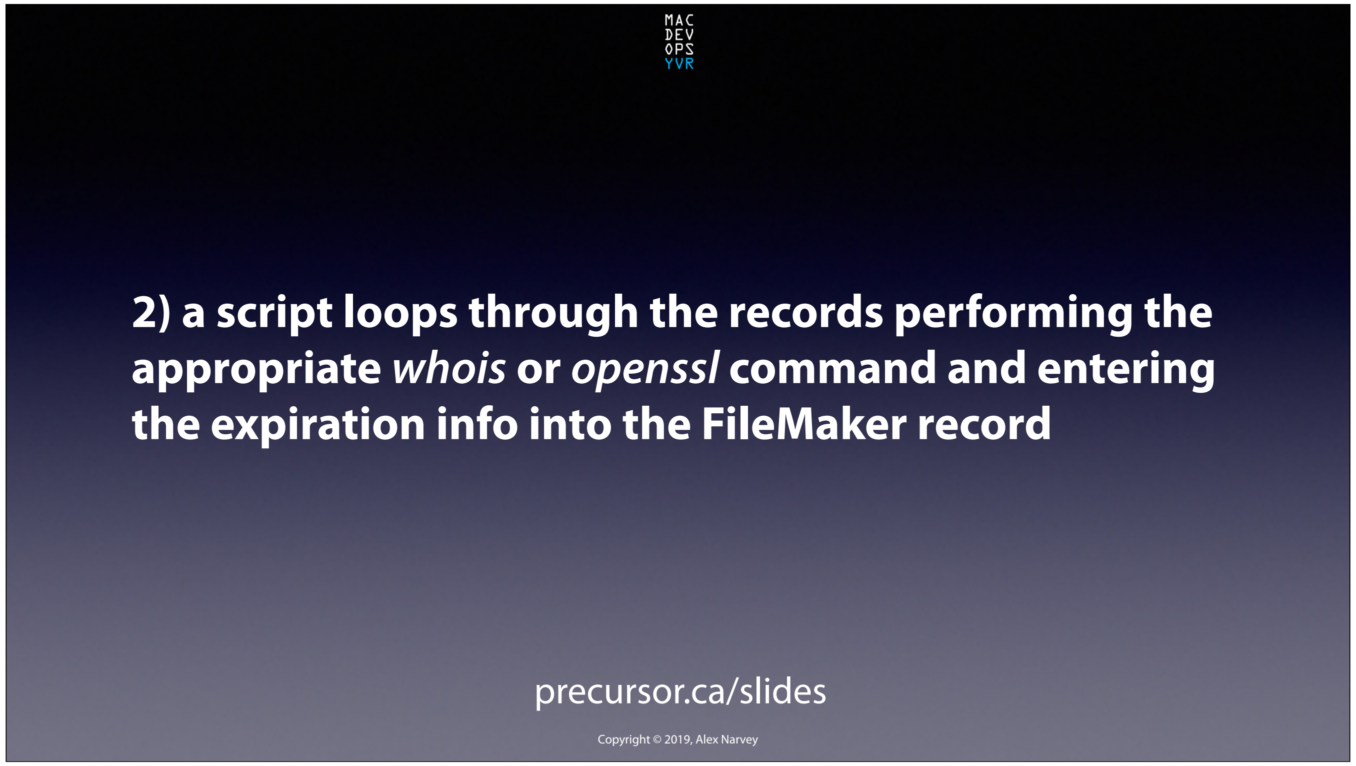


EXPIRATION DETAILS

watchman_group MDOYVR
FQDN mdoyvr.com
manufacturer
manufacturerUID
PeriodCount
product
productUID
expiration_date
expiration_uid
Results

Copyright © 2019, Alex Narvey

FileMaker now has some records with the Watchman Group name and FQDN we want expiration info for.
CLICK

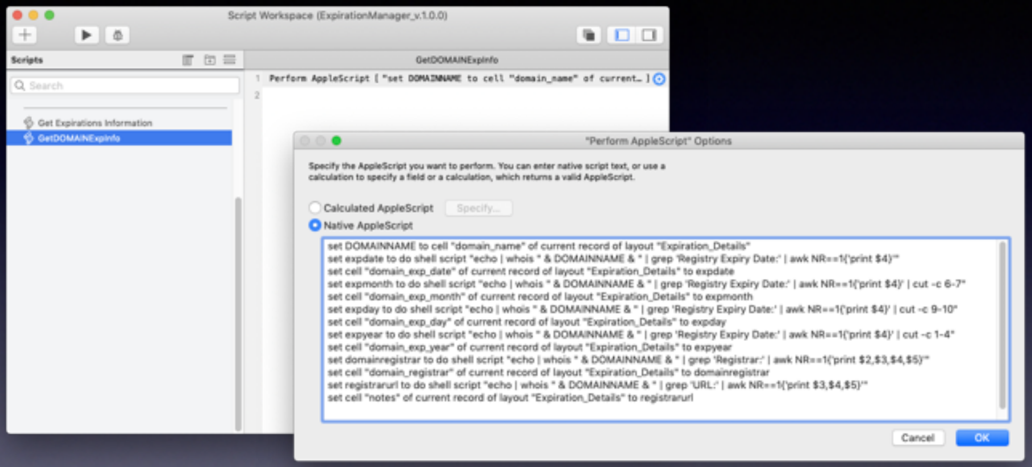


2) a script loops through the records performing the appropriate *whois* or *openssl* command and entering the expiration info into the FileMaker record

precursor.ca/slides

We get FileMaker to loop through these records, perform the scripts and put the info it collects into the appropriate fields of our records.
CLICK

MAC
DEV
OPS
YVR


Script Workspace (ExpirationManager_v.1.0.0)
Scripts
Search
Get Expirations Information
GetDOMAINExpInfo

```
Perform AppleScript [ "set DOMAINNAME to cell "domain_name" of current… ]
```

"Perform AppleScript" Options

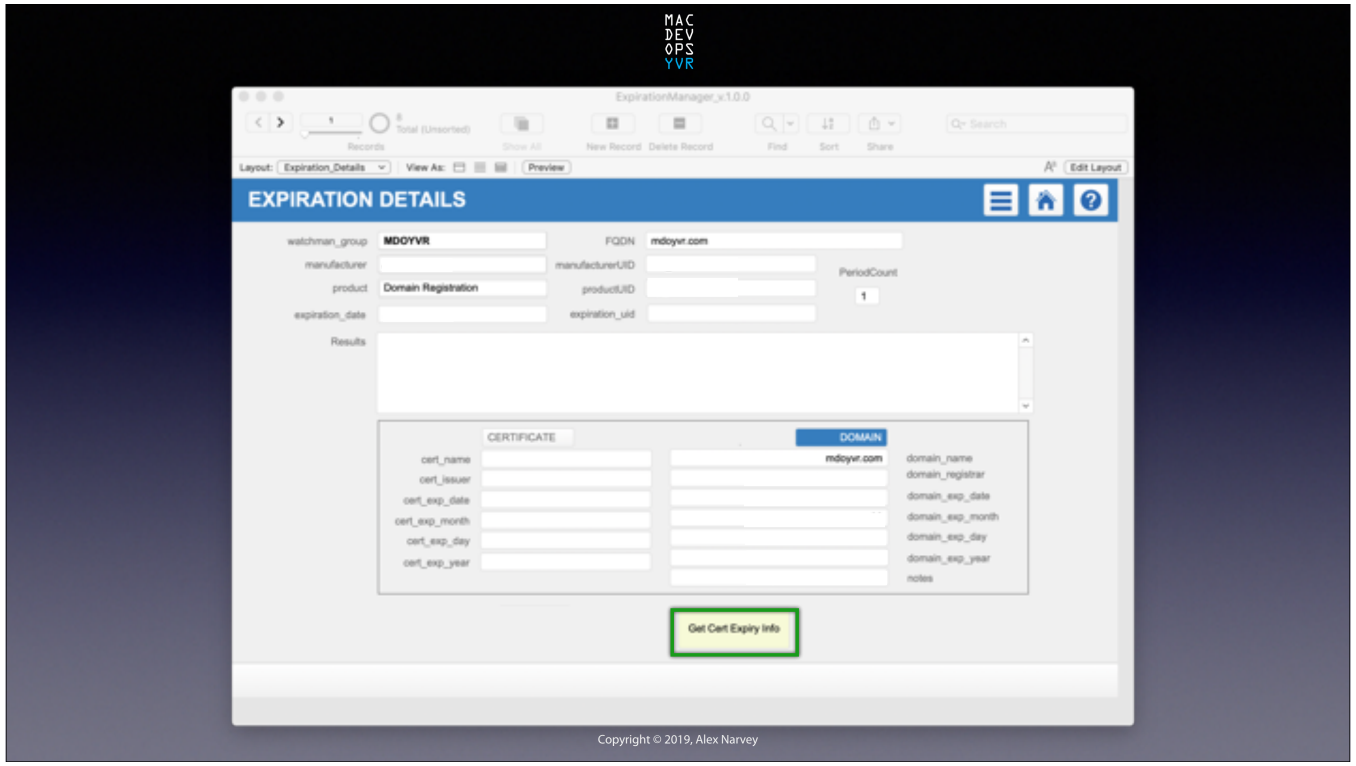
Specify the AppleScript you want to perform. You can enter native script text, or use a calculation to specify a field or a calculation, which returns a valid AppleScript.

