



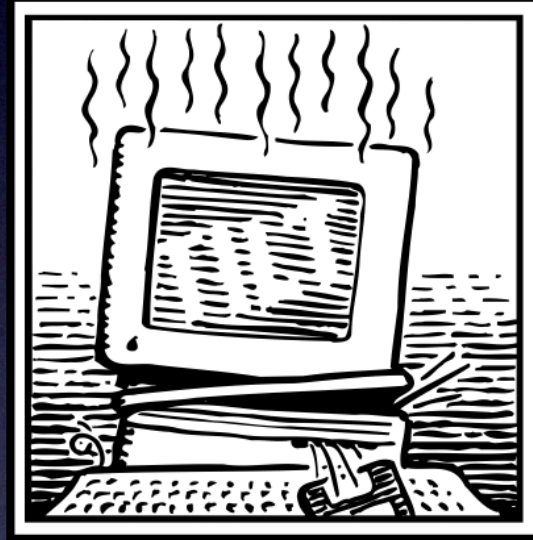
arRESTed Development

Using REST APIs for Monthly Reporting

precursor.ca/slides



Precursor Systems



EST'd 1994

precursor.ca/slides



“Alex Narvey is the original Mac MSP”

- Allen Hancock

precursor.ca/slides



Mac *mini* Dev Ops

precursor.ca/slides



Basics

Interacting with REST APIs

Examples

Monthly Reporting with REST

precursor.ca/slides



Software Patches —

Support Tickets —

Warnings —

DNS Filtering —



— Malware

— macOS Version

— MDM Devices

— Hardware Age

precursor.ca/slides



Here is your executive summary for last month.

For more detail on how to interpret these results please see :

<https://precursor.ca/pmr>

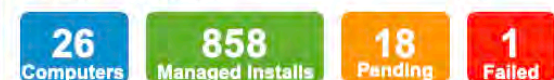
Sincerely,

Alex Narvey
Precursor Systems

Monthly Report

Managed Software Center

SOFTWARE PATCHING



DetectX

MALWARE SCANNING



ZenDesk

TROUBLE TICKETS



Watchman

HEALTH REPORTS



macOS



■ 11.06.1 ■ 11.06 ■ 10.15.7

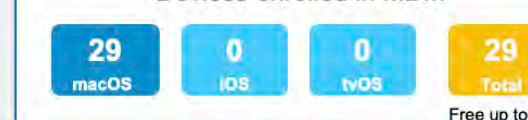
DNSFilter

HEALTH REPORTS



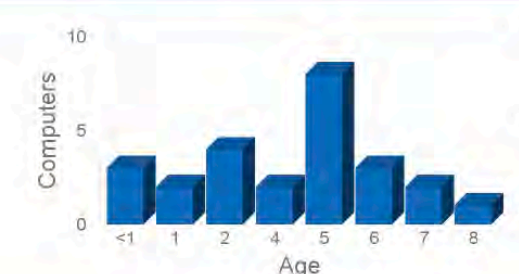
MOSYLE

Devices enrolled in MDM



Asset Age

Suggest replacement budget for this year:
\$18,000.00



precursor.ca/slides



The screenshot shows a web browser window with the URL `precursor.ca/pmr/`. The page features the Precursor.ca logo at the top, followed by navigation links for ABOUT, SUPPORT, SALES, and RESOURCES. The main heading is "Interpreting Your Precursor Monthly Report". Below this is a breadcrumb trail: [Overview](#) - [Managed Software](#) - [DetectX](#) - [Zendesk](#) - [Watchman Monitoring](#) - [macOS](#) - [DNSFilter](#) - [Mosyle MDM](#) - [Asset Age](#) - [FAQ](#). The "Overview" section is highlighted in blue and contains the following text: "The Precursor Systems Monthly Report is an executive summary to help you understand what is happening support-wise and maintenance-wise with your technology. We gather this information from various services and tools and present it in an easy to view and understand format." Below this is a "Managed Software" section with a "Managed Software Center" box. This box displays "SOFTWARE PATCHING" statistics: 24 Computers, 760 Managed Installs, 15 Pending, and 0 Failed. A paragraph below the box explains that Managed Software Center is used by Disney and other corporations to keep devices provisioned with curated and fully patched software applications. The text "Here you can see:" follows.

precursor.ca/slides



REST APIs

precursor.ca/slides



“The world is just a collection of REST endpoints.”

Charles Edge

precursor.ca/slides



{ REST }

Representational State Transfer

precursor.ca/slides



cURL

precursor.ca/slides



cURL, which stands for client URL, is a command line tool that developers use to transfer data to and from a server.

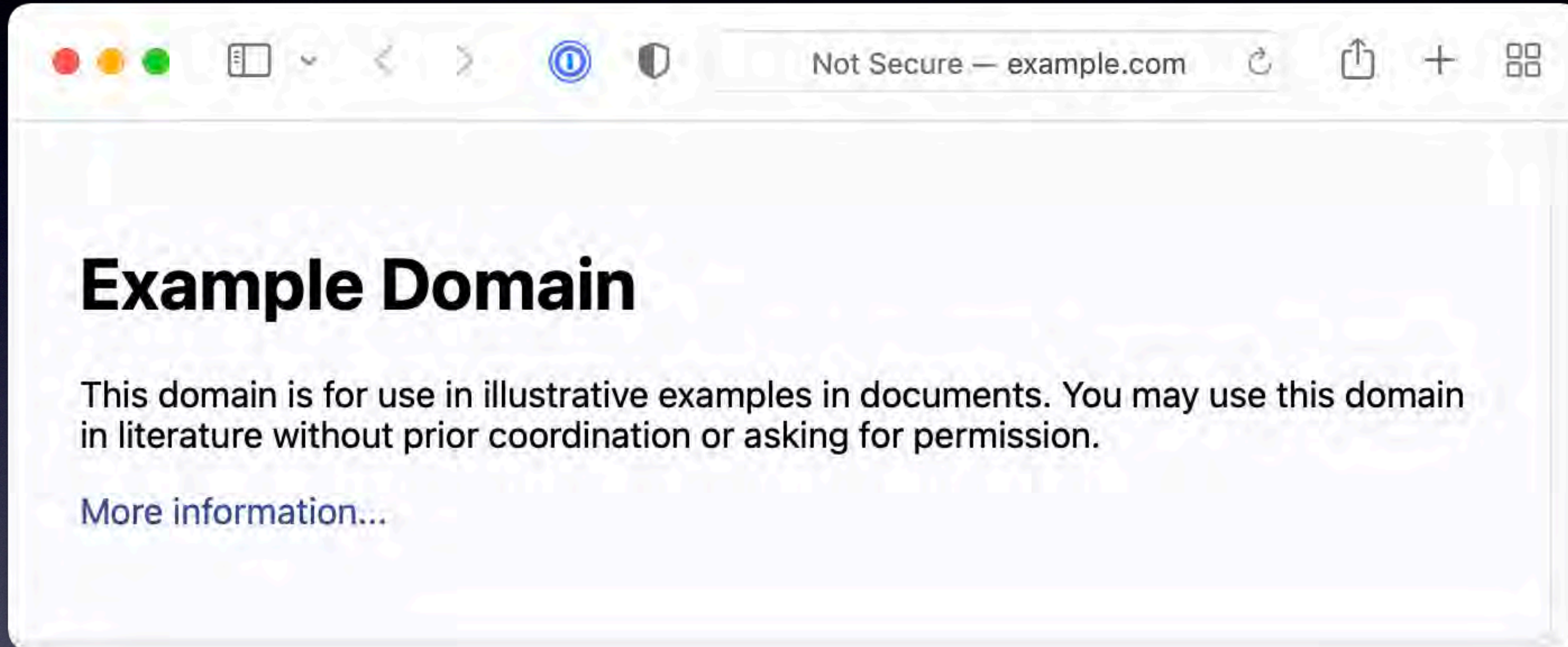
precursor.ca/slides



The most basic command in curl is:

```
curl http://example.com
```

precursor.ca/slides



precursor.ca/slides



```
<!doctype html>
<html>
<head>
  <title>Example Domain</title>

  <meta charset="utf-8" />
  <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <style type="text/css">
body {
  background-color: #f0f0f2;
  margin: 0;
  padding: 0;
  font-family: -apple-system, system-ui, BlinkMacSystemFont, "Segoe UI", "Open Sans",
"Helvetica Neue", Helvetica, Arial, sans-serif;
}
div {
  width: 600px;
  margin: 5em auto;
  padding: 2em;
  background-color: #fdfdff;
  border-radius: 0.5em;
  box-shadow: 2px 3px 7px 2px rgba(0,0,0,0.02);
}
a:link, a:visited {
```

precursor.ca/slides



JavaScript Object Notation

precursor.ca/slides



Buildinfo.json

```
{
  "distribution_style": true,
  "identifier": "com.erikng.installapplications",
  "install_location": "/",
  "name": "InstallApplications-${version}.pkg",
  "ownership": "recommended",
  "postinstall_action": "none",
  "signing_info": {
    "identity": "Mac Installer: Erik Gomez (XXXXXXXXXXXX)",
    "timestamp": true
  },
  "suppress_bundle_relocation": true,
  "version": "2.1"
}
```

precursor.ca/slides



cURL `https://businessapi.mosyle.com/v1/users`

precursor.ca/slides



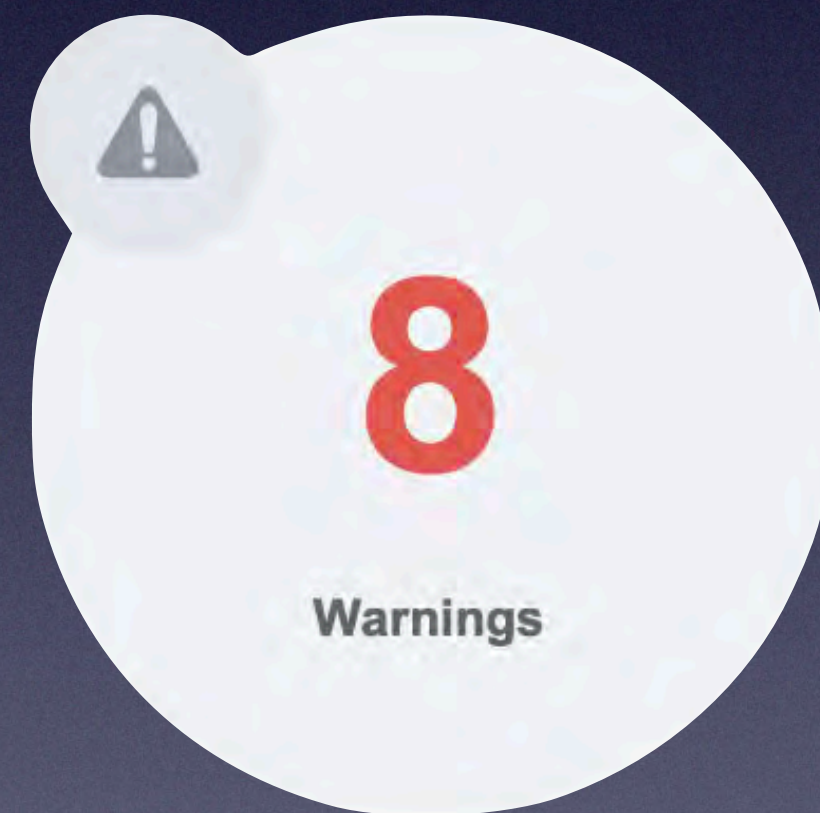
```
{
  "status": "OK",
  "response": [
    {
      "users": [
        {
          "iduser": "69",
          "code": "2QTBH",
          "name": "Alex Narvey",
          "type": "ADMIN",
          "identifier": "alex",
          "email": "anarvey@precursor.ca",
          "is_removed": false
        }
      ],
      "page": 1
    }
  ]
}
```

precursor.ca/slides



cURL

```
https://exampleco.monitoringclient.com/v2.5/computers/?  
api_key="123456789"&group_id="24"&expand[]=plugin_results"
```



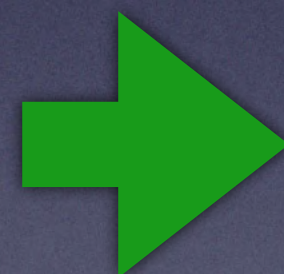
precursor.ca/slides



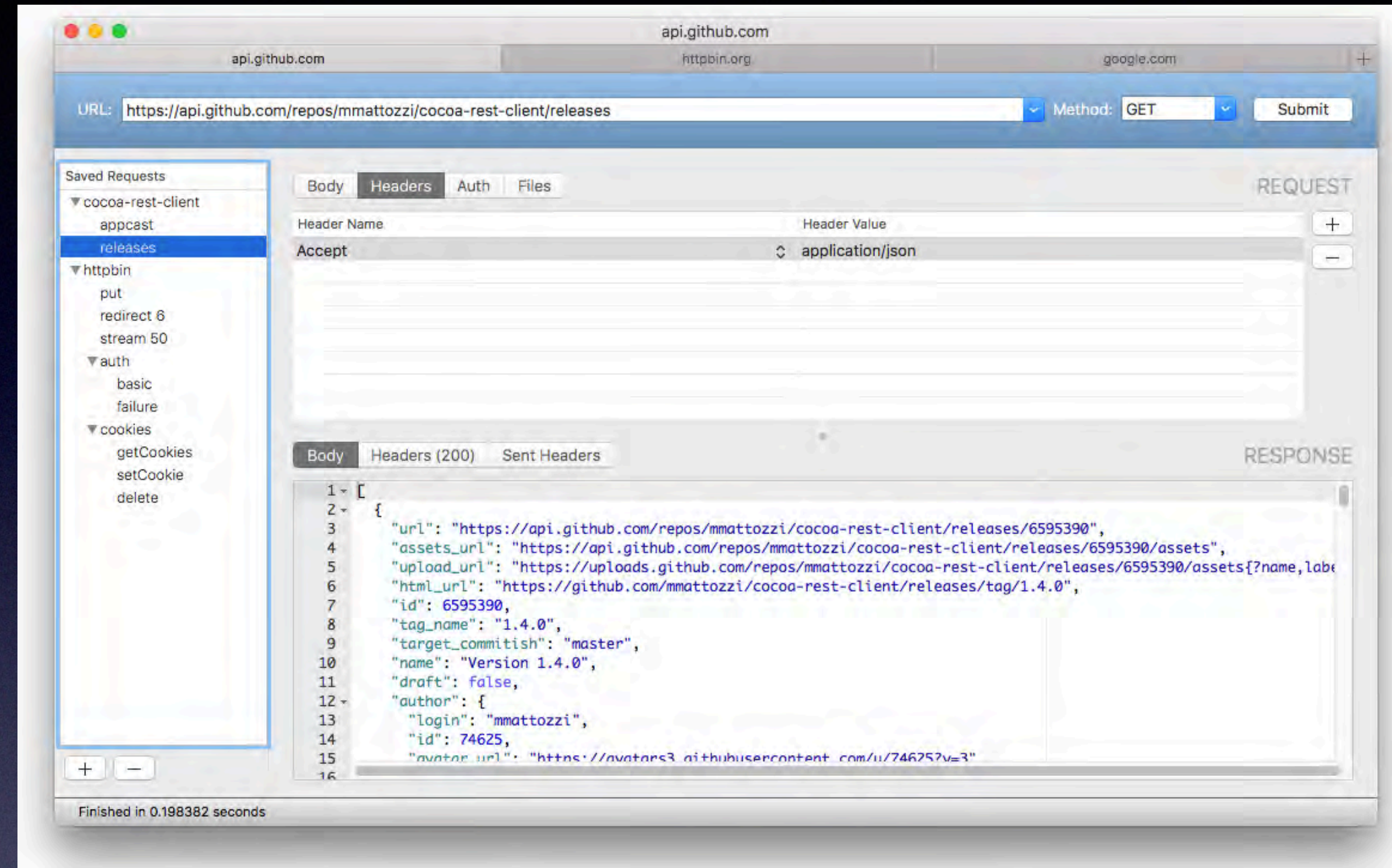
```
1 {
2   "plugins" :
3   [
4     {
5       "active_mac_address" : "A8:60:B6:03:E5:91",
6       "agent_removed" : false,
7       "agent_version" : "6.6.9.101",
8       "apple_product_description" : "Mac mini (Late 2014)",
```

...

```
14710     "teamviewer_id" : null,
14711     "teamviewer_release" : null,
14712     "teamviewer_unattended" : null,
14713     "uid" : "c_5553c4e38a",
14714     "updates_enabled" : true,
14715     "watchman_id" : "20160106-BF59-UN4R8B"
14716   }
14717 ]
14718 }
```