○ Calculated AppleScript   Specify…
● Native AppleScript

```
set DOMAINNAME to cell "domain_name" of current record of layout "Expiration_Details"
set expdate to do shell script "echo | whois " & DOMAINNAME & " | grep 'Registry Expiry Date:' | awk NR==1{'print $4}'"
set cell "domain_exp_date" of current record of layout "Expiration_Details" to expdate
set expmonth to do shell script "echo | whois " & DOMAINNAME & " | grep 'Registry Expiry Date:' | awk NR==1{'print $4}' | cut -c 6-7"
set cell "domain_exp_month" of current record of layout "Expiration_Details" to expmonth
set expday to do shell script "echo | whois " & DOMAINNAME & " | grep 'Registry Expiry Date:' | awk NR==1{'print $4}' | cut -c 9-10"
set cell "domain_exp_day" of current record of layout "Expiration_Details" to expday
set expyear to do shell script "echo | whois " & DOMAINNAME & " | grep 'Registry Expiry Date:' | awk NR==1{'print $4}' | cut -c 1-4"
set cell "domain_exp_year" of current record of layout "Expiration_Details" to expyear
set domainregistrar to do shell script "echo | whois " & DOMAINNAME & " | grep 'Registrar:' | awk NR==1{'print $2,$3,$4,$5}'"
set cell "domain_registrar" of current record of layout "Expiration_Details" to domainregistrar
set registrarurl to do shell script "echo | whois " & DOMAINNAME & " | grep 'URL:' | awk NR==1{'print $3,$4,$5}'"
set cell "notes" of current record of layout "Expiration_Details" to registrarurl
```

precursor.ca/slides

Here is an example of the script to get the info for a domain name.
CLICK



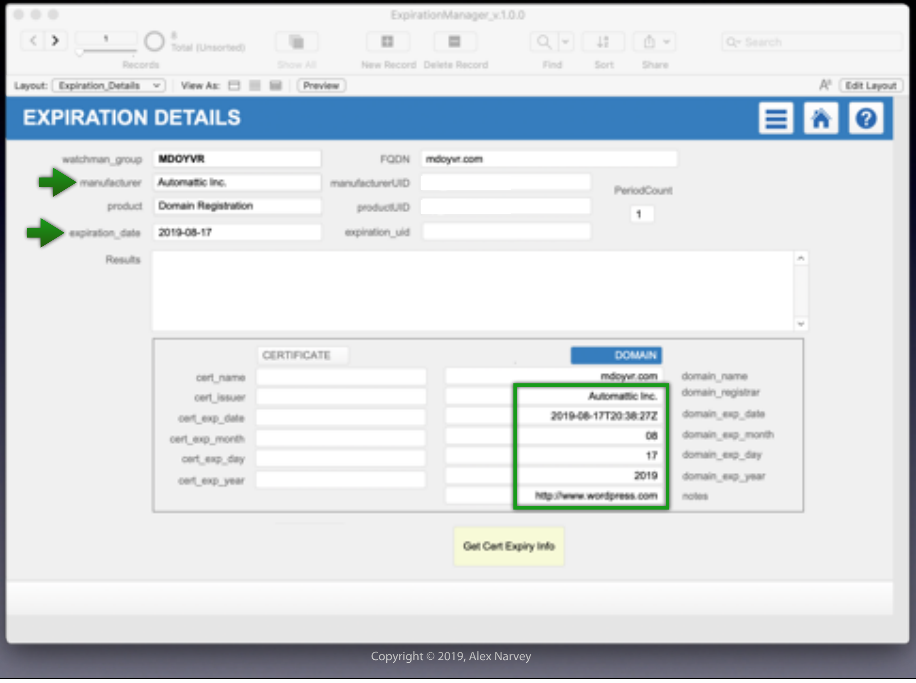
EXPIRATION DETAILS

watchman_group: MDOYVR
FQDN: mdoyvr.com
manufacturer:
manufacturerUID:
product: Domain Registration
productUID:
PeriodCount: 1
expiration_date:
expiration_uid:
Results:

cert_name:
cert_issuer:
cert_exp_date:
cert_exp_month:
cert_exp_day:
cert_exp_year:

domain_name: mdoyvr.com
domain_registrar:
domain_exp_date:
domain_exp_month:
domain_exp_day:
domain_exp_year:
notes:

We click the button to start the script and it loops through the records and…
CLICK



EXPIRATION DETAILS

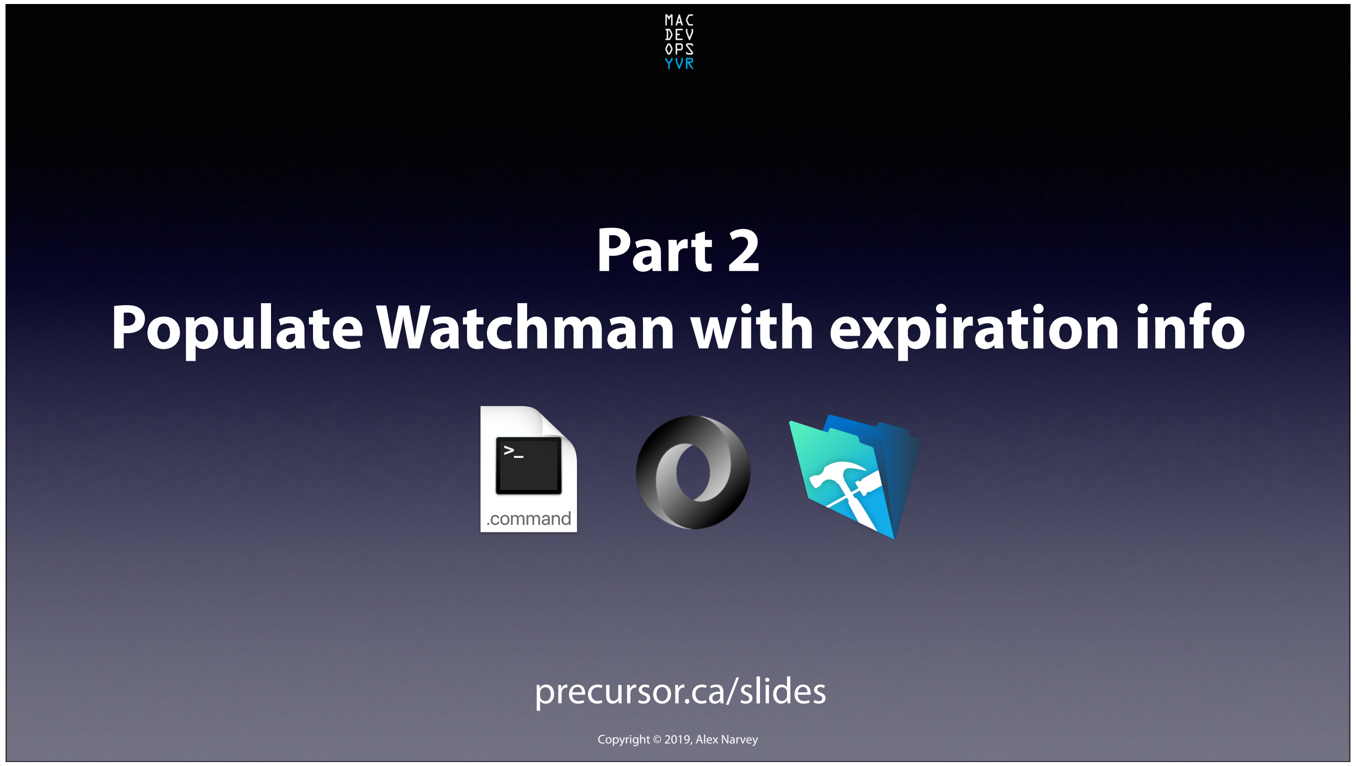
watchman_group	MDOYVR	FQDN	mdoyvr.com
manufacturer	Automattic Inc.	manufacturerUID	
product	Domain Registration	productUID	
expiration_date	2019-08-17	expiration_uid	

PeriodCount: 1

Results

CERTIFICATE		DOMAIN	
cert_name		mdoyvr.com	domain_name
cert_issuer		Automattic Inc.	domain_registrar
cert_exp_date		2019-08-17T20:38:27Z	domain_exp_date
cert_exp_month		08	domain_exp_month
cert_exp_day		17	domain_exp_day
cert_exp_year		2019	domain_exp_year
		http://www.wordpress.com	notes

Voila! We have the expiration information.
CLICK



MAC DEV OPS YVR

Part 2
Populate Watchman with expiration info

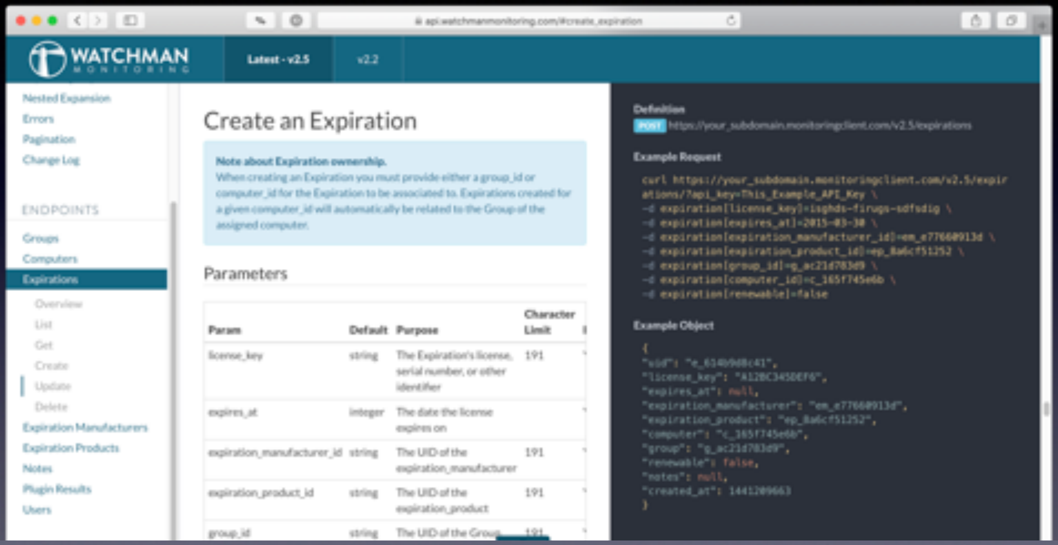
precursor.ca/slides

Now all we have to do is push that info up to our Watchman Monitoring Dashboard.

But how can we do that?

CLICK

MAC
DEV
OPS
YVR



Create an Expiration

Note about Expiration ownership.
When creating an Expiration you must provide either a group_id or computer_id for the Expiration to be associated to. Expirations created for a given computer_id will automatically be related to the Group of the assigned computer.

Parameters

Param	Default	Purpose	Character Limit
license_key	string	The Expiration's license, serial number, or other identifier	191
expires_at	integer	The date the license expires on	
expiration_manufacturer_id	string	The UID of the expiration_manufacturer	191
expiration_product_id	string	The UID of the expiration_product	191
group_id	string	The UID of the Group	191

Definition

POST https://your_subdomain.monitoringclient.com/v2.5/expirations

Example Request

```
curl https://your_subdomain.monitoringclient.com/v2.5/expir
ations/?api_key=This_Example_API_Key \
-d expiration[license_key]=isghds-firugs-sdfsdig \
-d expiration[expires_at]=2015-03-30 \
-d expiration[expiration_manufacturer_id]=em_e77660913d \
-d expiration[expiration_product_id]=ep_8a6cf51252 \
-d expiration[group_id]=g_ac21d783d9 \
-d expiration[computer_id]=c_165f745e6b \
-d expiration[renewable]=false
```

Example Object

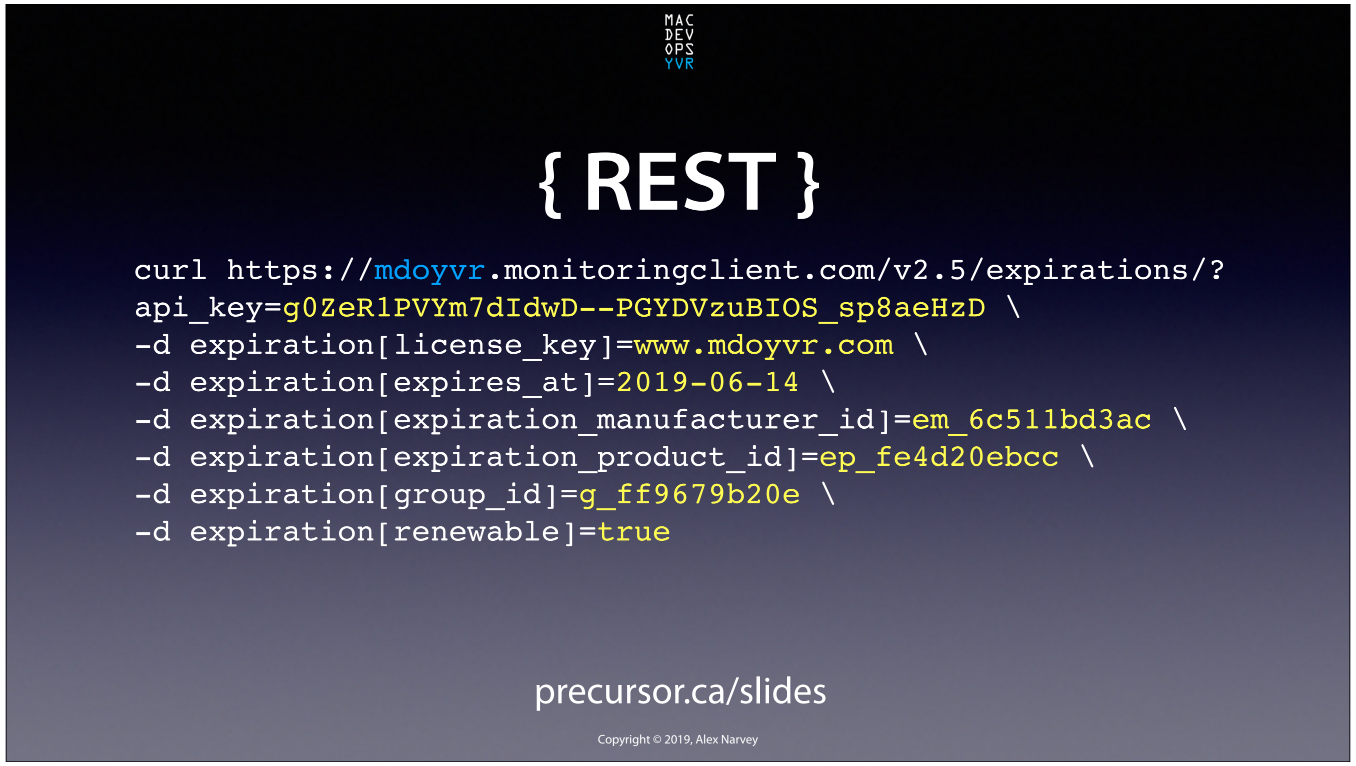
```
{
"uid": "e_614b9d8c41",
"license_key": "A12BC345DEF6",
"expires_at": null,
"expiration_manufacturer": "em_e77660913d",
"expiration_product": "ep_8a6cf51252",
"computer": "c_165f745e6b",
"group": "g_ac21d783d9",
"renewable": false,
"notes": null,
"created_at": 1441209663
}
```