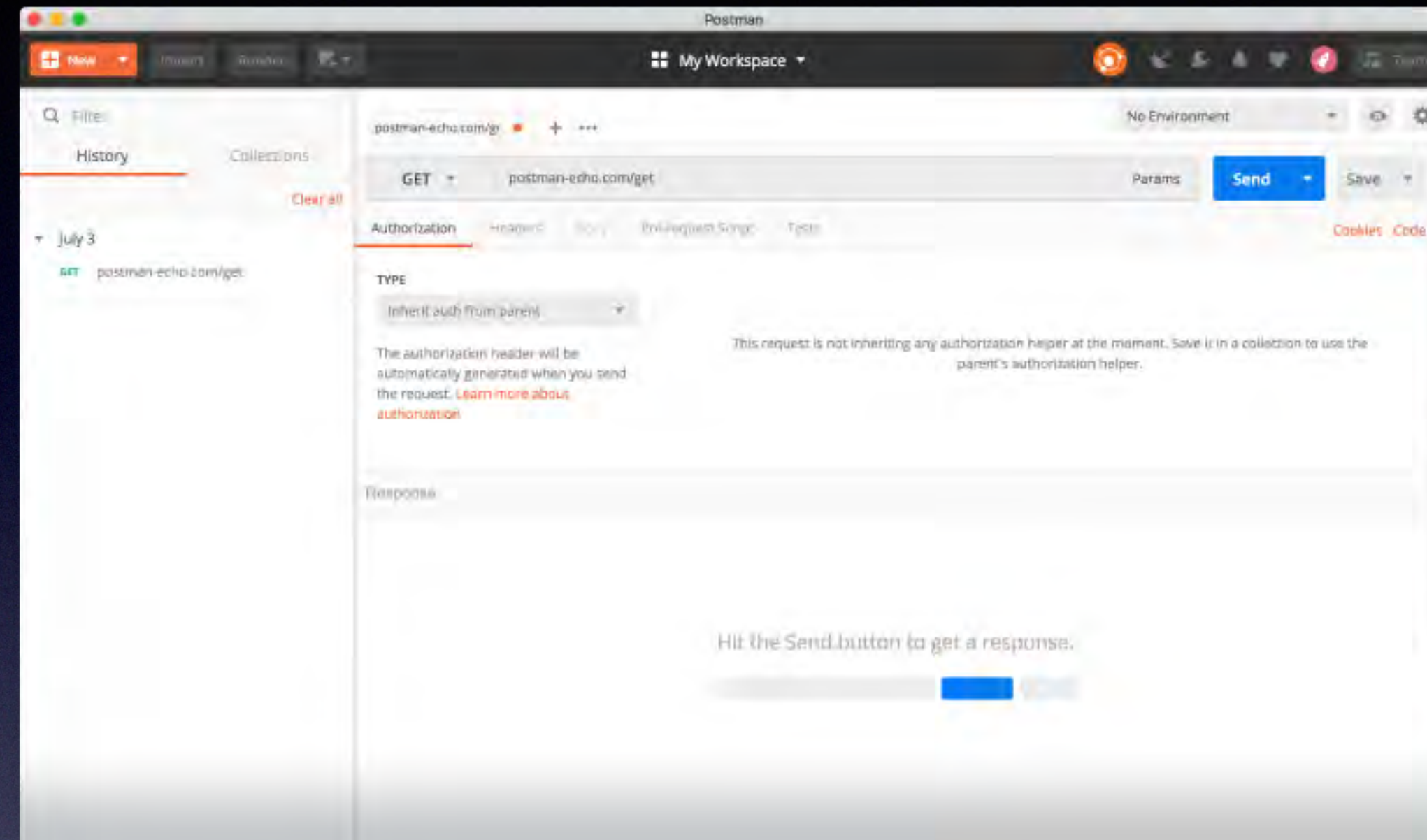


precursor.ca/slides



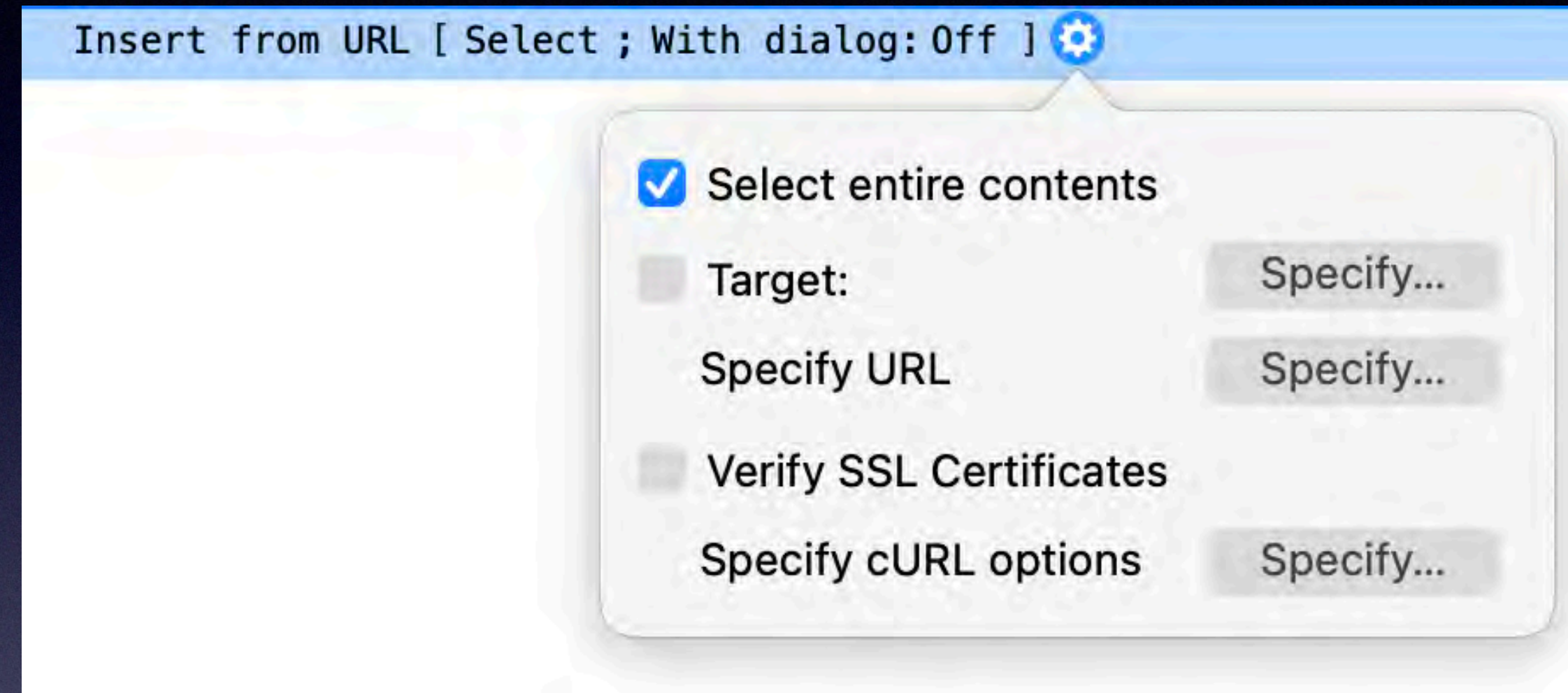
<https://mmattozzi.github.io/cocoa-rest-client/>

[precursor.ca/slides](https://mmattozzi.github.io/cocoa-rest-client/precursor.ca/slides)



<https://www.getpostman.com/downloads/>

precursor.ca/slides



<https://claris.com/filemaker/>

precursor.ca/slides



precursor.ca/slides



Insert From URL

Takes cURL commands

precursor.ca/slides



Target

Verify SSL Certificates
 Select entire contents
 Target:
Specify URL
Specify cURL options



FileMaker Field

FileMaker Variable

precursor.ca/slides



URL

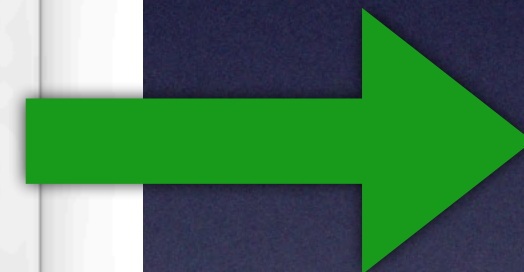
Verify SSL Certificates

Select entire contents

Target:

Specify URL

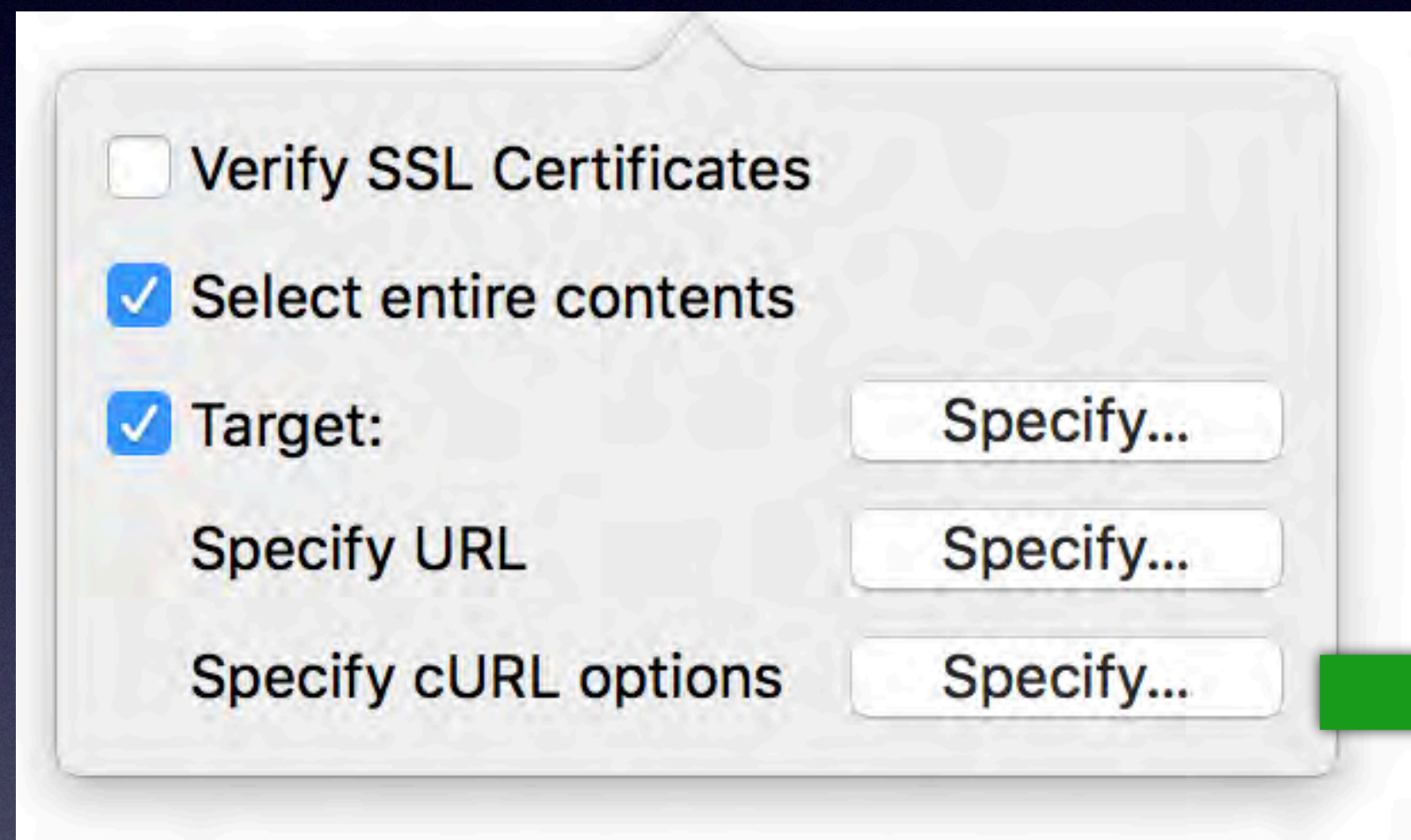
Specify cURL options



`https://api.example.com/v1/groups`



cURL Options



Header
Data
Form data
Authentication
Cookies

precursor.ca/slides



“Getting started with the new cURL options is quite daunting and a steep learning curve.”

<https://support.claris.com/s/article/new-curl-options-in-insert-from-url>

precursor.ca/slides



Basic GET Command

Authorization in URL

precursor.ca/slides

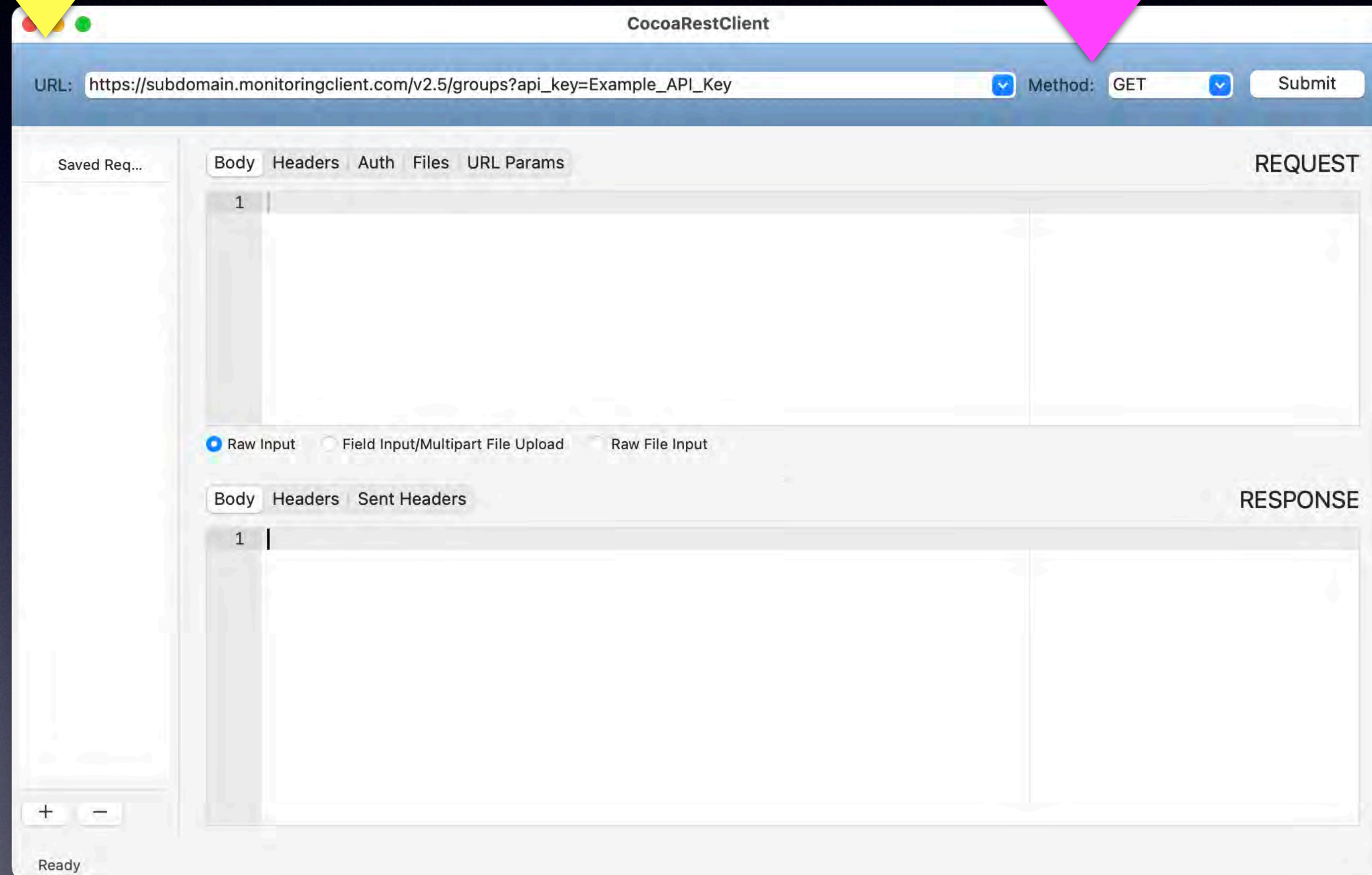


GET `curl https://subdomain.monitoringclient.com/v2.5/groups?api_key=Example_API_Key`

precursor.ca/slides



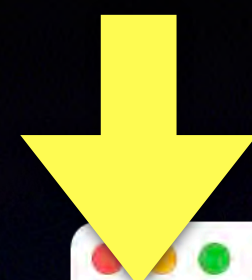
`curl https://subdomain.monitoringclient.com/v2.5/groups?api_key=Example_API_Key`



precursor.ca/slides



`curl https://subdomain.monitoringclient.com/v2.5/groups?api_key=Example_API_Key`



CocoaRestClient

URL: `https://subdomain.monitoringclient.com/v2.5/groups?api_key=Example_API_Key` Method: GET Submit

Body Headers Auth Files URL Params REQUEST

1

Raw Input Field Input/Multipart File Upload Raw File Input

Body Headers Sent Headers RESPONSE

```
{
  "contact_email": null,
  "created_at": 1504211891,
  "description": "This is the Group's Description",
  "hidden": false,
  "hidden_computer_count": 0,
  "mac_installer_ready": false,
  "mac_installer_url": "https://monitoringclient.s3.amazonaws.com/SubscriberInstallers/ors/MonitoringClient-group-name.pk",
  "name": "[Blank]",
  "reference_email": null,
  "show_contact_menu": true,
  "slug": "blank",
  "uid": "g_56257078ed",
  "visible_computer_count": 2,
  "windows_installer_ready": false,
  "windows_installer_url": "https://monitoringclient.s3.amazonaws.com/SubscriberInstallers/ors/MonitoringClient-group-name.pk"
}
```

Ready

precursor.ca/slides



https://subdomain.monitoringclient.com/v2.5/groups?api_key=Example_API_Key



Insert From URL script step

precursor.ca/slides



https://subdomain.monitoringclient.com/v2.5/groups?api_key=Example_API_Key

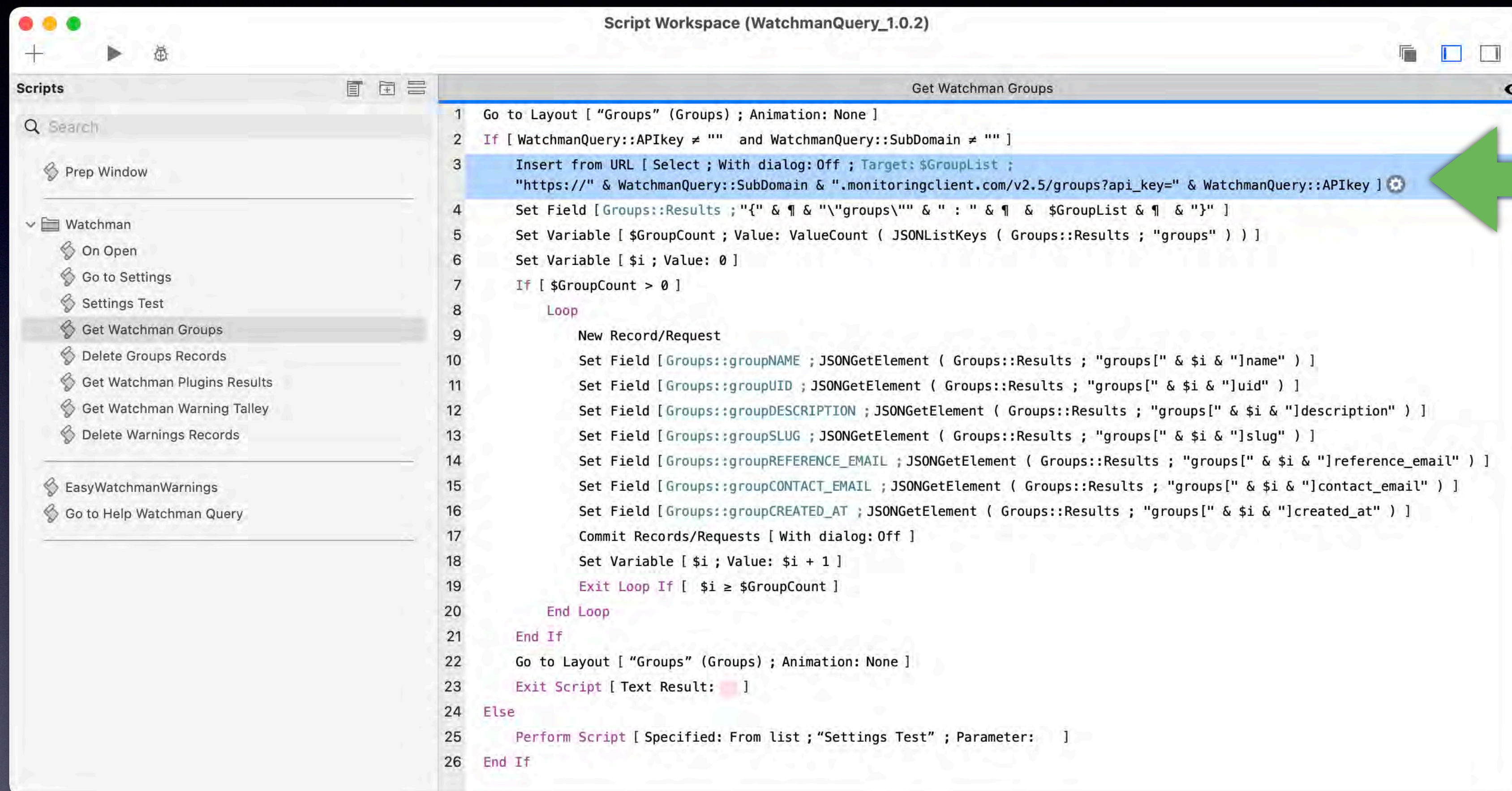
The screenshot shows a 'Script Workspace (WatchmanQuery_1.0.2)' window. On the left is a 'Scripts' sidebar with a search bar and a list of scripts under a 'Watchman' folder, including 'Get Watchman Groups'. The main area displays the script code for 'Get Watchman Groups'.

```
1 Go to Layout [ "Groups" (Groups) ; Animation: None ]
2 If [ WatchmanQuery::APIKey ≠ "" and WatchmanQuery::SubDomain ≠ "" ]
3   Insert from URL [ Select ; With dialog: Off ; Target: $GroupList ;
4     "https://" & WatchmanQuery::SubDomain & ".monitoringclient.com/v2.5/groups?api_key=" & WatchmanQuery::APIKey ]
5   Set Field [ Groups::Results ; "{" & ¶ & "\"groups\"& " : " & ¶ & $GroupList & ¶ & "]" ]
6   Set Variable [ $GroupCount ; Value: ValueCount ( JSONListKeys ( Groups::Results ; "groups" ) ) ]
7   Set Variable [ $i ; Value: 0 ]
8   If [ $GroupCount > 0 ]
9     Loop
10      New Record/Request
11      Set Field [ Groups::groupNAME ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]"name" ) ]
12      Set Field [ Groups::groupUID ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]"uid" ) ]
13      Set Field [ Groups::groupDESCRIPTION ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]"description" ) ]
14      Set Field [ Groups::groupSLUG ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]"slug" ) ]
15      Set Field [ Groups::groupREFERENCE_EMAIL ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]"reference_email" ) ]
16      Set Field [ Groups::groupCONTACT_EMAIL ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]"contact_email" ) ]
17      Set Field [ Groups::groupCREATED_AT ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]"created_at" ) ]
18      Commit Records/Requests [ With dialog: Off ]
19      Set Variable [ $i ; Value: $i + 1 ]
20    Exit Loop If [ $i ≥ $GroupCount ]
21  End Loop
22 End If
23 Go to Layout [ "Groups" (Groups) ; Animation: None ]
24 Exit Script [ Text Result: ]
25 Else
26   Perform Script [ Specified: From list ; "Settings Test" ; Parameter: ]
27 End If
```