precursor.ca/slides

Watchman Monitoring has a REST API to interact with.

And it has excellent documentation with plenty of detailed examples.

CLICK



MAC
DEV
OPS
YVR

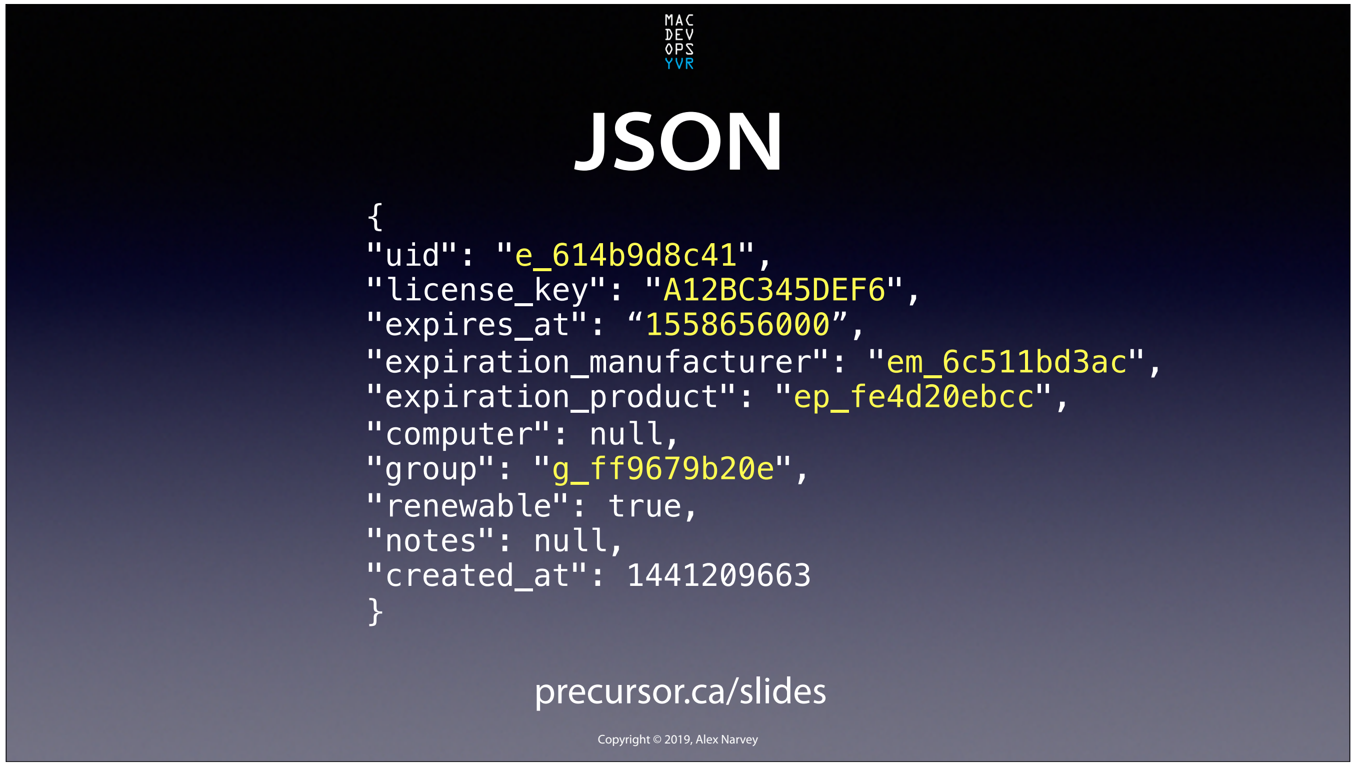
{ REST }

```
curl https://mdoyvr.monitoringclient.com/v2.5/expirations/?
api_key=g0ZeR1PVYm7dIdwD--PGYDVzuBIOS_sp8aeHzD \
-d expiration[license_key]=www.mdoyvr.com \
-d expiration[expires_at]=2019-06-14 \
-d expiration[expiration_manufacturer_id]=em_6c511bd3ac \
-d expiration[expiration_product_id]=ep_fe4d20ebcc \
-d expiration[group_id]=g_ff9679b20e \
-d expiration[renewable]=true
```

precursor.ca/slides

According to Watchman we have to make a POST that looks something like this.

CLICK



JSON

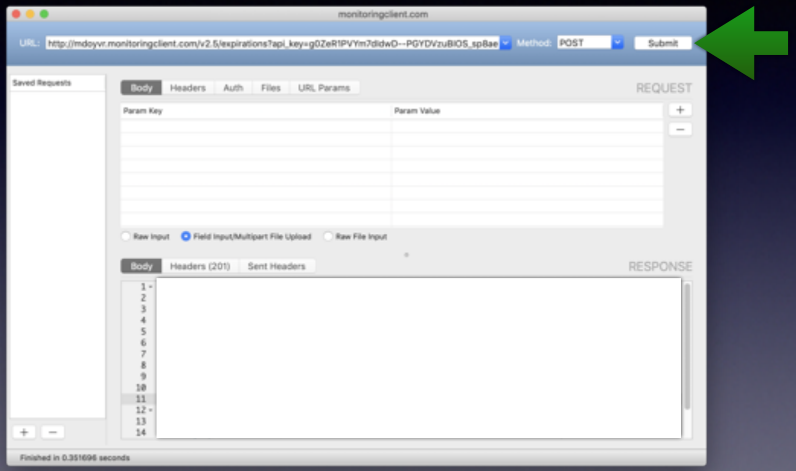
```
{
"uid": "e_614b9d8c41",
"license_key": "A12BC345DEF6",
"expires_at": “1558656000”,
"expiration_manufacturer": "em_6c511bd3ac",
"expiration_product": "ep_fe4d20ebcc",
"computer": null,
"group": "g_ff9679b20e",
"renewable": true,
"notes": null,
"created_at": 1441209663
}
```

precursor.ca/slides

And if our POST is successful it will populate Watchman with the info and return the following JSON to us.
CLICK

MAC
DEV
OPS
YVR

https://github.com/mmattozzi/cocoa-rest-client


URL: http://mdoyvr.monitoringclient.com/v2.5/expirations?api_key=g0ZeR1PVYm7didwD--PGYDVzuBIOS_spBae

Method: POST

○ Raw Input ● Field Input/Multipart File Upload ○ Raw File Input

Finished in 0.351696 seconds

precursor.ca/slides

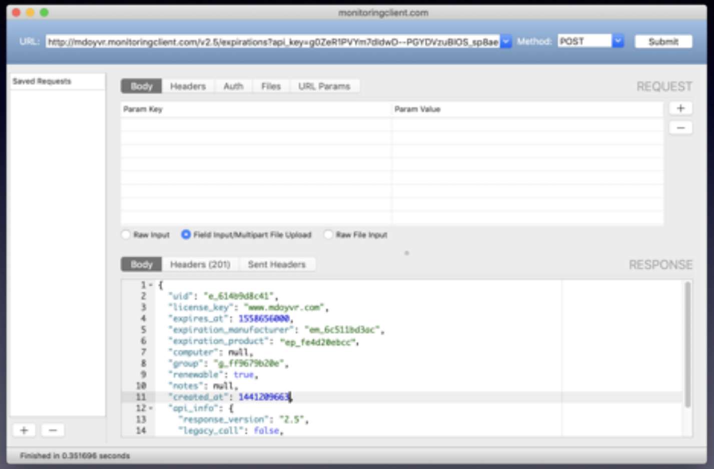
To test this out and practice I used the Cocoa Rest Client.

I just plugged the appropriate values into the URL line at the top and click the “Submit” button.

CLICK

MAC
DEV
OPS
YVR

https://github.com/mmattozzi/cocoa-rest-client



URL: http://mdoyvr.monitoringclient.com/v2.5/expirations?api_key=g0ZeR1PVYm7didwD--PGYDVzuBIOS_spBae

Method: POST

○ Raw Input ● Field Input/Multipart File Upload ○ Raw File Input

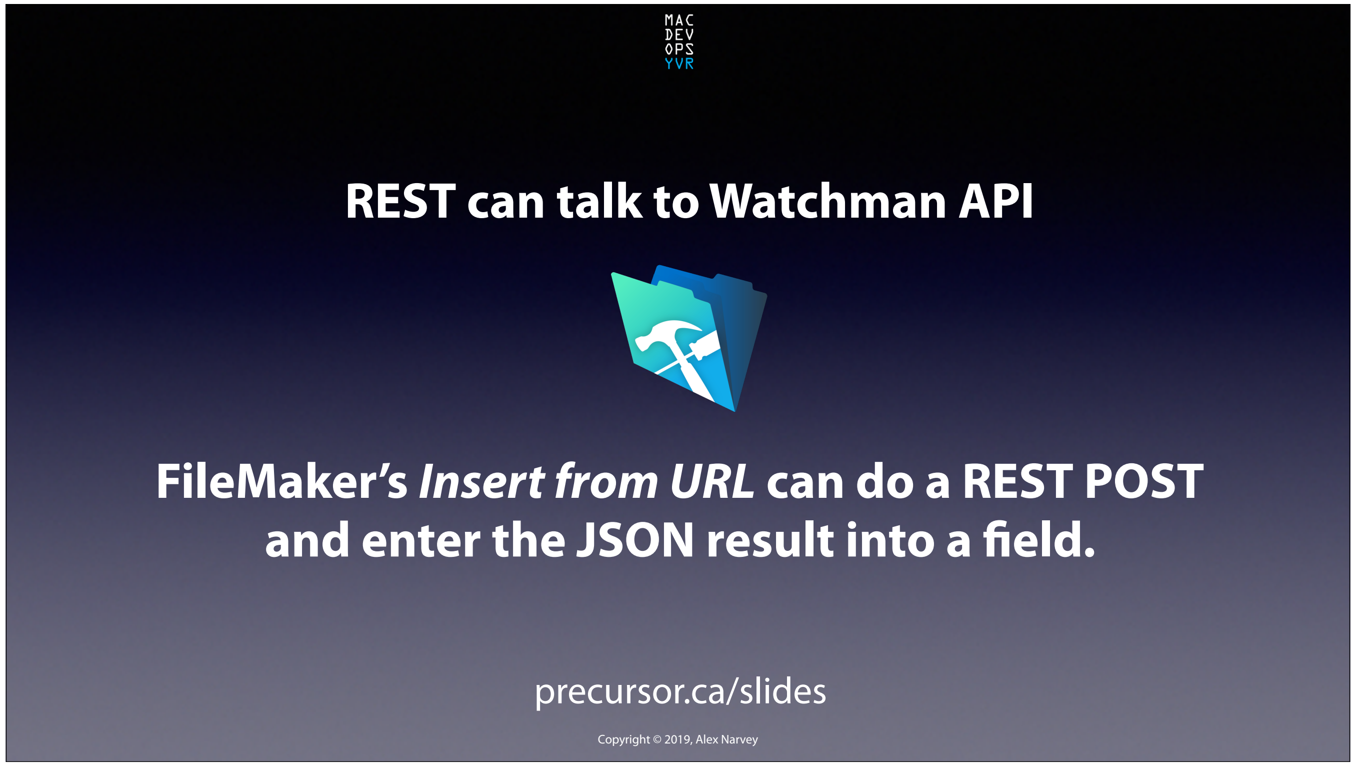
```
{
  "uid": "e_614b9d8c41",
  "license_key": "www.mdoyvr.com",
  "expires_at": 1558656000,
  "expiration_manufacturer": "em_6c511bd3ac",
  "expiration_product": "ep_fe4d20ebcc",
  "computer": null,
  "group": "g_ff9679b20e",
  "renewable": true,
  "notes": null,
  "created_at": 1441209663,
  "api_info": {
    "response_version": "2.5",
    "legacy_call": false,
```

Finished in 0.351696 seconds

precursor.ca/slides

Copyright © 2019, Alex Narvey

And sure enough, I got a valid JSON file returned with all the correct details.
CLICK



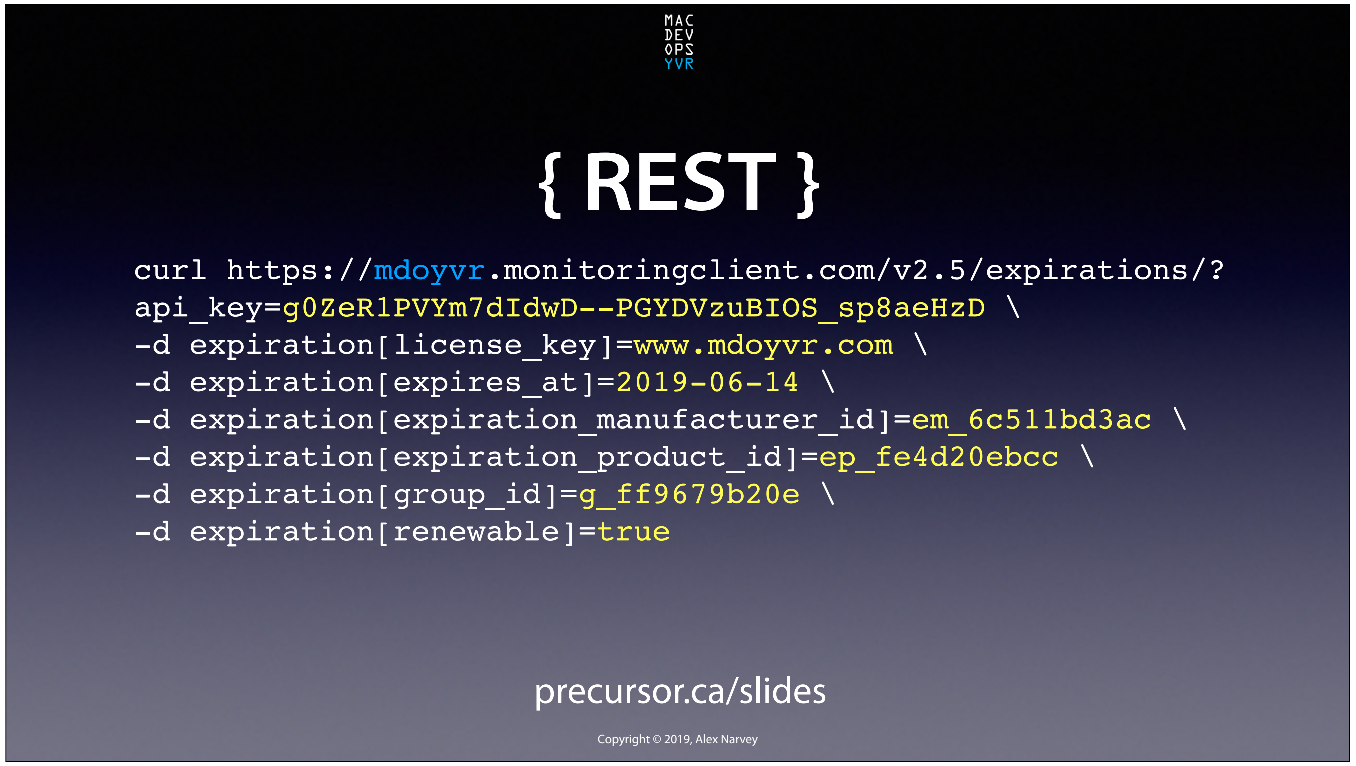
MAC DEV OPS YVR

REST can talk to Watchman API

FileMaker’s *Insert from URL* can do a REST POST and enter the JSON result into a field.