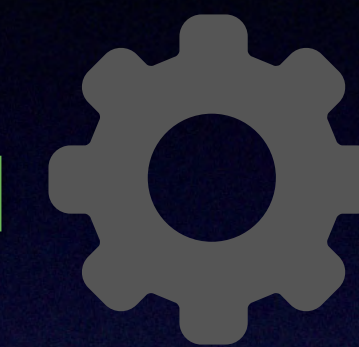
precursor.ca/slides



https://subdomain.monitoringclient.com/v2.5/groups?api_key=Example_API_Key



```
1 Go to Layout [ "Groups" (Groups) ; Animation: None ]
2 If [ WatchmanQuery::APIkey ≠ "" and WatchmanQuery::SubDomain ≠ "" ]
3   Insert from URL [ Select ; With dialog: Off ; Target: $GroupList ;
4     "https://" & WatchmanQuery::SubDomain & ".monitoringclient.com/v2.5/groups?api_key=" & WatchmanQuery::APIkey ]
5   Set Field [ Groups::Results ; "{" & ¶ & "\"groups\"& " : " & ¶ & $GroupList & ¶ & "]" ]
6   Set Variable [ $GroupCount ; Value: ValueCount ( JSONListKeys ( Groups::Results ; "groups" ) ) ]
7   Set Variable [ $i ; Value: 0 ]
8   If [ $GroupCount > 0 ]
9     Loop
10      New Record/Request
11      Set Field [ Groups::groupNAME ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]"name" ) ]
12      Set Field [ Groups::groupUID ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]"uid" ) ]
13      Set Field [ Groups::groupDESCRIPTION ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]"description" ) ]
14      Set Field [ Groups::groupSLUG ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]"slug" ) ]
15      Set Field [ Groups::groupREFERENCE_EMAIL ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]"reference_email" ) ]
16      Set Field [ Groups::groupCONTACT_EMAIL ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]"contact_email" ) ]
17      Set Field [ Groups::groupCREATED_AT ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]"created_at" ) ]
18      Commit Records/Requests [ With dialog: Off ]
19      Set Variable [ $i ; Value: $i + 1 ]
20    Exit Loop If [ $i ≥ $GroupCount ]
21  End Loop
22 End If
23 Go to Layout [ "Groups" (Groups) ; Animation: None ]
24 Exit Script [ Text Result: ]
25 Else
26 Perform Script [ Specified: From list ; "Settings Test" ; Parameter: ]
27 End If
```



precursor.ca/slides



https://subdomain.monitoringclient.com/v2.5/groups?api_key=Example_API_Key

Script Workspace (WatchmanQuery_1.0.2)

Scripts

- Prep Window
- Watchman
 - On Open
 - Go to Settings
 - Settings Test
 - Get Watchman Groups
 - Delete Groups Records
 - Get Watchman Plugins Results
 - Get Watchman Warning Talley
 - Delete Warnings Records
- EasyWatchmanWarnings
- Go to Help Watchman Query

```
1 Go to Layout [ "Groups" (Groups) ; Animation: None ]
2 If [ WatchmanQuery::APIkey ≠ "" and WatchmanQuery::SubDomain ≠ "" ]
3   Insert from URL [ Select ; With dialog: Off ; Target: $GroupList ;
4     "https://" & WatchmanQuery::SubDomain & ".monitoringclient.com/v2.5/groups?api_key=" & WatchmanQuery::APIkey ]
5   Set Field [ Groups::Results ; "{ " & ¶ & "\"groups\" \" : \" & ¶ & $GroupList & ¶ & " } " ]
6   Set Variable [ $GroupCount ; Value: ValueCount ( JSONListKeys ( Groups::Results ; "groups" ) )
7   Set Variable [ $i ; Value: 0 ]
8   If [ $GroupCount > 0 ]
9     Loop
10      New Record/Request
11      Set Field [ Groups::groupNAME ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]" ) ]
12      Set Field [ Groups::groupUID ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]" ) ]
13      Set Field [ Groups::groupDESCRIPTION ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]"description" ) ]
14      Set Field [ Groups::groupSLUG ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]"slug" ) ]
15      Set Field [ Groups::groupREFERENCE_EMAIL ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]"reference_email" ) ]
16      Set Field [ Groups::groupCONTACT_EMAIL ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]"contact_email" ) ]
17      Set Field [ Groups::groupCREATED_AT ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]"created_at" ) ]
18      Commit Records/Requests [ With dialog: Off ]
19      Set Variable [ $i ; Value: $i + 1 ]
20    Exit Loop If [ $i ≥ $GroupCount ]
21  End Loop
22 End If
23 Go to Layout [ "Groups" (Groups) ; Animation: None ]
24 Exit Script [ Text Result: ]
25 Else
26 Perform Script [ Specified: From list ; "Settings Test" ; Parameter: ]
27 End If
```

Context Menu:

- Select entire contents
- Target: Specify...
- Specify URL Specify...
- Verify SSL Certificates
- Specify cURL options Specify...

precursor.ca/slides



https://subdomain.monitoringclient.com/v2.5/groups?api_key=Example_API_Key

The screenshot shows a script workspace titled "Script Workspace (WatchmanQuery_1.0.2)". The script is named "Get Watchman Groups" and contains the following code:

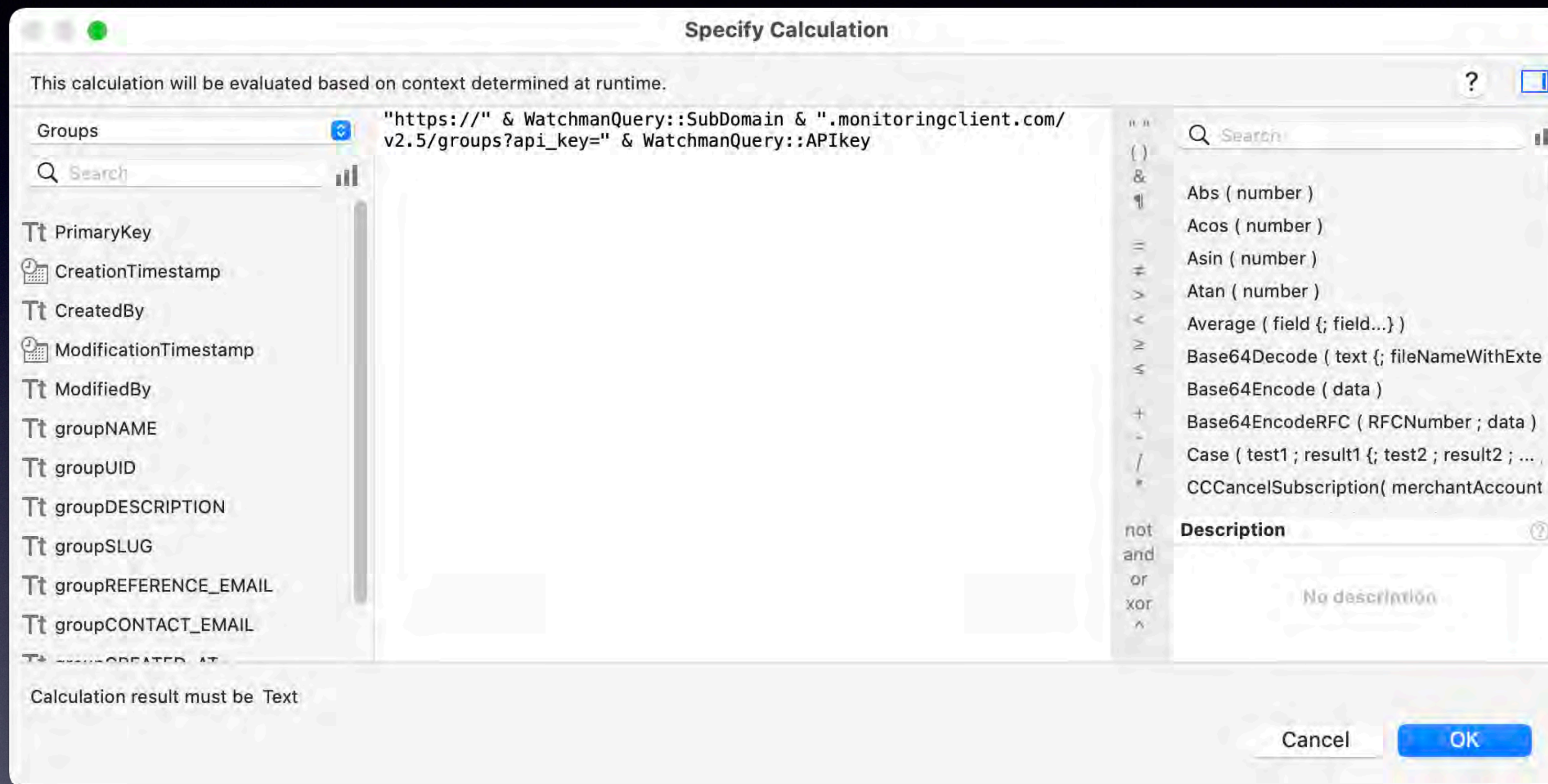
```
1 Go to Layout [ "Groups" (Groups) ; Animation: None ]
2 If [ WatchmanQuery::APIkey ≠ "" and WatchmanQuery::SubDomain ≠ "" ]
3   Insert from URL [ Select ; With dialog: Off ; Target: $GroupList ;
4     "https://" & WatchmanQuery::SubDomain & ".monitoringclient.com/v2.5/groups?api_key=" & WatchmanQuery::APIkey ]
5   Set Field [ Groups::Results ; "{" & ¶ & "\"groups\"": " & ¶ & $GroupList &
6   Set Variable [ $GroupCount ; Value: ValueCount ( JSONListKeys ( Groups::Results ;
7   If [ $GroupCount > 0 ]
8     Loop
9       New Record/Request
10      Set Field [ Groups::groupNAME ; JSONGetElement ( Groups::Results ; "groups
11      Set Field [ Groups::groupUID ; JSONGetElement ( Groups::Results ; "groups
12      Set Field [ Groups::groupDESCRIPTION ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]description" ) ]
13      Set Field [ Groups::groupSLUG ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]slug" ) ]
14      Set Field [ Groups::groupREFERENCE_EMAIL ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]reference_email" ) ]
15      Set Field [ Groups::groupCONTACT_EMAIL ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]contact_email" ) ]
16      Set Field [ Groups::groupCREATED_AT ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]created_at" ) ]
17      Commit Records/Requests [ With dialog: Off ]
18      Set Variable [ $i ; Value: $i + 1 ]
19    Exit Loop If [ $i ≥ $GroupCount ]
20  End Loop
21 End If
22 Go to Layout [ "Groups" (Groups) ; Animation: None ]
23 Exit Script [ Text Result: ]
24 Else
25   Perform Script [ Specified: From list ; "Settings Test" ; Parameter: ]
26 End If
```

An "Insert from URL" Options dialog box is open, showing the URL field with the text: "https://" & WatchmanQuery::SubDomain. A green arrow points to the "Specify..." button next to the URL field. The dialog also has a checked box for "Automatically encode URL" and buttons for "Cancel" and "OK".

precursor.ca/slides



https://subdomain.monitoringclient.com/v2.5/groups?api_key=Example_API_Key



precursor.ca/slides



https://subdomain.monitoringclient.com/v2.5/groups?api_key=Example_API_Key

Specify Calculation

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

Groups

`"https://" & WatchmanQuery::SubDomain & ".monitoringclient.com/v2.5/groups?api_key=" & WatchmanQuery::APIKey`

Search

precursor.ca/slides



Authentication in Header

precursor.ca/slides



```
GET curl https://[YOUR_DOMAIN]/api/v1/companies? -H "x-api-key: YOUR_API_KEY_HERE"
```

precursor.ca/slides



```
curl https://[YOUR_DOMAIN]/api/v1/companies? -H "x-api-key: YOUR_API_KEY_HERE"
```



The screenshot shows the CocoaRestClient interface. The URL bar contains `https://hudu.precursor.ca/api/v1/companies?`. The Method is set to GET. The Headers tab is active, showing a table with the following content:

Header Name	Header Value
Content-Type	application/x-www-form-urlencoded
x-api-key	ptG axx

The RESPONSE tab is also active, showing a JSON array of company objects. The first object is expanded, showing the following fields:

```
{
  "id": 3,
  "slug": "s-209e92665694",
  "name": "s",
  "nickname": "",
  "address_line_1": " Ave.",
  "address_line_2": "",
  "city": "Winnipeg",
  "state": "MB",
  "zip": "R3",
  "country_name": "CA",
}
```

At the bottom of the window, it says "Finished in 0.513039 seconds".

precursor.ca/slides



```
curl https://[YOUR_DOMAIN]/api/v1/companies? -H "x-api-key: YOUR_API_KEY_HERE"
```

URL: `https://hudu.precursor.ca/api/v1/companies?` Method: `GET` Submit

Body Headers Auth Files Params

Header Name	Header Value
Content-Type	application/x-www-form-urlencoded
x-api-key	ptGaxx

RESPONSE

Body Headers (200) Sent Headers

1
2
3
4
5
6
7
8
9
10
11
12
13
14

Finished in 0.513039 seconds

precursor.ca/slides



```
curl https://[YOUR_DOMAIN]/api/v1/companies? -H "x-api-key: YOUR_API_KEY_HERE"
```



Insert From URL script step

precursor.ca/slides



`https://[YOUR_DOMAIN]/api/v1/companies? -H "x-api-key: YOUR_API_KEY_HERE"`

The screenshot shows a script workspace titled "Script Workspace (HuduQuery_1.0.0)". On the left, there is a "Scripts" sidebar with a search bar and a list of scripts, including "Get Hudu Companies" which is selected. The main area displays the script code for "Get Hudu Companies". The code is as follows:

```
1 Go to Layout [ "Organizations" (Organizations) ; Animation: None ]
2 If [ HuduQuery::APIKey ≠ "" and HuduQuery::Domain ≠ "" ]
3   Insert from URL [ Select ; With dialog: Off ; Target: $OrganizationList ; "https://" & HuduQuery::Domain & "/api/v1/companies" ;
4     cURL options: //The cURL details "-X GET" & //The cURL details //Error reporting "--trace $$Trace" & "--show-error $$Er... ]
5   Set Field [ Organizations::Results ; $OrganizationList ]
6   Set Variable [ $OrganizationCount ; Value: ValueCount ( JSONListKeys ( Organizations::Results ; "companies" ) ) ]
7   Set Variable [ $i ; Value: 0 ]
8   If [ $OrganizationCount > 0 ]
9     Loop
10      New Record/Request
11      Set Field [ Organizations::organizationNAME ;
12        JSONGetElement ( Organizations::Results ; "companies[" & $i & "]name" ) ]
13      Set Field [ Organizations::organizationID ; JSONGetElement ( Organizations::Results ; "companies[" & $i & "]id" ) ]
14      Set Field [ Organizations::organizationDESCRIPTION ;
15        JSONGetElement ( Organizations::Results ; "organizations[" & $i & "]details" ) ]
16      Set Field [ Organizations::organizationURL ;
17        JSONGetElement ( Organizations::Results ; "companies[" & $i & "]full_url" ) ]
18      Set Field [ Organizations::organizationDOMAINS ;
19        JSONGetElement ( Organizations::Results ; "organizations[" & $i & "]domain_names" ) ]
20      Set Field [ Organizations::organizationNOTES ;
21        JSONGetElement ( Organizations::Results ; "organizations[" & $i & "]notes" ) ]
22      Set Field [ Organizations::organizationNICKNAME ;
23        JSONGetElement ( Organizations::Results ; "companies[" & $i & "]nickname" ) ]
24      Commit Records/Requests [ With dialog: Off ]
25      Set Variable [ $i ; Value: $i + 1 ]
26      Exit Loop If [ $i ≥ $OrganizationCount ]
27    End Loop
28  End If
29  Exit Script [ Text Result: ]
```

A yellow arrow points to the URL in line 3 of the script.

precursor.ca/slides



`https://[YOUR_DOMAIN]/api/v1/companies? -H "x-api-key: YOUR_API_KEY_HERE"`

The screenshot shows a script workspace titled "Script Workspace (HuduQuery_1.0.0)". On the left, a "Scripts" sidebar lists various scripts, with "Get Hudu Companies" selected under a "Hudu" folder. The main editor displays a script with the following steps:

- 1 Go to Layout ["Organizations" (Organizations) ; Animation: None]
- 2 If [HuduQuery::APIKey ≠ "" and HuduQuery::Domain ≠ ""]
- 3 Insert from URL [Select ; With dialog: Off ; Target: \$OrganizationList ; "https://" & HuduQuery::Domain & "/api/v1/companies" ; cURL options: //The cURL details "-X GET" & //The cURL details //Error reporting " --trace \$\$Trace" & " --show-error \$\$Er...]
- 4 Set Field [Organizations::Results ; \$OrganizationList]
- 5 Set Variable [\$OrganizationCount ; Value: ValueCount (JSONListKeys (Organizations::Results ; "companies"))]
- 6 Set Variable [\$i ; Value: 0]
- 7 If [\$OrganizationCount > 0]
- 8 Loop
- 9 New Record/Request
- 10 Set Field [Organizations::organizationNAME ; JSONGetElement (Organizations::Results ; "companies[" & \$i & "]"name")]
- 11 Set Field [Organizations::organizationID ; JSONGetElement (Organizations::Results ; "companies[" & \$i & "]"id")]
- 12 Set Field [Organizations::organizationDESCRIPTION ; JSONGetElement (Organizations::Results ; "organizations[" & \$i & "]"details")]
- 13 Set Field [Organizations::organizationURL ; JSONGetElement (Organizations::Results ; "companies[" & \$i & "]"full_url")]
- 14 Set Field [Organizations::organizationDOMAINS ; JSONGetElement (Organizations::Results ; "organizations[" & \$i & "]"domain_names")]
- 15 Set Field [Organizations::organizationNOTES ; JSONGetElement (Organizations::Results ; "organizations[" & \$i & "]"notes")]
- 16 Set Field [Organizations::organizationNICKNAME ; JSONGetElement (Organizations::Results ; "companies[" & \$i & "]"nickname")]
- 17 Commit Records/Requests [With dialog: Off]
- 18 Set Variable [\$i ; Value: \$i + 1]
- 19 Exit Loop If [\$i ≥ \$OrganizationCount]
- 20 End Loop
- 21 End If
- 22 Exit Script [Text Result:]