precursor.ca/slides

So REST can talk to Watchman Monitoring and FileMaker has an *Insert from URL* command that can do a POST and then enter the resulting JSON into FileMaker fields.
CLICK



MAC
DEV
OPS
YVR

{ REST }

```
curl https://mdoyvr.monitoringclient.com/v2.5/expirations/?
api_key=g0ZeR1PVYm7dIdwD--PGYDVzuBIOS_sp8aeHzD \
-d expiration[license_key]=www.mdoyvr.com \
-d expiration[expires_at]=2019-06-14 \
-d expiration[expiration_manufacturer_id]=em_6c511bd3ac \
-d expiration[expiration_product_id]=ep_fe4d20ebcc \
-d expiration[group_id]=g_ff9679b20e \
-d expiration[renewable]=true
```

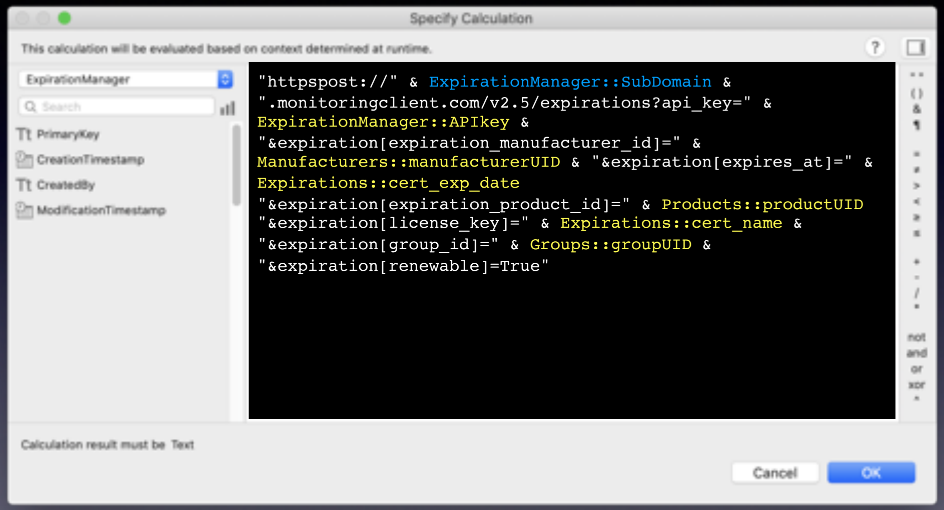
precursor.ca/slides

Now here’s that example of a REST POST command from the Watchman documention…

CLICK

MAC
DEV
OPS
YVR

POST



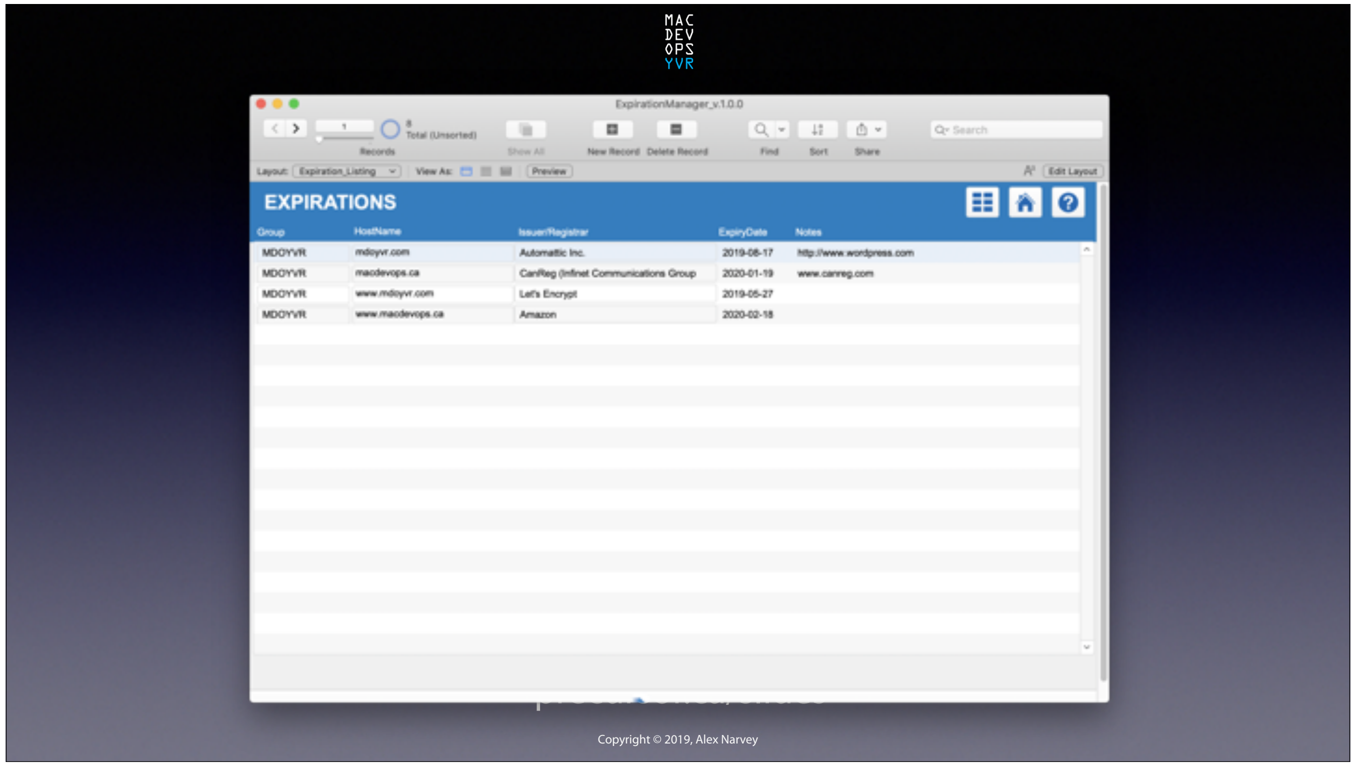
This calculation will be evaluated based on context determined at runtime.

```
"httpspost://" & ExpirationManager::SubDomain &
".monitoringclient.com/v2.5/expirations?api_key=" &
ExpirationManager::APIkey &
"&expiration[expiration_manufacturer_id]=" &
Manufacturers::manufacturerUID & "&expiration[expires_at]=" &
Expirations::cert_exp_date
"&expiration[expiration_product_id]=" & Products::productUID
"&expiration[license_key]=" & Expirations::cert_name &
"&expiration[group_id]=" & Groups::groupUID &
"&expiration[renewable]=True"
```

Calculation result must be Text

precursor.ca/slides

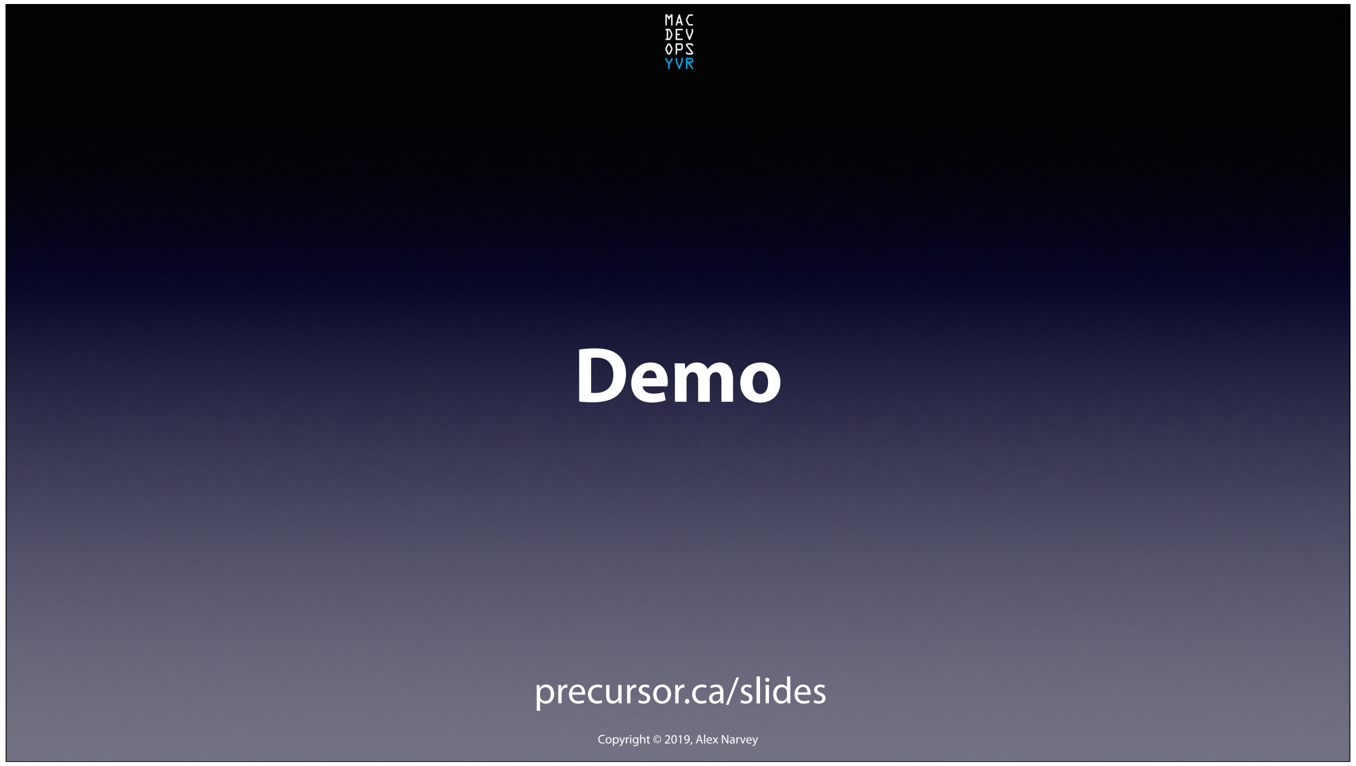
And here is what it looks like in the "*Insert from URL*" command within FileMaker.
CLICK



EXPIRATIONS

Group	HostName	Issuer/Registrar	ExpiryDate	Notes
MDOYVR	mdoyvr.com	Automattic Inc.	2019-08-17	http://www.wordpress.com
MDOYVR	macdevops.ca	CanReg (Infinet Communications Group	2020-01-19	www.canreg.com
MDOYVR	www.mdoyvr.com	Let's Encrypt	2019-05-27	
MDOYVR	www.macdevops.ca	Amazon	2020-02-18	

Here’s a look at all the records that got created in Watchman and recorded to FIleMaker when we clicked that button.
CLICK



MAC
DEV
OPS
YVR

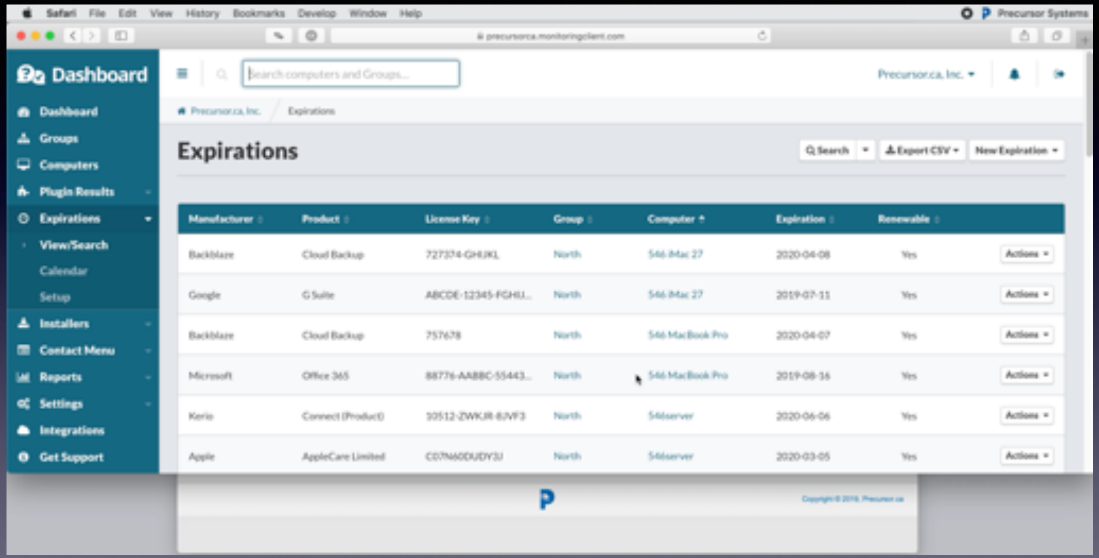
Demo

precursor.ca/slides

And now we just barely have time for a Demo.

The demo is pre-recorded; but it has NOT been sped up. This is in actual time.

CLICK



Expirations

Manufacturer	Product	License Key	Group	Computer	Expiration	Renewable	
Backblaze	Cloud Backup	727374-GHIJKL	North	546 iMac 27	2020-04-08	Yes	Actions
Google	G Suite	ABCDE-12345-FGHIJ...	North	546 iMac 27	2019-07-11	Yes	Actions
Backblaze	Cloud Backup	757678	North	546 MacBook Pro	2020-04-07	Yes	Actions
Microsoft	Office 365	88776-AABBC-55443...	North	546 MacBook Pro	2019-08-16	Yes	Actions
Kerio	Connect (Product)	10512-ZWKJR-8JVF3	North	546server	2020-06-06	Yes	Actions
Apple	AppleCare Limited	C07N60DUDY3J	North	546server	2020-03-05	Yes	Actions

precursor.ca/slides

You can see a Watchman Dashboard with some computer expirations. In this particular web view the Domain and Certificate expiration would appear at the top if there were any. So let’s start the video and make some.