A context menu is open over the "Insert from URL" step, with a yellow arrow pointing to it. The menu options are:

- Select entire contents
- Target: Specify...
- Specify URL Specify...
- Verify SSL Certificates
- Specify cURL options Specify...

precursor.ca/slides



`https://[YOUR_DOMAIN]/api/v1/companies? -H "x-api-key: YOUR_API_KEY_HERE"`

The screenshot shows a script workspace titled "Script Workspace (HuduQuery_1.0.0)". The main editor displays a script for "Get Hudu Companies" with the following code:

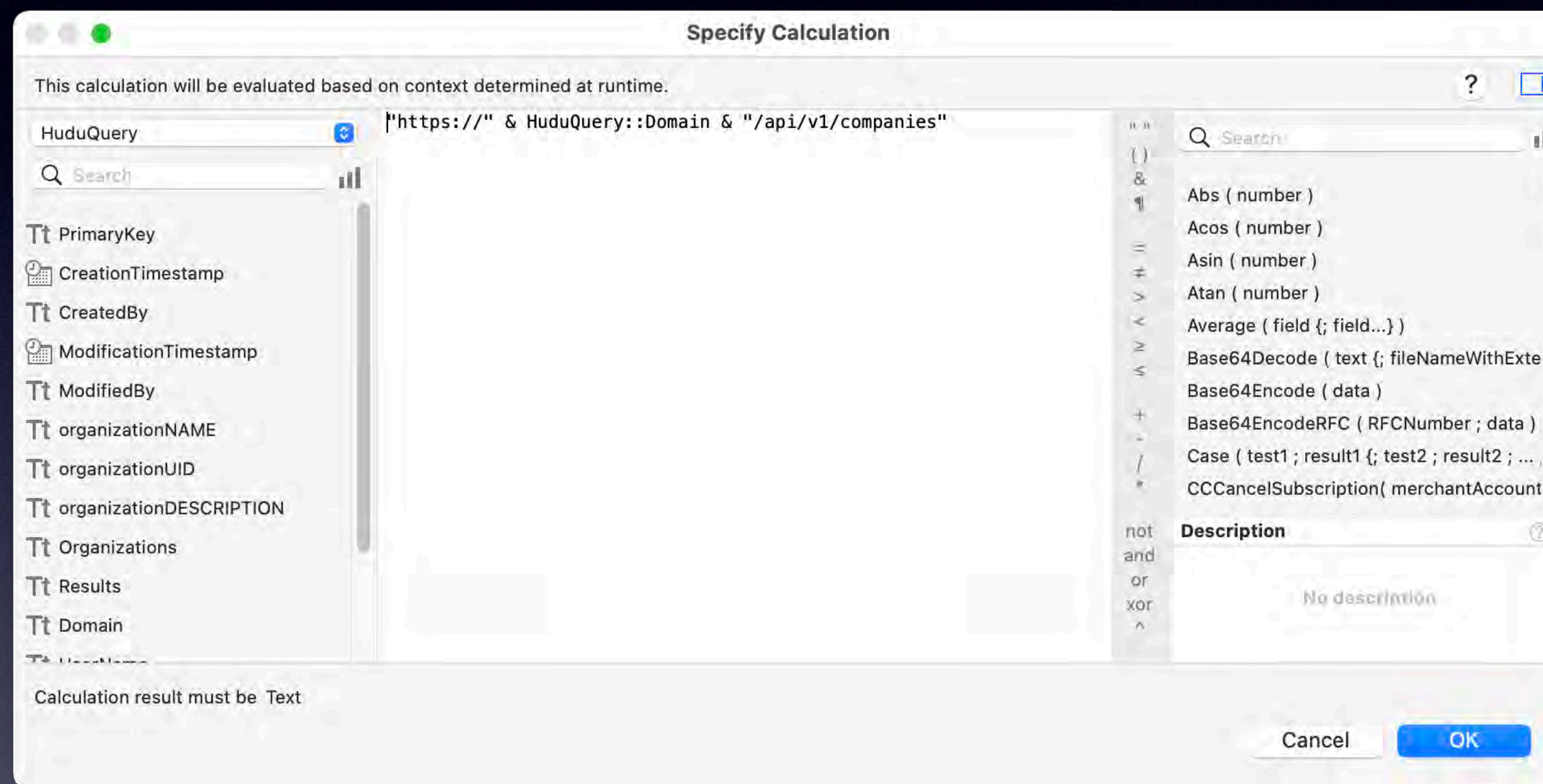
```
1 Go to Layout [ "Organizations" (Organizations) ; Animation: None ]
2 If [ HuduQuery::APIKey ≠ "" and HuduQuery::Domain ≠ "" ]
3   Insert from URL [ Select ; With dialog: Off ; Target: $OrganizationList ; "https://" & HuduQuery::Domain & "/api/v1/companies" ;
4     cURL options: //The cURL details "-X GET" & //The cURL details //Error reporting "--trace $$Trace" & "--show-error $$Er... ]
5   Set Field [ Organizations::Results ; $OrganizationList ]
6   Set Variable [ $OrganizationCount ; Value: ValueCount ( JSONListKeys ( Organizations::Results ; "
7   Set Variable [ $i ; Value: 0 ]
8   If [ $OrganizationCount > 0 ]
9     Loop
10      New Record/Request
11      Set Field [ Organizations::organizationNAME ;
12        JSONGetElement ( Organizations::Results ; "companies[" & $i & "]"name" ) ]
13      Set Field [ Organizations::organizationID ; JSONGetElement ( Organizations::Results ; "companies[" & $i & "]"id" ) ]
14      Set Field [ Organizations::organizationDESCRIPTION ;
15        JSONGetElement ( Organizations::Results ; "organizations[" & $i & "]"details" ) ]
16      Set Field [ Organizations::organizationURL ;
17        JSONGetElement ( Organizations::Results ; "companies[" & $i & "]"full_url" ) ]
18      Set Field [ Organizations::organizationDOMAINS ;
19        JSONGetElement ( Organizations::Results ; "organizations[" & $i & "]"domain_names" ) ]
20      Set Field [ Organizations::organizationNOTES ;
21        JSONGetElement ( Organizations::Results ; "organizations[" & $i & "]"notes" ) ]
22      Set Field [ Organizations::organizationNICKNAME ;
23        JSONGetElement ( Organizations::Results ; "companies[" & $i & "]"nickname" ) ]
24      Commit Records/Requests [ With dialog: Off ]
25      Set Variable [ $i ; Value: $i + 1 ]
26      Exit Loop If [ $i ≥ $OrganizationCount ]
27    End Loop
28  End If
29  Exit Script [ Text Result: ]
```

An "Insert from URL" Options dialog box is open over the script, showing the URL field with the text `"https://" & HuduQuery::Domain & "/api/`. A yellow arrow points to the "Specify..." button next to the URL field. The dialog also has a checked "Automatically encode URL" checkbox and "Cancel" and "OK" buttons.

precursor.ca/slides



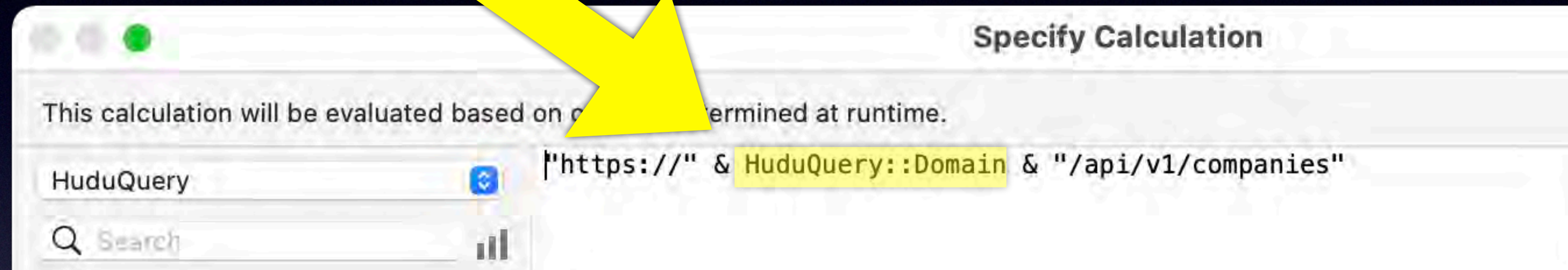
`https://[YOUR_DOMAIN]/api/v1/companies? -H "x-api-key: YOUR_API_KEY_HERE"`



precursor.ca/slides



```
https://[YOUR_DOMAIN]/api/v1/companies? -H "x-api-key: YOUR_API_KEY_HERE"
```



precursor.ca/slides



`https://[YOUR_DOMAIN]/api/v1/companies? -H "x-api-key: YOUR_API_KEY_HERE"`

Script Workspace (HuduQuery_1.0.0)

Get Hudu Companies

```
1 Go to Layout [ "Organizations" (Organizations) ; Animation: None ]
2 If [ HuduQuery::APIKey ≠ "" and HuduQuery::Domain ≠ "" ]
3 Insert from URL [ Select ; With dialog: Off ; Target: $OrganizationList ; "https://" & HuduQuery::Domain & "/api/v1/companies" ;
  cURL options: //The cURL details "-X GET" & //The cURL details //Error reporting "--trace $$Trace" & "--show-error $$Er... ]
4 Set Field [ Organizations::Results ; $OrganizationList ]
5 Set Variable [ $OrganizationCount ; Value: ValueCount ( JSONListKeys ( Organizations::Results ; "companies" ) ) ]
6 Set Variable [ $i ; Value: 0 ]
7 If [ $OrganizationCount > 0 ]
8 Loop
9 New Record/Request
10 Set Field [ Organizations::organizationNAME ;
  JSONGetElement ( Organizations::Results ; "companies[" & $i & "]"name" ) ]
11 Set Field [ Organizations::organizationID ; JSONGetElement ( Organizations::Results ; "companies[" & $i & "]"id" ) ]
12 Set Field [ Organizations::organizationDESCRIPTION ;
  JSONGetElement ( Organizations::Results ; "organizations[" & $i & "]"details" ) ]
13 Set Field [ Organizations::organizationURL ;
  JSONGetElement ( Organizations::Results ; "companies[" & $i & "]"full_url" ) ]
14 Set Field [ Organizations::organizationDOMAINS ;
  JSONGetElement ( Organizations::Results ; "organizations[" & $i & "]"domain_names" ) ]
15 Set Field [ Organizations::organizationNOTES ;
  JSONGetElement ( Organizations::Results ; "organizations[" & $i & "]"notes" ) ]
16 Set Field [ Organizations::organizationNICKNAME ;
  JSONGetElement ( Organizations::Results ; "companies[" & $i & "]"nickname" ) ]
17 Commit Records/Requests [ With dialog: Off ]
18 Set Variable [ $i ; Value: $i + 1 ]
19 Exit Loop If [ $i ≥ $OrganizationCount ]
20 End Loop
21 End If
22 Exit Script [ Text Result: ]
```

Scripts

- Prep Window
- On Open
- Settings Test
- EasyHudu365Apps
- Hudu
 - Get Hudu Companies
 - Delete Hudu Organizations
 - Get Hudu 365 Apps
 - Sort Hudu 365 for only 365
 - Sort Hudu 365 remove EXT accounts
 - Sort Hudu 365 Total
 - Delete Hudu Records
 - Go to Hudu Settings
- Go to Help Hudu Query

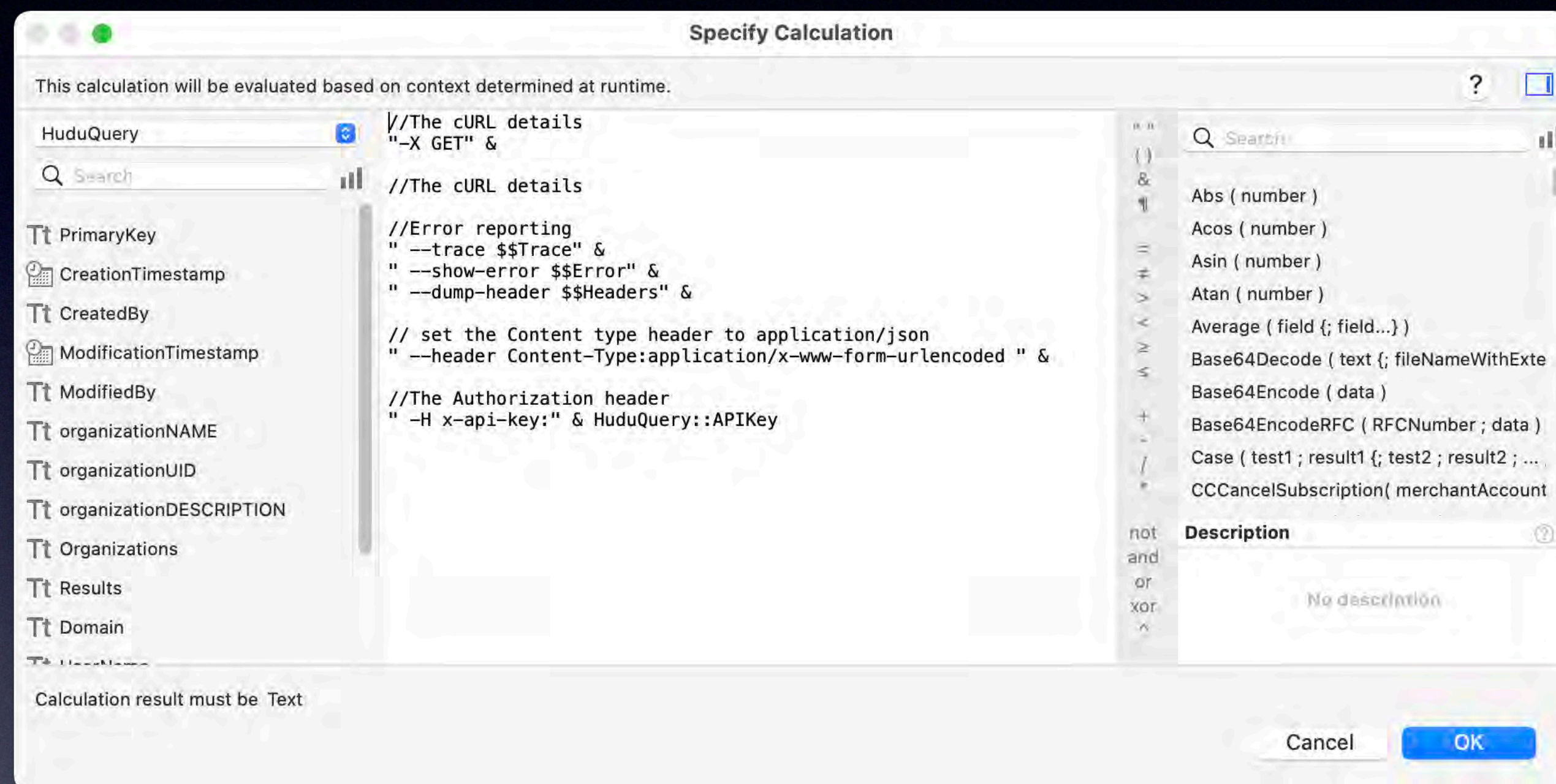
Context Menu:

- Select entire contents
- Target: Specify...
- Specify URL Specify...
- Verify SSL Certificates
- Specify cURL options Specify...

precursor.ca/slides



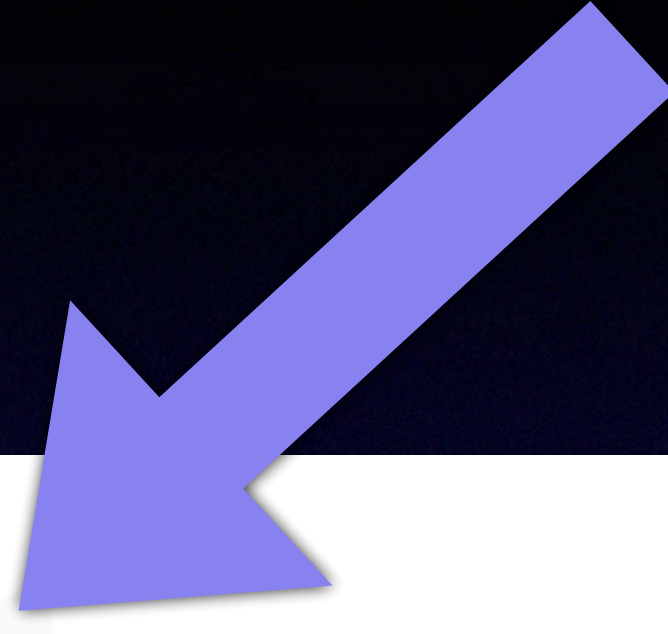

`https://[YOUR_DOMAIN]/api/v1/companies? -H "x-api-key: YOUR_API_KEY_HERE"`



precursor.ca/slides



```
https://[YOUR_DOMAIN]/api/v1/companies? -H "x-api-key: YOUR_API_KEY_HERE"
```



```
//The Authorization header  
" -H x-api-key:" & HuduQuery::APIKey
```

precursor.ca/slides



Data in URL

precursor.ca/slides



POST

```
curl https://your_subdomain.monitoringclient.com/v2.5/expirations
```

precursor.ca/slides



POST

```
curl https://your_subdomain.monitoringclient.com/v2.5/expirations \
-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
```

precursor.ca/slides



POST

```
curl https://your_subdomain.monitoringclient.com/v2.5/expirations \
-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
```

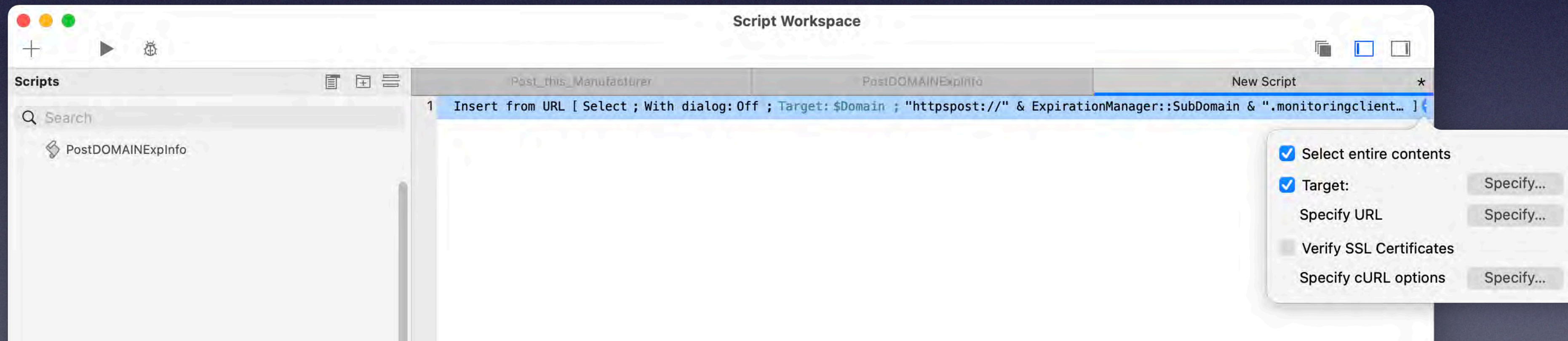
URL: Method:

precursor.ca/slides



POST

```
curl https://your_subdomain.monitoringclient.com/v2.5/expirations \
-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
```

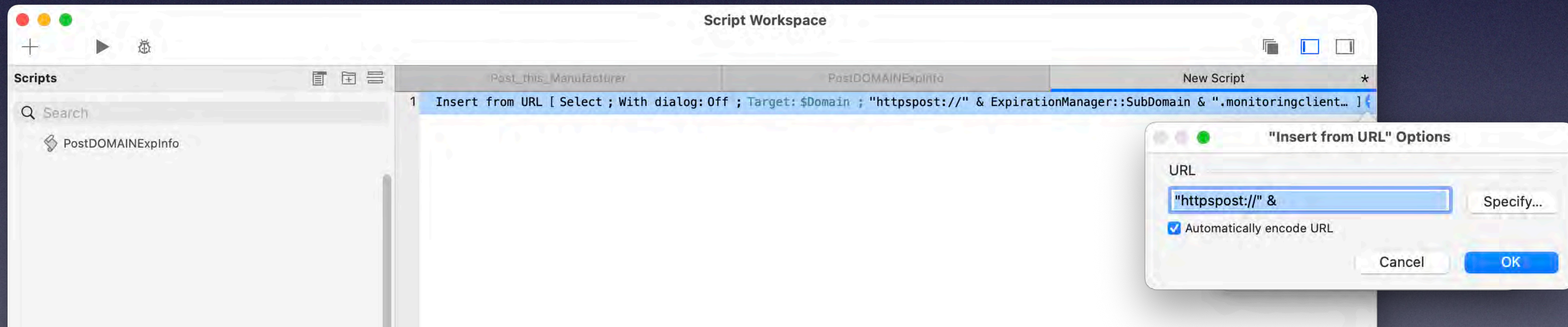


precursor.ca/slides



POST

```
curl https://your_subdomain.monitoringclient.com/v2.5/expirations \
-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
```



precursor.ca/slides



POST

```
curl https://your_subdomain.monitoringclient.com/v2.5/expirations \
-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
```



precursor.ca/slides



`"text" & table::fielddata & $variable & "&key=data"`

precursor.ca/slides



FileMaker Ampersand

`“text” & table::fielddata & $variable & “&key=data”`

precursor.ca/slides



URL Ampersand

`“text” & table::fielddata & $variable & “&key=data”`

precursor.ca/slides



Data sent by JSON

precursor.ca/slides



POST curl **http://businessapi.mosyle.com/v1/devices**

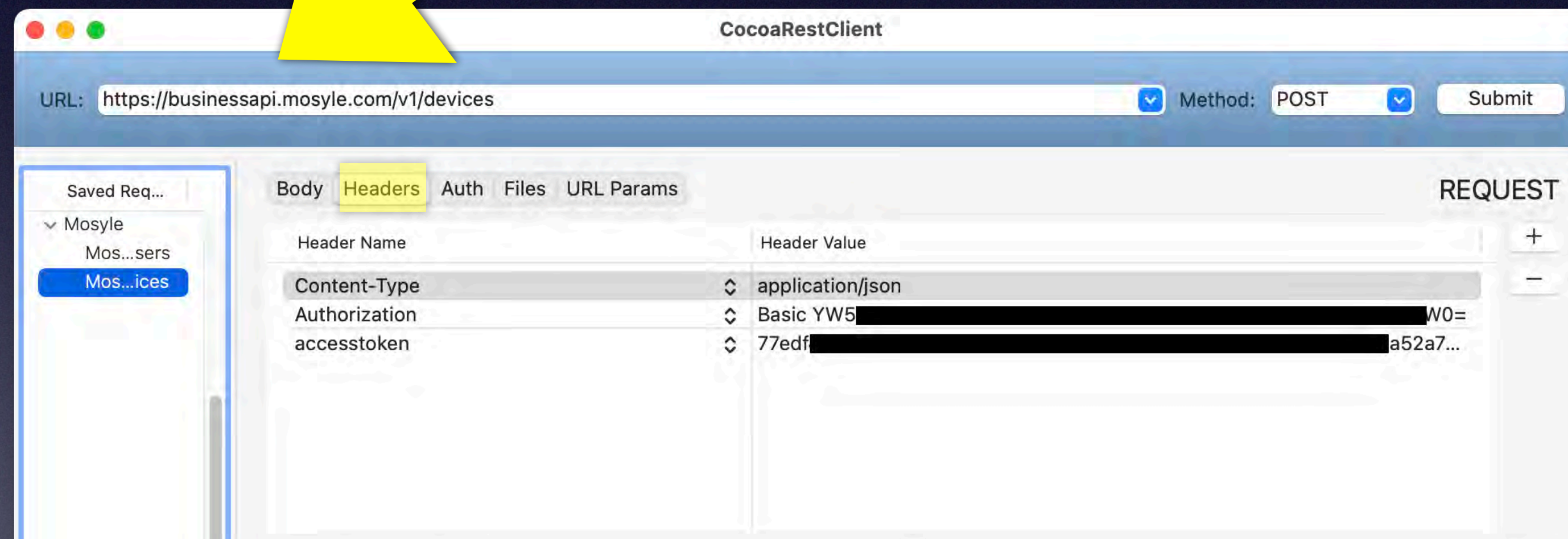
```
{  
  "operation": "list",  
  "options" : { "os": "mac"}  
}
```

precursor.ca/slides



POST curl **http://businessapi.mosyle.com/v1/devices**

```
{  
  "operation": "list",  
  "options": { "os": "mac" }
```

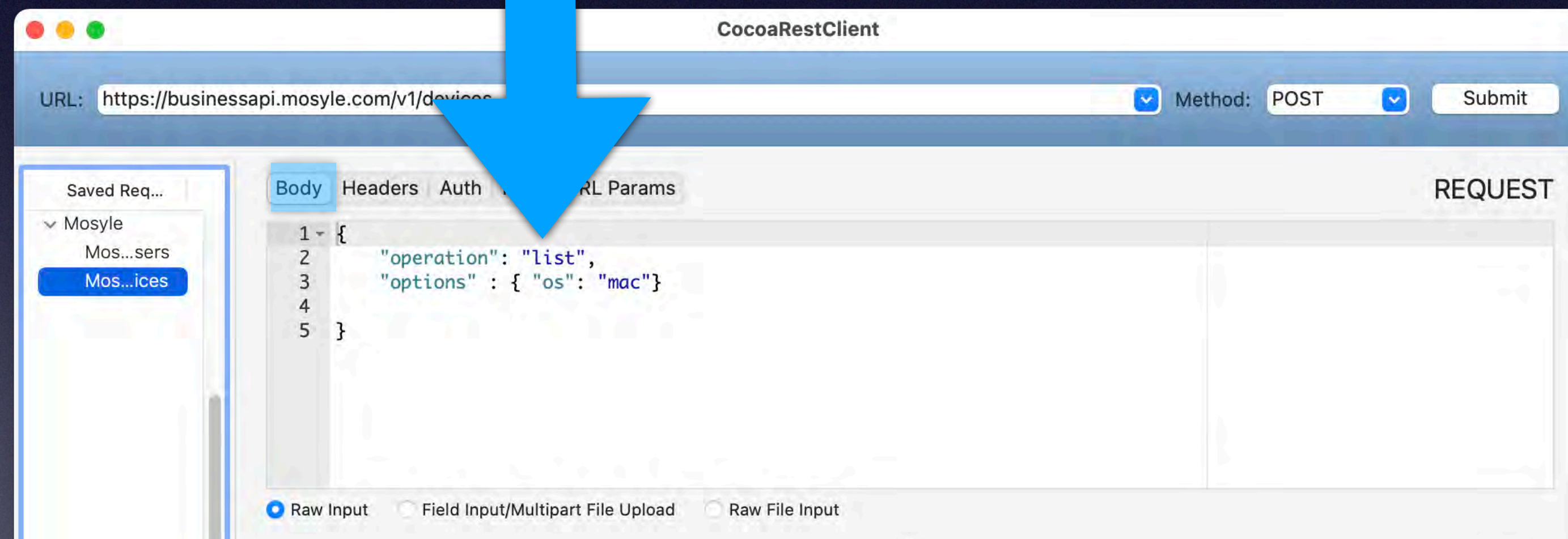


precursor.ca/slides



POST curl **http://businessapi.mosyle.com/v1/devices**

```
{  
  "operation": "list",  
  "options" : { "os": "mac"}  
}
```



precursor.ca/slides



POST curl **http://businessapi.mosyle.com/v1/devices**

```
{  
  "operation": "list",  
  "options" : { "os": "mac"}  
}
```

```
Script Workspace (MosyleQuery_2.0)  
Get Mosyle Devices  
1 If [ MosyleQuery::APIKey ≠ "" and MosyleQuery::SubDomain ≠ "" ]  
2   Set Variable [ $data ; Value: "{ \"operation\": \"list\", \"options\": { \"os\": \"mac\" } }" ]  
3   Insert from URL [ Select ; With dialog: Off ; Target: $DeviceList ; "https://businessapi.mosyle.com/v1/devices" ; cURL options:  
  //The cURL details "-X POST" & //The cURL details "-d @$data" & //Error reporting //" --trace $$Trace" & //" --show-erro... ]  
4   Set Variable [ $DeviceResultTrim ; Value: Right ( $DeviceList ; Length ( $DeviceList ) -27 ) ]  
5   Set Field [ Devices::Results ; Left ( $DeviceResultTrim ; Length ( $DeviceResultTrim ) - 2 ) ]  
6   Set Variable [ $DeviceCount ; Value: ValueCount ( JSONListKeys ( Devices::Results ; "devices" ) ) ]  
7   Set Variable [ $i ; Value: 0 ]  
8   If [ $DeviceCount > 0 ]  
9     Loop  
10      New Record/Request  
11      Set Field [ Devices::deviceudid ; JSONGetElement ( Devices::Results ; "devices[" & $i & "]deviceudid" ) ]  
12      Set Field [ Devices::total_disk ; JSONGetElement ( Devices::Results ; "devices[" & $i & "]total_disk" ) ]  
13      Set Field [ Devices::userCODE ; JSONGetElement ( Devices::Results ; "devices[" & $i & "]userCODE" ) ]  
14      Set Field [ Devices::os ; JSONGetElement ( Devices::Results ; "devices[" & $i & "]os" ) ]
```

precursor.ca/slides



POST curl **http://businessapi.mosyle.com/v1/devices**

```
{  
  "operation": "list",  
  "options" : { "os": "mac"}  
}
```

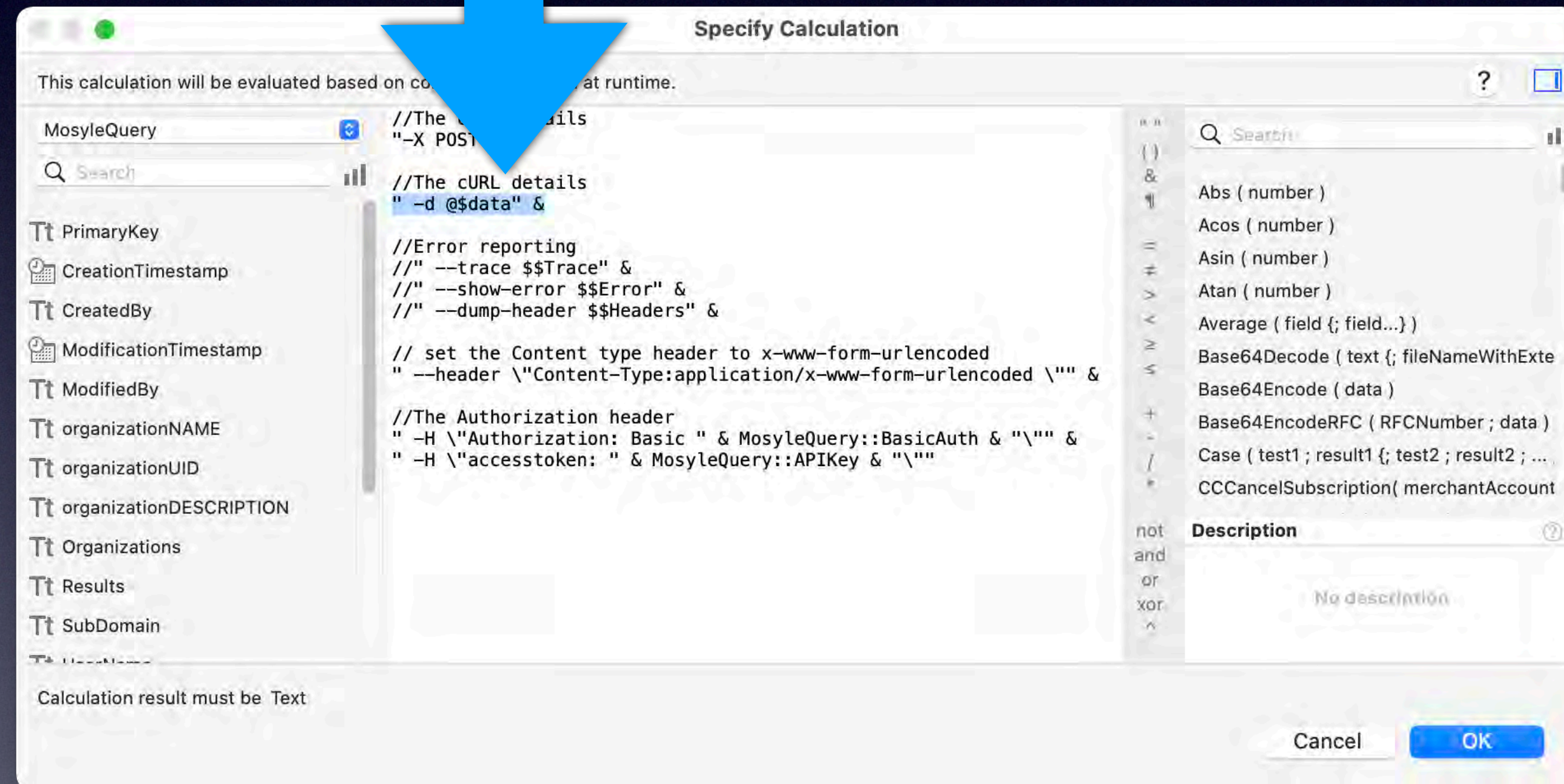
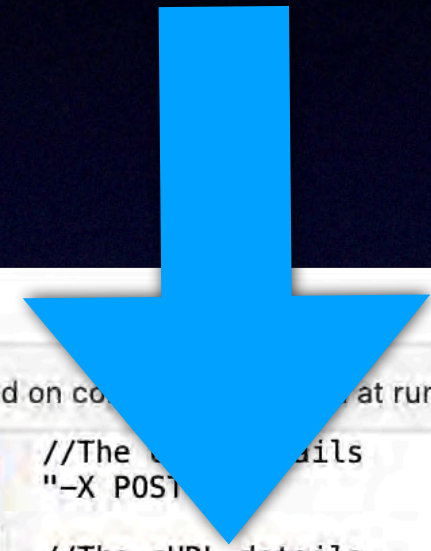


precursor.ca/slides



POST curl **http://businessapi.mosyle.com/v1/devices**

```
{  
  "operation": "list",  
  "options" : { "os": "mac"}  
}
```

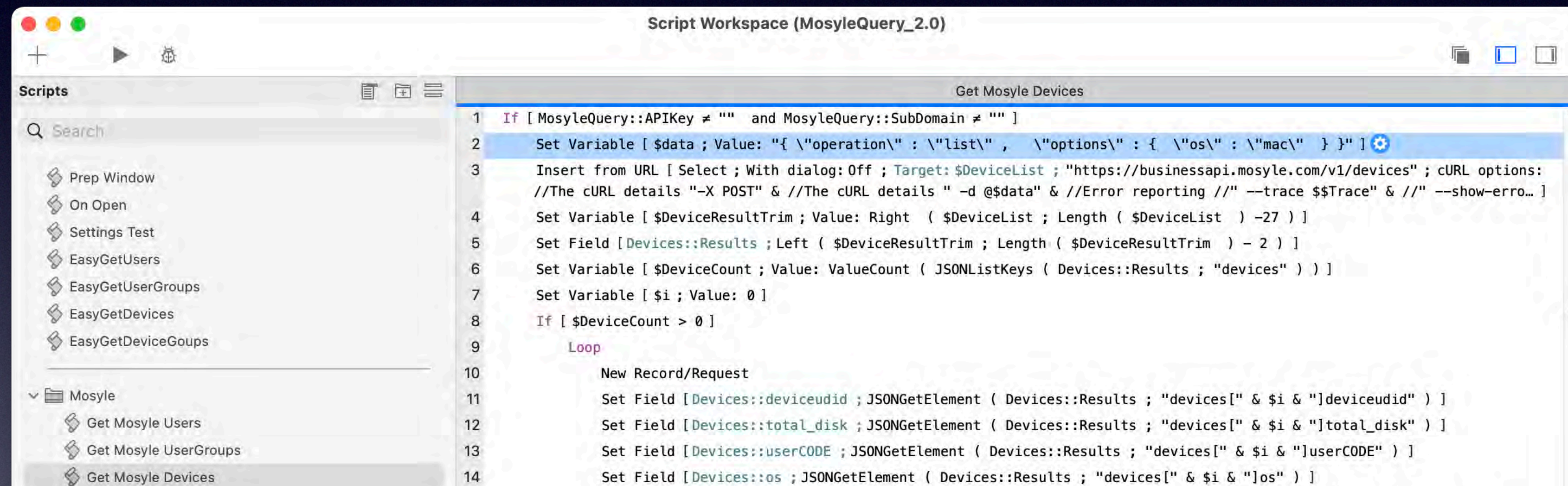


precursor.ca/slides



POST curl **http://businessapi.mosyle.com/v1/devices**

```
{  
  "operation": "list",  
  "options" : { "os": "mac"}  
}
```



precursor.ca/slides



POST curl <http://businessapi.mosyle.com/v1/devices>

```
{  
  "operation": "list",  
  "options" : { "os": "mac"}  
}
```

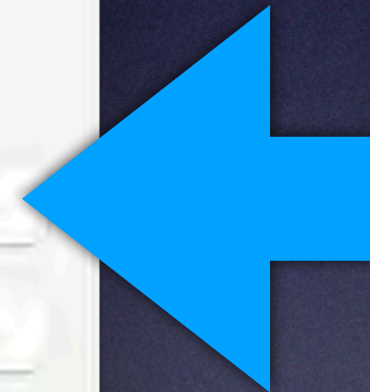
"Set Variable" Options

Names prefixed by "\$" are local variables available only within the current script. Prefix the name with "\$\$" to make the variable available throughout the current file (global).

Name:

Value: Specify...