CLICK

First we import the records from our spreadsheet…

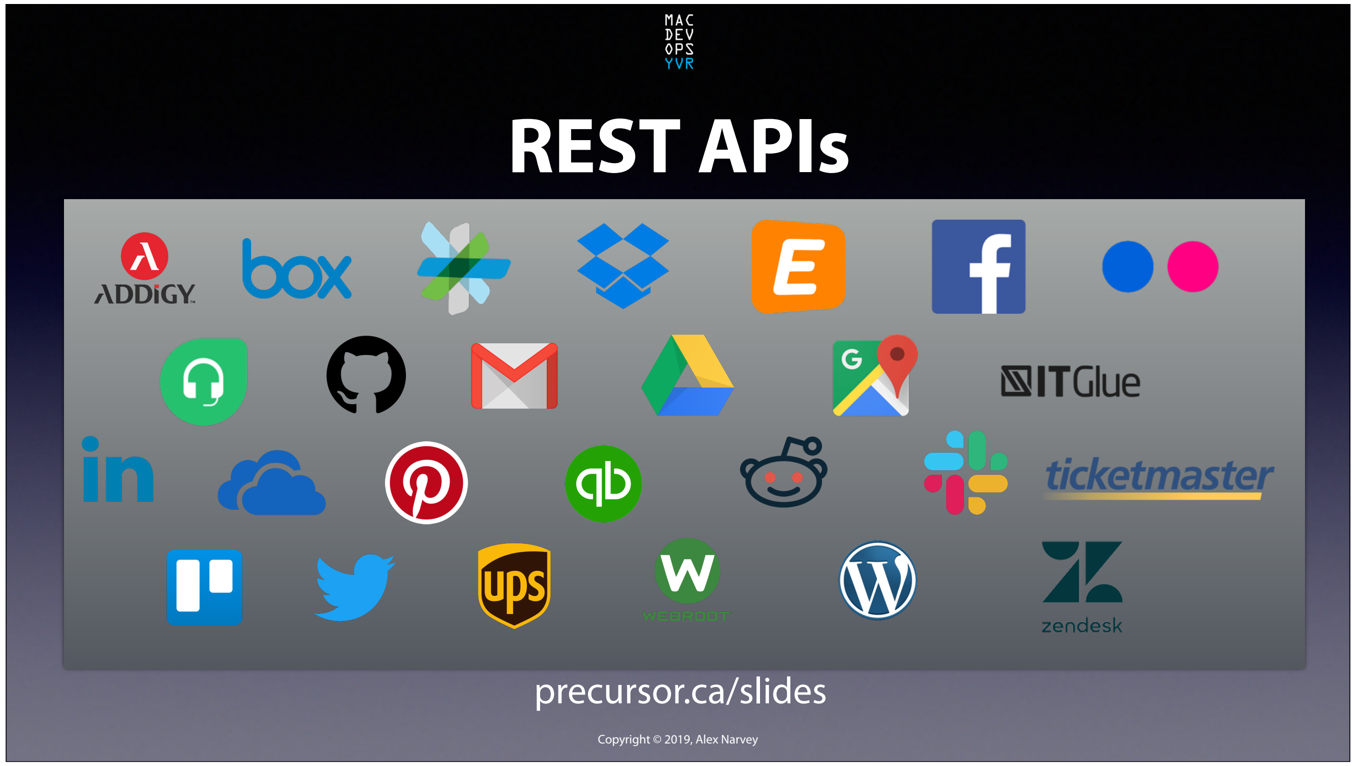
Now we get the expiration info. The first two are Domains and will appear on the right. The second two are Certificates and will appear on the left. In the real world when I used this on my actual Watchman Dashboard it took about 3-1/2 minutes to get the info for 65 domains and certs and 3 seconds to push them up to the Dashboard.

Now we are ready to add them Watchman…

They have been added to Watchman so I will refresh the page so you can see them…

TADA!

CLICK



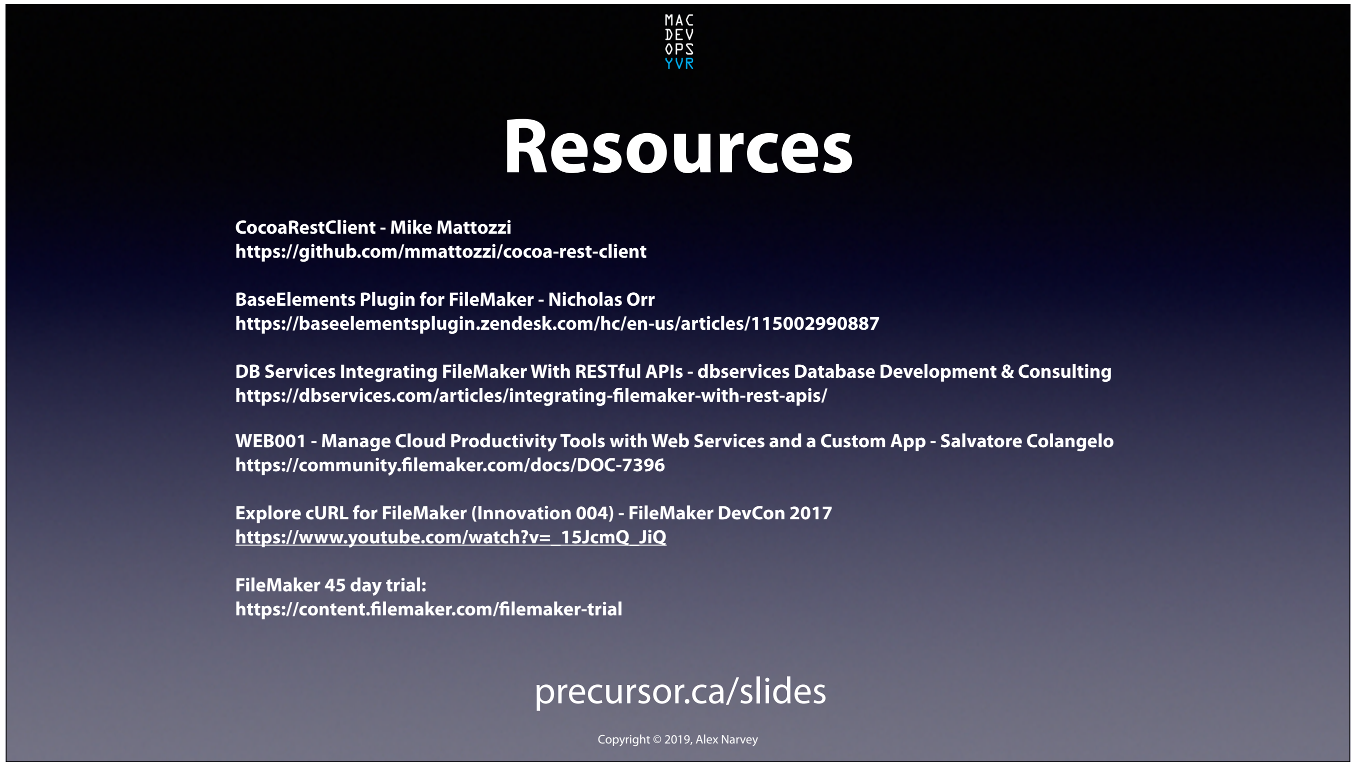
REST APIs

precursor.ca/slides

Watchman isn’t the only online service that has a REST API. In fact, it is harder to find an online service that does NOT have one.

So FileMaker might be a good tool for you to work with to integrate your various services.

CLICK



MAC
DEV
OPS
YVR

Resources

CocoaRestClient - Mike Mattozzi
https://github.com/mmattozzi/cocoa-rest-client

BaseElements Plugin for FileMaker - Nicholas Orr
https://baseelementsplugin.zendesk.com/hc/en-us/articles/115002990887

DB Services Integrating FileMaker With RESTful APIs - dbservices Database Development & Consulting
https://dbservices.com/articles/integrating-filemaker-with-rest-apis/

WEB001 - Manage Cloud Productivity Tools with Web Services and a Custom App - Salvatore Colangelo
https://community.filemaker.com/docs/DOC-7396

Explore cURL for FileMaker (Innovation 004) - FileMaker DevCon 2017
https://www.youtube.com/watch?v=_15JcmQ_JiQ

FileMaker 45 day trial:
https://content.filemaker.com/filemaker-trial

precursor.ca/slides

Here are some resources including some good presentations from FileMaker Devcon which are more detailed.

FileMaker is free if you are a member of the Apple Consultants Network or a FileMaker Reseller.

And it is also available as a 45 day trial.

Don’t worry about writing these url’s down as the slides are available at **precursor.ca/sildes**

CLICK

MAC
DEV
OPS
YVR

Alex Narvey

https://Precursor.ca

@precursorca

precursor.ca/slides

Thanks for listening. I hope you find this useful.
THE END.