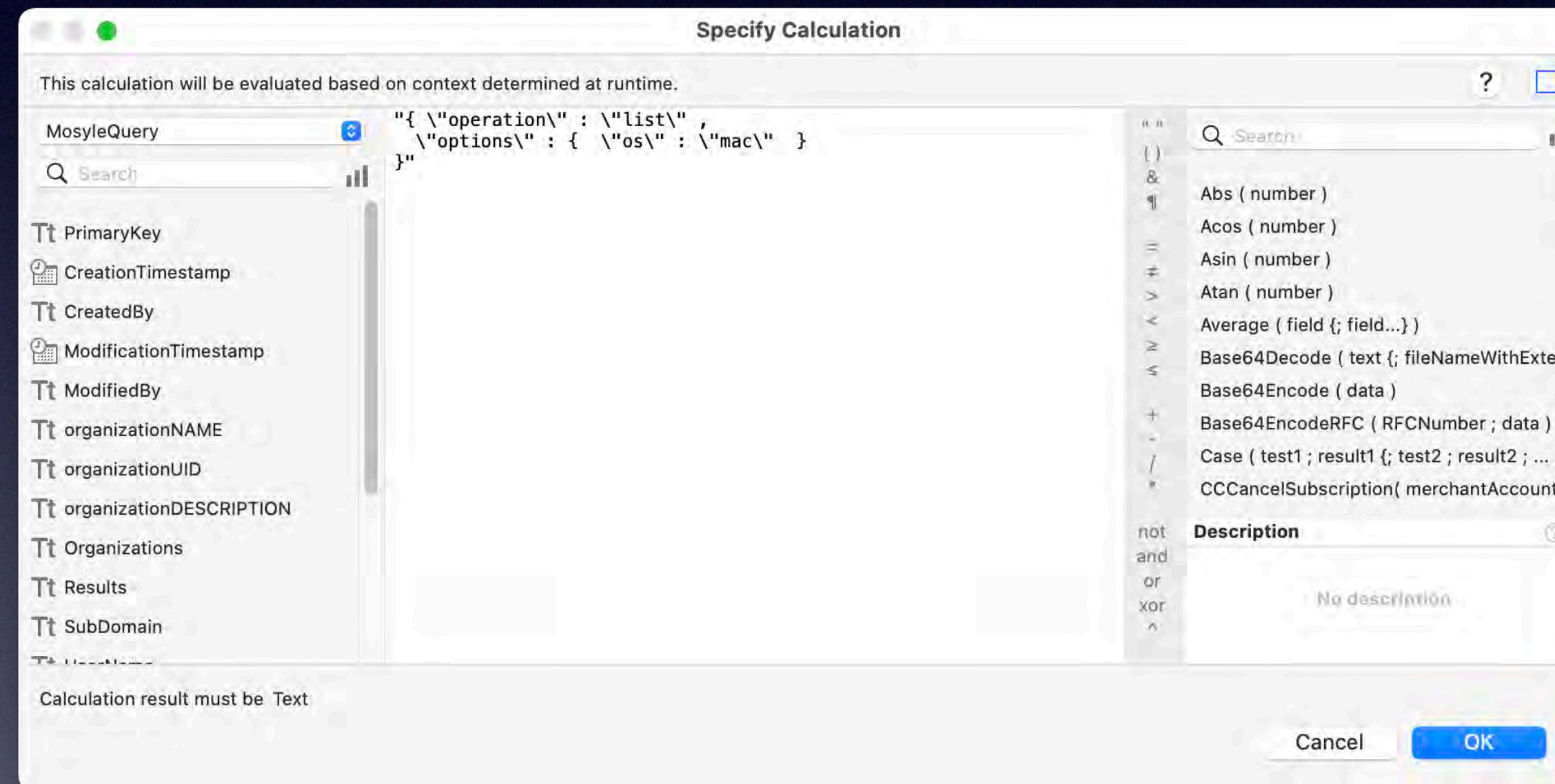
Repetition: Specify...



precursor.ca/slides



```
POST curl http://businessapi.mosyle.com/v1/devices
{
  "operation": "list",
  "options" : { "os": "mac"}
}
```

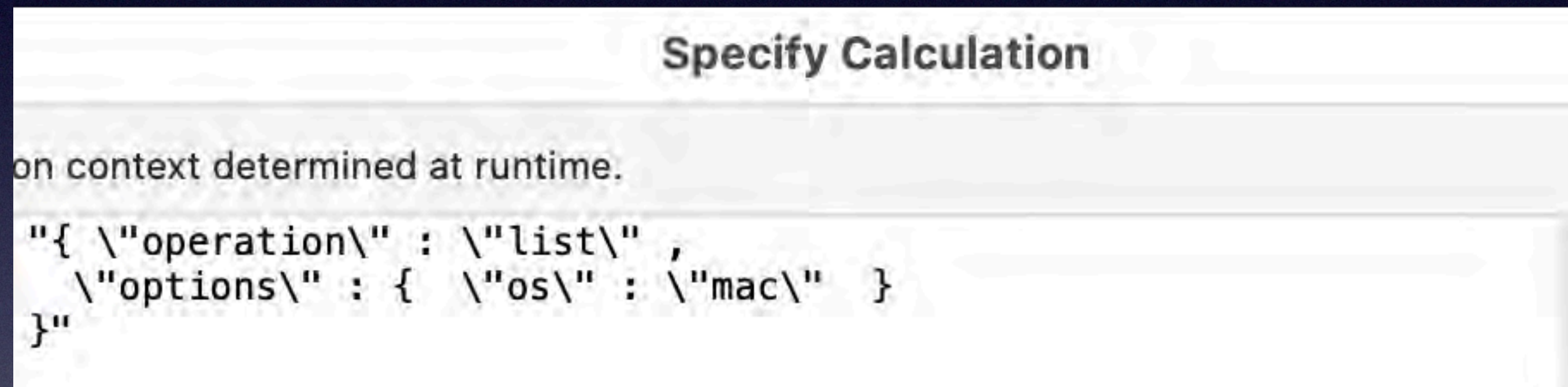


precursor.ca/slides



POST curl **http://businessapi.mosyle.com/v1/devices**

```
{  
  "operation": "list",  
  "options" : { "os": "mac"}  
}
```



precursor.ca/slides



- H / --header <name: value>** Use this option multiple times to set multiple header entries.
- F / --form <name=content>** // form data (quoted?)
- F / --form <name=@[\$]fmvariable>** // form data value from variable
- b / --cookie-jar <[\$]fmvariable>** Direct file access replaced with FM variables.
- d / --data <data>** Data (quoted)
- d / --data @<[\$]fmvariable>** Data from FileMaker variable
(@ character prefixing a FM variable replaces direct file access)



JSON Parsing

FileMaker's JSON Parsing tools

precursor.ca/slides



JSON functions

JSON functions parse, create, modify, and format data in JSON format. See [Working with the JSON functions](#).

This function	Returns
JSONDeleteElement	Deletes a JSON data element specified by an object name, an array index, or a path.
JSONFormatElements	Formats elements in JSON data to make them easier to read.
JSONGetElement	Queries JSON data for an element specified by an object name, an array index, or a path.
JSONListKeys	Lists the object names (keys) or array indexes in JSON data for an element specified by an object name, an array index, or a path.
JSONListValues	Lists the values in JSON data for an element specified by an object name, an array index, or a path.
JSONSetElement	Adds or modifies an element in JSON data specified by an object name, an array index, or a path.



JSONFormatElements

precursor.ca/slides



```
{"firstName": "John", "lastName": "doe", "age": "26", "address": {"streetAddress": "123 Main street", "city": "Anytown", "postalCode": "12345"}, "phoneNumbers": [{"type": "iPhone", "number": "123-456-8888"}, {"type": "home", "number": "123-557-8910"}]}
```

precursor.ca/slides



JSONFormatElements

```
{
  "firstName": "John",
  "lastName" : "doe",
  "age"      : "26",
  "address"  : {
    "streetAddress": "123 Main street",
    "city"         : "Anytown",
    "postalCode"  : "12345"
  },
  "phoneNumbers": [
    {
      "type"    : "iPhone",
      "number"  : "123-456-8888"
    },
    {
      "type"    : "home",
      "number"  : "123-557-8910"
    }
  ]
}
```

precursor.ca/slides



```
{
  "firstName": "John",
  "lastName" : "doe",
  "age"      : "26",
  "address"  : {
    "streetAddress": "123 Main street",
    "city"         : "Anytown",
    "postalCode"  : "12345"
  },
  "phoneNumbers": [
    {
      "type" : "iPhone",
      "number": "123-456-8888"
    },
    {
      "type" : "home",
      "number": "123-557-8910"
    }
  ]
}
```

<https://www.soliantconsulting.com/blog/parsing-json-filemaker-16/>

precursor.ca/slides



Keys

```
{
  "firstName": "John",
  "lastName" : "doe",
  "age"      : "26",
  "address"  : {
    "streetAddress": "123 Main street",
    "city"         : "Anytown",
    "postalCode"  : "12345"
  },
  "phoneNumbers": [
    {
      "type" : "iPhone",
      "number": "123-456-8888"
    },
    {
      "type" : "home",
      "number": "123-557-8910"
    }
  ]
}
```

precursor.ca/slides



```
{  
  "firstName": "John",  
  "lastName" : "doe",  
  "age"       : "26",  
  "address"   : {  
    "streetAddress": "123 Main street",  
    "city"         : "Anytown",  
    "postalCode"  : "12345"  
  },  
  "phoneNumbers": [  
    {  
      "type"   : "iPhone",  
      "number": "123-456-8888"  
    },  
    {  
      "type"   : "home",  
      "number": "123-557-8910"  
    }  
  ]  
}
```

Values

precursor.ca/slides



Arrays

```
{
  "firstName": "John",
  "lastName" : "doe",
  "age"      : "26",
  "address"  : {
    "streetAddress": "123 Main street",
    "city"         : "Anytown",
    "postalCode"  : "12345"
  },
  "phoneNumbers": [
    {
      "type" : "iPhone",
      "number": "123-456-8888"
    },
    {
      "type" : "home",
      "number": "123-557-8910"
    }
  ]
}
```

precursor.ca/slides



```
{
  "firstName": "John",
  "lastName" : "doe",
  "age"      : "26",
  "address"  : {
    "streetAddress": "123 Main street",
    "city"         : "Anytown",
    "postalCode"  : "12345"
  },
  "phoneNumbers": [
    {
      "type" : "iPhone",
      "number": "123-456-8888"
    },
    {
      "type" : "home",
      "number": "123-557-8910"
    }
  ]
}
```

JSONGetElement(JSON ; "firstName")

"John"



```
{
  "firstName": "John",
  "lastName" : "doe",
  "age"      : "26",
  "address"  : {
    "streetAddress": "123 Main street",
    "city"         : "Anytown",
    "postalCode"  : "12345"
  },
  "phoneNumbers": [
    {
      "type" : "iPhone",
      "number": "123-456-8888"
    },
    {
      "type" : "home",
      "number": "123-557-8910"
    }
  ]
}
```

JSONGetElement(JSON ; "address")

```
{
  "streetAddress": "123 Main street",
  "city"         : "Anytown",
  "postalCode"  : "12345"
}
```




```
{
  "firstName": "John",
  "lastName" : "doe",
  "age"      : "26",
  "address"  : {
    "streetAddress": "123 Main street",
    "city"         : "Anytown",
    "postalCode"  : "12345"
  },
  "phoneNumbers": [
    {
      "type" : "iPhone",
      "number": "123-456-8888"
    },
    {
      "type" : "home",
      "number": "123-557-8910"
    }
  ]
}
```

JSONGetElement(JSON ; "address.streetAddress")

"123 Main Street"



```
{
  "firstName": "John",
  "lastName" : "doe",
  "age"      : "26",
  "address"  : {
    "streetAddress": "123 Main street",
    "city"         : "Anytown",
    "postalCode"  : "12345"
  },
  "phoneNumbers": [
    {
      "type" : "iPhone",
      "number": "123-456-8888"
    },
    {
      "type" : "home",
      "number": "123-557-8910"
    }
  ]
}
```

JSONGetElement(JSON ; "phoneNumbers")

```
[
  {
    "type" : "iPhone",
    "number": "123-456-8888"
  },
  {
    "type" : "home",
    "number": "123-557-8910"
  }
]
```




```
{
  "firstName": "John",
  "lastName" : "doe",
  "age"      : "26",
  "address"  : {
    "streetAddress": "123 Main street",
    "city"         : "Anytown",
    "postalCode"  : "12345"
  },
  "phoneNumbers": [
    {
      "type" : "iPhone",
      "number": "123-456-8888"
    },
    {
      "type" : "home",
      "number": "123-557-8910"
    }
  ]
}
```

```
JSONGetElement( JSON ; "phoneNumbers[0]" )
```

```
{"number": "123-456-8888", "type": "iPhone"}
```




```
{
  "firstName": "John",
  "lastName" : "doe",
  "age"      : "26",
  "address"  : {
    "streetAddress": "123 Main street",
    "city"         : "Anytown",
    "postalCode"  : "12345"
  },
  "phoneNumbers": [
    {
      "type" : "iPhone",
      "number": "123-456-8888"
    },
    {
      "type" : "home",
      "number": "123-557-8910"
    }
  ]
}
```

```
JSONGetElement( JSON ; "phoneNumbers[0].number" )
```

```
"123-456-8888"
```




```
{
  "firstName": "John",
  "lastName" : "doe",
  "age"      : "26",
  "address"  : {
    "streetAddress": "123 Main street",
    "city"         : "Anytown",
    "postalCode"  : "12345"
  },
  "phoneNumbers": [
    {
      "type" : "iPhone",
      "number": "123-456-8888"
    },
    {
      "type" : "home",
      "number": "123-557-8910"
    }
  ]
}
```

JSONGetElement(JSON ; "phoneNumbers[1].number")

"123-557-8910"



```
{
  "firstName": "John",
  "lastName" : "doe",
  "age"      : "26",
  "address"  : {
    "streetAddress": "123 Main street",
    "city"         : "Anytown",
    "postalCode"  : "12345"
  },
  "phoneNumbers": [
    {
      "type" : "iPhone",
      "number": "123-456-8888"
    },
    {
      "type" : "home",
      "number": "123-557-8910"
    }
  ]
}
```

```
ValueCount( JSONListKeys( JSON; "phoneNumbers" ) )
```

"2"



```
JSONGetElement( JSON ; "phoneNumbers[" & $i & "].number" )
```

precursor.ca/slides



Script Workspace (WatchmanGroups)

Scripts

Search

Get Watchman Groups *

```
1 Insert from URL [ Select ; With dialog: Off ; Target: $GroupList ;  
  "https://" & WatchmanQuery::SubDomain & ".monitoringclient.com/v2.5/groups?api_key=" & WatchmanQuery::APIkey ]  
2 Set Variable [ $GroupCount ; Value: ValueCount ( JSONListKeys ( Groups::Results ; "groups" ) ) ]  
3 Set Variable [ $i ; Value: 0 ]  
4 If [ $GroupCount > 0 ]  
5   Loop  
6     New Record/Request  
7     Set Field [ Groups::groupNAME ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]"name" ) ]  
8     Set Field [ Groups::groupUID ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]"uid" ) ]  
9     Commit Records/Requests [ With dialog: Off ]  
10    Exit Loop If [ $i ≥ $GroupCount ]  
11    Set Variable [ $i ; Value: $i + 1 ]  
12  End Loop  
13 End If
```




Script Workspace (WatchmanGroups)

Scripts

Search

Get Watchman Groups *

```
1 Insert from URL [ Select ; With dialog: Off ; Target: $GroupList ;  
  "https://" & WatchmanQuery::SubDomain & ".monitoringclient.com/v2.5/groups?api_key=" & WatchmanQuery::APIkey ]  
2 Set Variable [ $GroupCount ; Value: ValueCount ( JSONListKeys ( Groups::Results ; "groups" ) ) ]  
3 Set Variable [ $i ; Value: 0 ]  
4 If [ $GroupCount > 0 ]  
5   Loop  
6     New Record/Request  
7     Set Field [ Groups::groupNAME ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]"name" ) ]  
8     Set Field [ Groups::groupUID ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]"uid" ) ]  
9     Commit Records/Requests [ With dialog: Off ]  
10    Exit Loop If [ $i ≥ $GroupCount ]  
11    Set Variable [ $i ; Value: $i + 1 ]  
12  End Loop  
13 End If
```




Script Workspace (WatchmanGroups)

Scripts

Get Watchman Groups

```
1 Insert from URL [ Select ; With dialog: Off ; Target: $GroupList ;  
  "https://" & WatchmanQuery::SubDomain & ".monitoringclient.com/v2.5/groups?api_key=" & WatchmanQuery::APIkey ]  
2 Set Variable [ $GroupCount ; Value: ValueCount ( JSONListKeys ( Groups::Results ; "groups" ) ) ]  
3 Set Variable [ $i ; Value: 0 ]  
4 If [ $GroupCount > 0 ]  
5   Loop  
6     New Record/Request  
7     Set Field [ Groups::groupNAME ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]"name" ) ]  
8     Set Field [ Groups::groupUID ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]"uid" ) ]  
9     Commit Records/Requests [ With dialog: Off ]  
10    Exit Loop If [ $i ≥ $GroupCount ]  
11    Set Variable [ $i ; Value: $i + 1 ]  
12  End Loop  
13 End If
```




Script Workspace (WatchmanGroups)

Scripts

Search

Get Watchman Groups *

```
1 Insert from URL [ Select ; With dialog: Off ; Target: $GroupList ;  
  "https://" & WatchmanQuery::SubDomain & ".monitoringclient.com/v2.5/groups?api_key=" & WatchmanQuery::APIkey ]  
2 Set Variable [ $GroupCount ; Value: ValueCount ( JSONListKeys ( Groups::Results ; "groups" ) ) ]  
3 Set Variable [ $i ; Value: 0 ]  
4 If [ $GroupCount > 0 ]  
5   Loop  
6     New Record/Request  
7     Set Field [ Groups::groupNAME ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]"name" ) ]  
8     Set Field [ Groups::groupUID ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]"uid" ) ]  
9     Commit Records/Requests [ With dialog: Off ]  
10    Exit Loop If [ $i ≥ $GroupCount ]  
11    Set Variable [ $i ; Value: $i + 1 ]  
12  End Loop  
13 End If
```




Script Workspace (WatchmanGroups)

Scripts

Search

Get Watchman Groups *

```
1 Insert from URL [ Select ; With dialog: Off ; Target: $GroupList ;  
  "https://" & WatchmanQuery::SubDomain & ".monitoringclient.com/v2.5/groups?api_key=" & WatchmanQuery::APIkey ]  
2 Set Variable [ $GroupCount ; Value: ValueCount ( JSONListKeys ( Groups::Results ; "groups" ) ) ]  
3 Set Variable [ $i ; Value: 0 ]  
4 If [ $GroupCount > 0 ]  
5   Loop  
6     New Record/Request  
7     Set Field [ Groups::groupNAME ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]"name" ) ]  
8     Set Field [ Groups::groupUID ; JSONGetElement ( Groups::Results ; "groups[" & $i & "]"uid" ) ]  
9     Commit Records/Requests [ With dialog: Off ]  
10    Exit Loop If [ $i ≥ $GroupCount ]  
11    Set Variable [ $i ; Value: $i + 1 ]  
12  End Loop  
13 End If
```




<https://www.soliantconsulting.com/blog/parsing-json-filemaker-16/>

<https://dbservices.com/articles/filemaker-json-functions/>

<https://github.com/SoliantMike/PHP-HTTPDebug>

precursor.ca/slides



Winnipeg

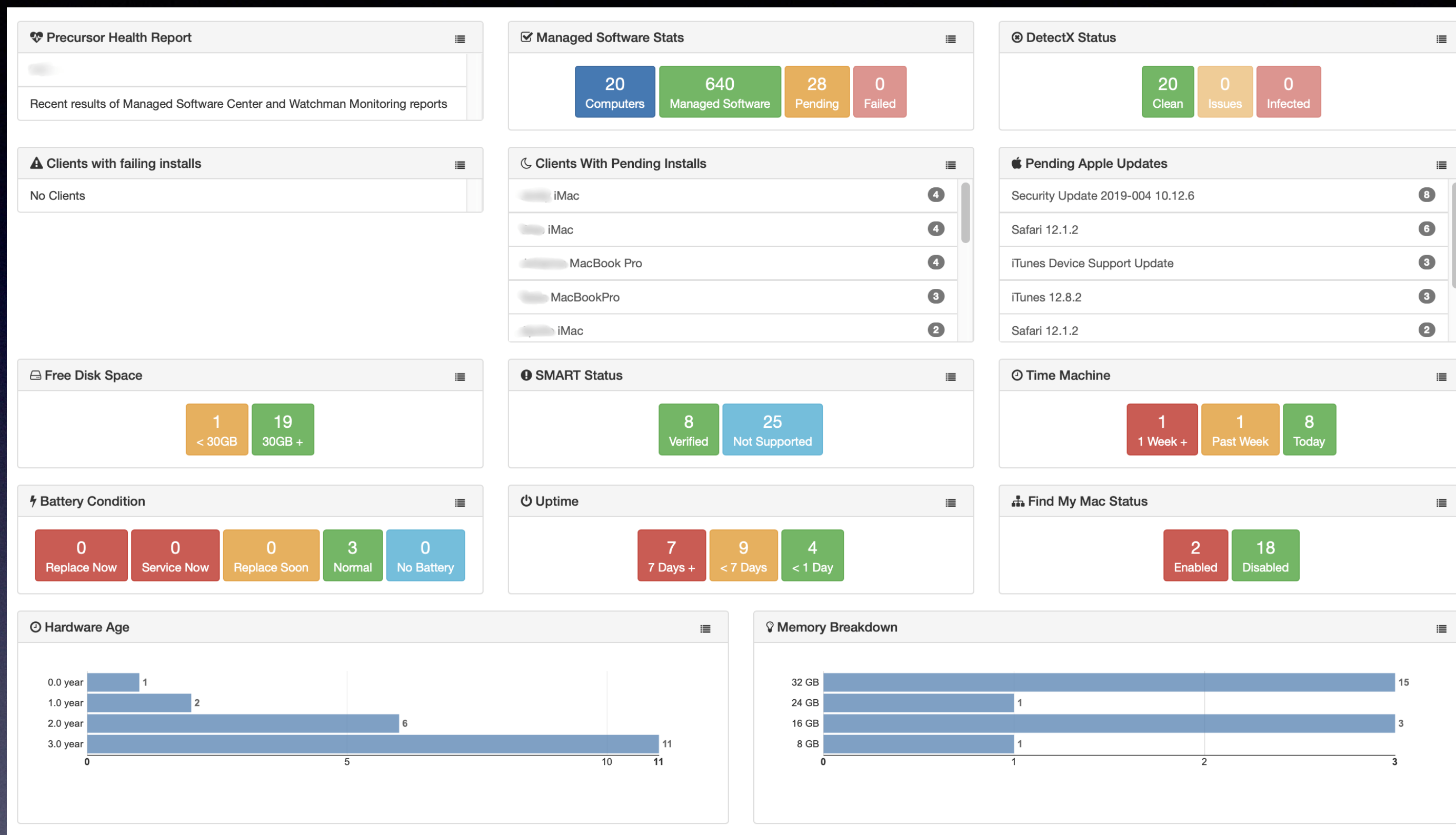
precursor.ca/slides





Monthly Reports

precursor.ca/slides



precursor.ca/slides



MunkiReport 9/5/19, 3:19 PM

♥ Precursor Health Report

Recent results of Managed Software Center and Watchman Monitoring reports

☑ Managed Software Stats

21 Computers
681 Managed Software
23 Pending
0 Failed

⊕ DetectX Status

20 Clean	1 Issue
0 Infected	

⚠ Clients with failing installs

No Clients

🌙 Clients With Pending Installs

iMac	4
MacBookPro	4
MacBook Pro	4
iMac	3
MB Pro	3
iMac	2
iMac	1
iMac	1
iMac	1

🍏 Pending Apple Updates

iTunes Device Support Update	3
iTunes 12.8.2	3
Safari 12.1.2	2
Safari 12.1.2	2
Security Update 2019-004 10.12.6	2
Security Update 2019-004 10.13.6	2
macOS Mojave 10.14.6 Supplemental Update	1
macOS 10.14.6 Update	1

<https://munkireport.precursor.ca/index.php?/show/dashboard/stats> Page 1 of 3

precursor.ca/slides



Each Month!

- **Log in to each client's**
- **Screenshot each client**
- **Rename screenshot**
- **Attach to email**
- **Send mail**

“Like an animal!”

precursor.ca/slides



precursor.ca/slides



Each Month!

- **Do nothing**



precursor.ca/slides



Watchman Monitoring



precursor.ca/slides



Group Status Report

Group

MacTech x 4 Computers

Computer warnings

Include details about Computer Warnings on Report

Notes

Include Notes from Group and its Computers

Start Date

End Date


Note: Pinned Notes are displayed regardless of the selected date range.


[Generate PDF](#)



Group created on 2019-02-12.


4
Computers


2
Warnings


0
Expirations

COMPUTER RECORDS



Server

First reported on 2019-02-12.



1 Error

LAST USER
701admin

SERIAL NUMBER
C07 [REDACTED] JY7

RAM
16 GB

OS VERSION

EST. MANUFACTURE DATE

precursor.ca/slides



New API Key ✕

Name

MyWatchmanKey

Names are for reference only.

Scope

Full Access

API Keys can have full access or be restricted to AppleCare lookup data only. [Learn More](#)

Cancel Save

`https://your_subdomain.monitoringclient.com/settings/api`

`precursor.ca/slides`



The screenshot shows the Watchman Monitoring API documentation for the 'Plugin Result Object'. The page is titled 'The Plugin Result Object' and includes a navigation sidebar on the left with sections for 'API V2.5' and 'ENDPOINTS'. The main content area contains a description of plugin results, a note about slug and uid attributes, a table of attributes, and a definition of the object with an example JSON object.

API V2.5

- Introduction
- Stay Up-to-date
- Versioning
- Authentication
- Expanding Objects
- Nested Expansion
- Errors
- Pagination
- Change Log

ENDPOINTS

- Groups
- Computers
- Expirations
- Expiration Manufacturers
- Expiration Products
- Notes
- Plugin Results**
 - Overview
 - Mute
 - Users

The Plugin Result Object

Plugin Results are returned with their related Computer Records. Within Watchman Monitoring there is a concept of "service plugins". These show in Plugin Results with visible set to false. They would not be displayed in the Dashboard and should be hidden in any alternate UI you build for your users.

Note: In former versions of the API the slug was being passed as the id. In version 2.5 and above, the slug and uid are different attributes.

Attributes

Param	Default	Purpose
uid	string	The unique UID for the plugin result
name	string	Name of the plugin
slug	string	Slug for the plugin
status	string	Current reported status: [OK, Warning, Alert, Informational]
details	text	The plugins reported details
last_run	integer	The last time the plugin ran
muted	boolean	Whether or not the plugin is muted
mute_type	string	The type of mute in place: ["muted_forever", "timed_mute", "ticket_muted"]
mute_description	string	A brief description of the mute <small>Only included if plugin is muted</small>

Definition

GET `https://your_subdomain.monitoringclient.com/v2.5/computers/[watchman_id]?api_key=This_Example_API_key&expand[]=plugin_results`

Example Object

```
{
  "uid": "pr_28d2c37b9a",
  "name": "Monitoring Agent",
  "slug": "monitoring-agent",
  "status": "OK",
  "details": "The internal check of the Watchman Monitoring software reports no issues.",
  "last_run": 1372324943,
  "muted": true,
  "mute_type": "timed_mute",
  "mute_description": "Plugin is muted until Sep 27th 2013.",
  "muted_until": 1380258000,
  "visible": true
}
```

https://api.watchmanmonitoring.com/#plugin_result

precursor.ca/slides



Specify Calculation

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

Plugins

Search

- PrimaryKey
- CreationTimestamp
- CreatedBy
- ModificationTimestamp
- ModifiedBy
- groupName
- groupID
- computerNAME
- pluginNAME
- pluginUID
- pluginDETAILS
- pluginSLUG

`"https://" & WatchmanQuery::SubDomain & ".monitoringclient.com/v2.5/groups?api_key=" & WatchmanQuery::APIkey`

Calculation result must be Text

Cancel OK

precursor.ca/slides



```
Scripts
Search
Get Watchman Groups

Get Watchman Groups

1 Go to Layout [ "Groups" (Groups) ; Animation: None ]
2 If [ WatchmanQuery::APIkey ≠ "" and WatchmanQuery::SubDomain ≠ "" ]
3   Insert from URL [ Select ; With dialog: Off ; Target: $GroupList ;
4     "https://" & WatchmanQuery::SubDomain & ".monitoringclient.com/v2.5/groups?api_key=" & WatchmanQuery::APIkey ]
5   Set Field [ Groups::Results ; "{" & ¶ & "\"groups\"& " : " & ¶ & $GroupList & ¶ & "]" ]
6   Set Variable [ $GroupCount ; Value: ValueCount ( JSONListKeys ( Groups::Results ; "groups" ) ) ]
7   Set Variable [ $i ; Value: 0 ]
8   If [ $GroupCount > 0 ]
9     Loop
10    New Record/Request
11    Set Field [ Groups::groupNAME ; JSONGetElement ( WatchmanQuery::Results ; "groups[" & $i & "]name" ) ]
12    Set Field [ Groups::groupUID ; JSONGetElement ( WatchmanQuery::Results ; "groups[" & $i & "]uid" ) ]
13    Set Field [ Groups::groupDESCRIPTION ; JSONGetElement ( WatchmanQuery::Results ; "groups[" & $i & "]description" ) ]
14    Set Field [ Groups::groupSLUG ; JSONGetElement ( WatchmanQuery::Results ; "groups[" & $i & "]slug" ) ]
15    Set Field [ Groups::groupREFERENCE_EMAIL ; JSONGetElement ( WatchmanQuery::Results ; "groups[" & $i & "]reference_email" ) ]
16    Set Field [ Groups::groupCONTACT_EMAIL ; JSONGetElement ( WatchmanQuery::Results ; "groups[" & $i & "]contact_email" ) ]
17    Set Field [ Groups::groupCREATED_AT ; JSONGetElement ( WatchmanQuery::Results ; "groups[" & $i & "]created_at" ) ]
18    Commit Records/Requests [ With dialog: Off ]
19    Set Variable [ $i ; Value: $i + 1 ]
20    Exit Loop If [ $i ≥ $GroupCount ]
21  End Loop
22 End If
23 Go to Layout [ "WatchmanQuery" (WatchmanQuery) ; Animation: None ]
24 Exit Script [ Text Result: ]
25 Else
26 Perform Script [ Specified: From list ; "Settings Test" ; Parameter: ]
27 End If
```




Specify Calculation

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

Plugins

Search

- PrimaryKey
- CreationTimestamp
- CreatedBy
- ModificationTimestamp
- ModifiedBy
- groupName
- groupID
- computerNAME
- pluginNAME
- pluginUID
- pluginDETAILS
- pluginSLUG

```
"https://" & WatchmanQuery::SubDomain & ".monitoringclient.com/v2.5/computers/?api_key=" & WatchmanQuery::APIkey & "&group_id=" & Plugins::groupID & "&expand[]=plugin_results"
```

Calculation result must be Text

Cancel OK

precursor.ca/slides



Specify Calculation

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

Plugins

Search

- PrimaryKey
- CreationTimestamp
- CreatedBy
- ModificationTimestamp
- ModifiedBy
- groupName
- groupID
- computerNAME
- pluginNAME
- pluginUID
- pluginDETAILS
- pluginSLUG

```
{ " & ¶ &  
  "plugins\" & ¶ & " : " & ¶ &  
  $PluginsList & ¶ &  
 }
```

Calculation result must be Text

Cancel OK

precursor.ca/slides



```
Scripts
Get Watchman Warning Talley

1 Set Variable [ $ComputerCount ; Value: ValueCount ( JSONListKeys ( Plugins::Results ; "plugins" ) ) ]
2 Set Variable [ $c ; Value: 0 ]
3 Set Variable [ $Warnings ; Value: 0 ]
4 If [ $ComputerCount > 0 ]
5     Loop
6         Set Variable [ $PluginsCount ;
7             Value: ValueCount ( JSONListKeys ( Plugins::Results ; "plugins[" & $c & "].plugin_results" ) ) ]
8         Set Variable [ $i ; Value: 0 ]
9         If [ $PluginsCount > 0 ]
10            Loop
11                If [ JSONGetElement ( Plugins::Results ; "plugins[" & $c & "].plugin_results[" & $i & "]status" ) = "warning" ]
12                    Set Variable [ $Warnings ; Value: $Warnings + 1 ]
13                End If
14                Set Variable [ $i ; Value: $i + 1 ]
15                Exit Loop If [ $i ≥ $PluginsCount ]
16            End Loop
17        End If
18        Set Variable [ $c ; Value: $c + 1 ]
19        Exit Loop If [ $c ≥ $ComputerCount ]
20    End Loop
21 Set Field [ Plugins::warningsTOTAL ; $Warnings ]
```




WARNINGS



groupName

groupID

Results

```
{
  "plugins":
  [{"uid":"c_f3b96b971d","description":null,"watchman_id":"20190212-G2SZ-
N4EEMA","client_id":"20190212-G2SZ-
N4EEMA","agent_version":"6.6.7.115","build_number":"6.6.7.115","last_report":156
8066069,"created_at":1550016745,"group":"g_021a3efa95","computer_name":"70
1 Server","custom_name":null,"contact_email":null,"reference_email":
null,"reference_email_computer":null,"reference_email_group":
```

ResultsFormatted

```
{
  "plugins":
  [
    {
      "active_mac_address": "C8:2A:14:58:F6:D9",
      "agent_removed": false,
      "agent_version": "6.6.7.115",
      "apple_product_description": "Mac mini Server (Mid 2011)",
      "ard_field1": "701 Server",
      "ard_field2": "",
      "ard_field3": "",
      "ard_field4": "",
      "asset_id": null,
      "beacon_missing_threshold_in_minutes": 5,
      "beacon_reporting": true,
      "bomgar_url": null,
      "boot_volume_capacity": "499.25 GB",
      "boot_volume_usage": "55.42 GB",
      "boot_volume_usage_percent": "11.1",
      "build_number": "6.6.7.115".
```

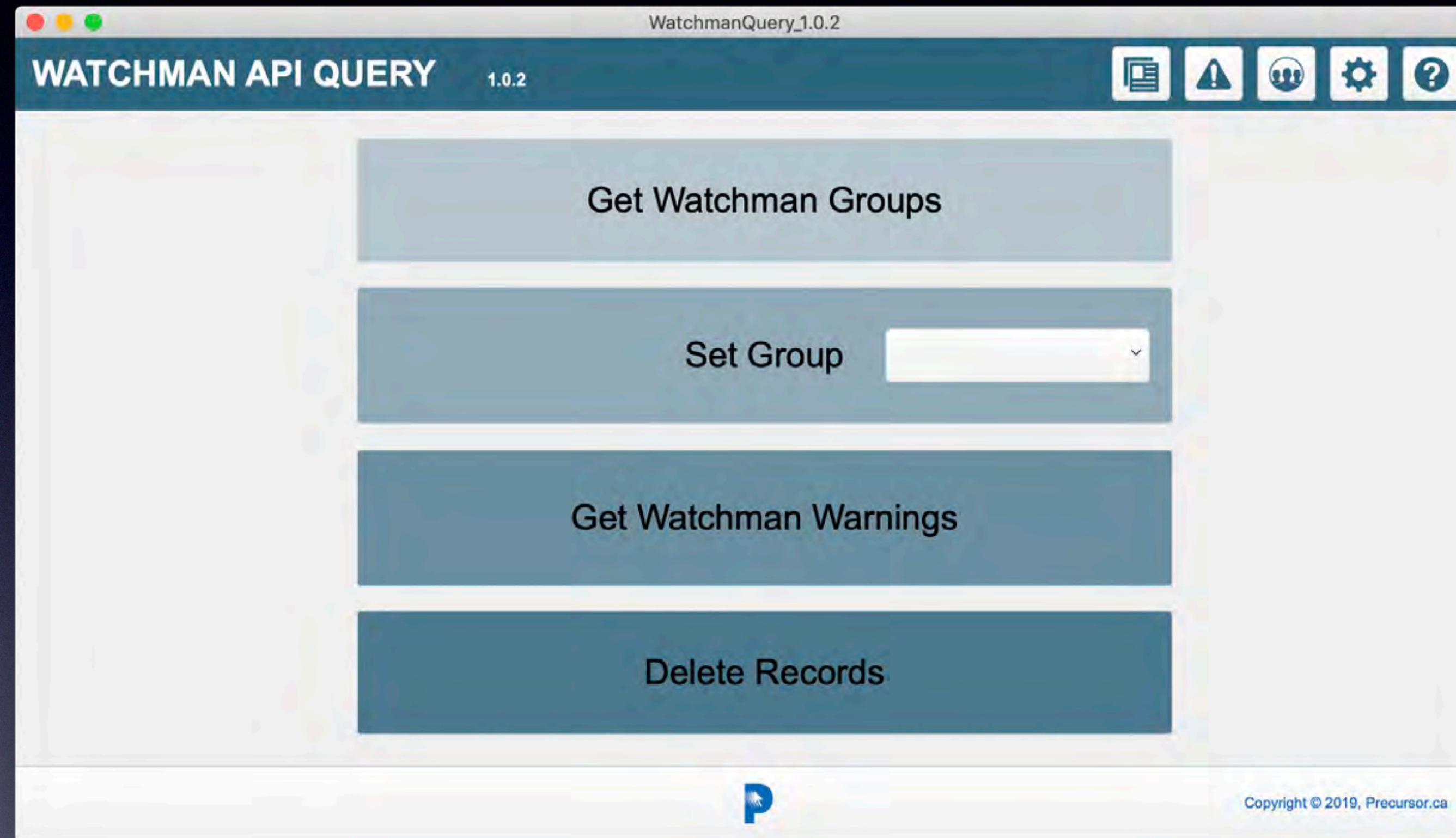
Get Watchman Plugins

Get Warning Total

Delete Records

2
Warnings





precursor.ca/slides



Zendesk



precursor.ca/slides



zendesk developers Q Docs ▾ Blog Get Started Login / Signup

Support API

Support API ▾

- Introduction
- Getting started
- Restrictions and Responsibilities
- API Changes
- Side-Loading
- Incremental Exports
- Search**
- Results limit
- Query basics
- JSON Format
- List Search Results
- Errors JSON Format

Search

The search API is a unified search API that returns tickets, users, and organizations. You can define filters to narrow your search results according to resource type, dates, and object properties, such as ticket requester or tag.

To search articles in Help Center, see [Search](#) in the Help Center API documentation.

To use the API with Python or Perl, see [Searching with the Zendesk API](#).

Note: It can take up to a few minutes for new tickets, users, and other resources to be indexed for search. If new resources don't appear in your search results, wait a few minutes and try again.

Results limit

The Search API returns a limit of 2,000 pages with 100 results per page, or 200,000 results per query. If you request a page past the limit (`?page=2001`), a 422 (Insufficient Resource Error) is returned.

If you need to retrieve large datasets, Zendesk recommends serializing the search into smaller chunks by limiting results to a specific date range. Alternatively, if you need to export and warehouse large amounts of data, consider using one of the [Incremental Export](#) endpoints.

Note: Starting **October 15, 2019**, the limit of 200,000 results per query will be **limited to 1,000** results. Please refer to our [Help Center article](#) for more information.

https://developer.zendesk.com/rest_api/docs/support/

precursor.ca/slides



API Token Description (optional)

API Token

⚠ Make sure to copy and store this token. We won't show it again after you click Save or leave this page.

https://your_subdomain.zendesk.com/agent/admin/api/settings/tokens

precursor.ca/slides



```
"https://subdomain.Zendesk.com/api/v2/search.json?  
query=type:ticket organization:orgName solved>YYYY-MM-DD"
```

```
"https://" & ZendeskQuery::SubDomain & ".Zendesk.com/api/v2/  
search.json?query=type:ticket organization:\"\" &  
Tickets::ticketORGANIZATION & "\" solved>" & $$Period & "\""
```

precursor.ca/slides



```
Insert from URL [ Select ; With dialog: Off ; Target: $ResultsList ;  
"https://" & ZenDeskQuery::SubDomain & ".zendesk.com/api/v2/search.json?query=type:ticket organization:\" & Tickets::ticketORGA..  
cURL options: "-user \" & ZenDeskQuery::UserName & " /token:" & ZenDeskQuery::APIkey & "\" ]
```

- Select entire contents
- Target:
- Specify URL
- Verify SSL Certificates
- Specify cURL options

precursor.ca/slides



Insert from URL [Select ; With dialog: Off ; Target: `$ResultsList` ;
"https://" & ZenDeskQuery::SubDomain & ".zendesk.com/api/v2/search.json?query=type:ticket organization:\"\" & Tickets::ticketORGA...
cURL options: "-user \"\" & ZenDeskQuery::UserName & " /token:" & ZenDeskQuery::APIkey & "\"]

- Select entire contents
- Target: Specify...
- Specify URL: Specify...
- Verify SSL Certificates
- Specify cURL options: Specify...



Specify Calculation

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

ZenDeskQuery: `"https://" & ZenDeskQuery::SubDomain & ".zendesk.com/api/v2/search.json?query=type:ticket organization:\"\" & Tickets::ticketORGANIZATION & "\" solved>" & $$SinceDATE & "\"`

Search: []

PrimaryKey
CreationTimestamp
CreatedBy

Insert from URL

```
"https://" & ZenDeskQuery::SubDomain & ".Zendesk.com/api/v2/  
search.json?query=type:ticket organization:\"\" &  
Tickets::ticketORGANIZATION & "\" solved>" & $$Period & "\""
```

precursor.ca/slides



```
Insert from URL [ Select ; With dialog: Off ; Target: $ResultsList ;  
"https://" & ZenDeskQuery::SubDomain & ".zendesk.com/api/v2/search.json?query=type:ticket organization:\" & Tickets::ticketORGA...  
cURL options: "-user \" & ZenDeskQuery::UserName & " /token:" & ZenDeskQuery::APIkey & "\" ]
```

- Select entire contents
- Target: Specify...
- Specify URL: Specify...
- Verify SSL Certificates
- Specify cURL options: Specify...

Specify Calculation

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

ZenDeskQuery: | "https://" & ZenDeskQuery::SubDomain & ".zendesk.com/api/v2/search.json?query=type:ticket organization:\" & Tickets::ticketORGANIZATION & "\" solved>" & \$\$SinceDATE & "\"

Search

PrimaryKey

CreationTimestamp

CreatedBy

Specify Calculation

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

ZenDeskQuery: | "-user \" & ZenDeskQuery::UserName & " /token:" & ZenDeskQuery::APIkey & "\"

Search

PrimaryKey

CreationTimestamp

CreatedBy

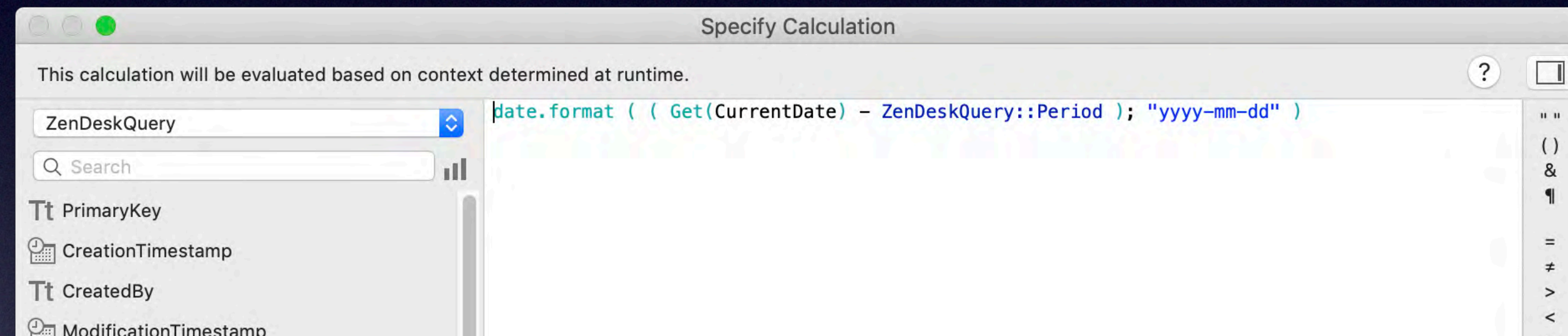
cURL Options

`"-user \" & ZendeskQuery::UserName & " /token:" & ZendeskQuery::APIkey & "\"`

precursor.ca/slides



```
date.format ( ( Get(CurrentDate) - ZendeskQuery::Period ); "yyyy-mm-dd" )
```



date.format custom function by Tim Anderson Group

<https://www.briandunning.com/cf/893>

precursor.ca/slides



ORGANIZATIONS



organizationNAME MacTech

organizationID 2832325606

organizationCREATED_AT 2019-09-30T12:13:31Z

organizationNOTES

organizationDESCRIPTION

organizationDOMAINS mactech.com

organizationURL https://precursorca.zendesk.com/api/v2/organizations/2832325606.json

Get ZenDesk Organizations

Delete Organizations

Results

```
{
  "organizations": [
    {
      "url": "https://precursorca.zendesk.com/api/v2/organizations/2832325606.json",
      "id": "2832325606",
      "name": "MacTech",
      "shared_tickets": false,
      "shared_comments": false,
      "external_id": null,
      "created_at": "2017-04-05T22:01:32Z",
      "updated_at": "2017-04-05T22:01:32Z",
      "domain_names": [
        "mactech.com"
      ],
      "details": "",
      "notes": "",
      "group_id": null,
      "tags": [],
      "organization_fields": {}
    },
    {
      "url": "https://precursorca.zendesk.com/api/v2/organizations/2832325606.json",
      "id": "2832325606",
      "name": "MacTech",
      "shared_tickets": false,
      "shared_comments": false,
      "external_id": null,
      "created_at": "2017-03-30T12:07:24Z",
      "updated_at": "2017-03-30T12:07:24Z",
      "domain_names": [
        "mactech.com"
      ],
      "details": "",
      "notes": "",
      "group_id": null,
      "tags": [],
      "organization_fields": {}
    },
    {
      "url": "https://precursorca.zendesk.com/api/v2/organizations/2832325606.json",
      "id": "2832325606",
      "name": "MacTech",
      "shared_tickets": false,
      "shared_comments": false,
      "external_id": null,
      "created_at": "2017-05-21T13:28:13Z",
      "updated_at": "2017-05-21T13:28:13Z",
      "domain_names": [
        "mactech.com"
      ],
      "details": "",
      "notes": "",
      "group_id": null,
      "tags": [],
      "organization_fields": {}
    },
    {
      "url": "https://precursorca.zendesk.com/api/v2/organizations/2832325606.json",
      "id": "2832325606",
      "name": "MacTech",
      "shared_tickets": false,
      "shared_comments": false,
      "external_id": null,
      "created_at": "2017-03-30T12:07:52Z",
      "updated_at": "2017-03-30T12:08:03Z",
      "domain_names": [
        "mactech.com"
      ],
      "details": "",
      "notes": "",
      "group_id": null,
      "tags": [],
      "organization_fields": {}
    },
    {
      "url": "https://precursorca.zendesk.com/api/v2/organizations/2832325606.json",
      "id": "2832325606",
      "name": "MacTech",
      "shared_tickets": false,
      "shared_comments": false,
      "external_id": null,
      "created_at": "2017-03-30T12:08:37Z",
      "updated_at": "2017-03-30T12:08:37Z",
      "domain_names": [
        "mactech.com"
      ],
      "details": "",
      "notes": "",
      "group_id": null,
      "tags": []
    }
  ]
}
```




```
If [ ZenDeskQuery::APIKey ≠ "" and ZenDeskQuery::SubDomain ≠ "" ]
  Insert from URL [ Select ; With dialog: Off ; Target: $OrganizationList ;
  "https://" & ZenDeskQuery::SubDomain & ".zendesk.com/api/v2/organizations.json" ; cURL options:
  "-user \" & ZenDeskQuery::UserName & " /token:" & ZenDeskQuery::APIKey & "\" ]
  Set Field [ Organizations::Results ; $OrganizationList ]
  Set Variable [ $OrganizationCount ; Value: ValueCount ( JSONListKeys ( Organizations::Results ; "organizations" ) ) ]
  Set Variable [ $i ; Value: 0 ]
  If [ $OrganizationCount > 0 ]
    Loop
      New Record/Request
      Set Field [ Organizations::organizationNAME ; JSONGetElement ( Organizations::Results ; "organizations[" & $i & "]"name" ) ]
      Set Field [ Organizations::organizationID ; JSONGetElement ( Organizations::Results ; "organizations[" & $i & "]"id" ) ]
      Set Field [ Organizations::organizationDESCRIPTION ;
      JSONGetElement ( Organizations::Results ; "organizations[" & $i & "]"details" ) ]
      Set Field [ Organizations::organizationURL ; JSONGetElement ( Organizations::Results ; "organizations[" & $i & "]"url" ) ]
      Set Field [ Organizations::organizationDOMAINS ;
      JSONGetElement ( Organizations::Results ; "organizations[" & $i & "]"domain_names" ) ]
      Set Field [ Organizations::organizationNOTES ; JSONGetElement ( Organizations::Results ; "organizations[" & $i & "]"notes" ) ]
      Set Field [ Organizations::organizationCREATED_AT ;
      JSONGetElement ( Organizations::Results ; "organizations[" & $i & "]"created_at" ) ]
      Commit Records/Requests [ With dialog: Off ]
      Set Variable [ $i ; Value: $i + 1 ]
      Exit Loop If [ $i ≥ $OrganizationCount ]
    End Loop
  End If
  Exit Script [ Text Result: ]
Else
  Perform Script [ Specified: From list ; "Settings Test" ; Parameter: ]
End If
```




Specify Calculation

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

Tickets

JSONGetElement (Tickets::Results ; "count")

Search

- PrimaryKey
- CreationTimestamp
- CreatedBy
- ModificationTimestamp
- ModifiedBy
- ticketORGANIZATION
- ticketSTATUS
- ticketID
- ticketOrganizationID
- ticketGroupID
- ticketCREATED_AT
- ticketUPDATED_AT

Calculation result must be Number

Cancel OK

not
and
or
xor
^

precursor.ca/slides



Zendesk API TICKETS

ticketOrganizationID: 2832909866
ticketID: 17289
ticketSTATUS: open
ticketGroupID: 22129925
ticketCREATED_AT: 2022-02-17T21:12:13Z
ticketUPDATED_AT: 2022-02-17T21:12:17Z
ticketCUSTOMID: 360027632291
ticketTAGS: []
ticketSUBJECT: SERVER : macOS

5 OPEN 78 SOLVED

Get Solved Tickets Get Tickets
Get Open Tickets

Delete Ticket Records

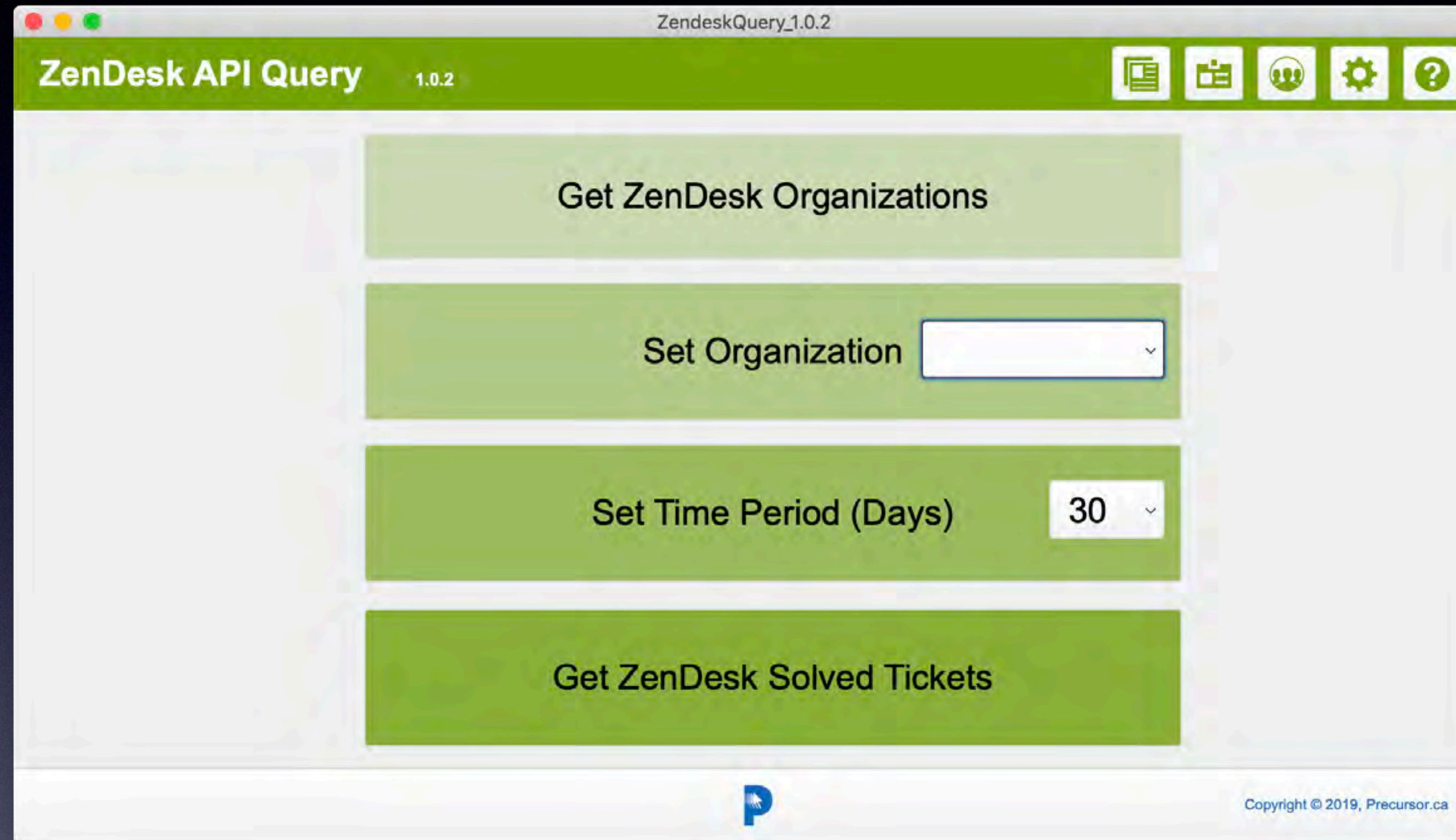
sinceDATE: 2022-02-04

Results:

```
{ "results": [ { "url": "https://[redacted].zendesk.com/api/v2/tickets/17471.json", "id": 17471, "external_id": null, "via": { "channel": "email", "source": { "from": { "address": "[redacted].ca", "name": "Scott", "to": { "name": "Precursor Systems", "address": "[redacted].com", "rel": null } }, "created_at": "2022-03-02T16:39:41Z", "updated_at": "2022-03-02T16:39:50Z", "type": null, "subject": "ADVICE : VectorWorks", "raw_subject": "ADVICE : VectorWorks", "description": "PA : scott\n" } } ] }
```

Formatted:

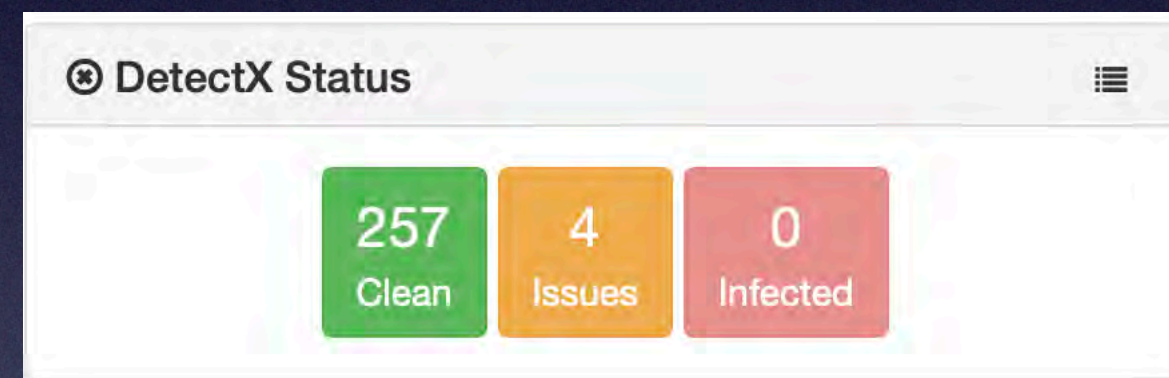
```
{ "count": 5, "facets": null, "next_page": null, "previous_page": null, "results": [ { "allow_attachments": true, "allow_channelback": false, "assignee_id": 379667215,
```

precursor.ca/slides



munkireport



precursor.ca/slides



Precursor Health Report

Recent results of Managed Software Center and Watchman Monitoring reports

Managed Software Stats

20 Computers	640 Managed Software	28 Pending	0 Failed
--------------	----------------------	------------	----------

DetectX Status

20 Clean	0 Issues	0 Infected
----------	----------	------------

Clients with failing installs

No Clients

Clients With Pending Installs

iMac	4
iMac	4
MacBook Pro	4
MacBookPro	3
iMac	2

Pending Apple Updates

Security Update 2019-004 10.12.6	8
Safari 12.1.2	6
iTunes Device Support Update	3
iTunes 12.8.2	3
Safari 12.1.2	2

Free Disk Space

1 < 30GB	19 30GB +
----------	-----------

SMART Status

8 Verified	25 Not Supported
------------	------------------

Time Machine

1 1 Week +	1 Past Week	8 Today
------------	-------------	---------

Battery Condition

0 Replace Now	0 Service Now	0 Replace Soon	3 Normal	0 No Battery
---------------	---------------	----------------	----------	--------------

Uptime

7 7 Days +	9 < 7 Days	4 < 1 Day
------------	------------	-----------

Find My Mac Status

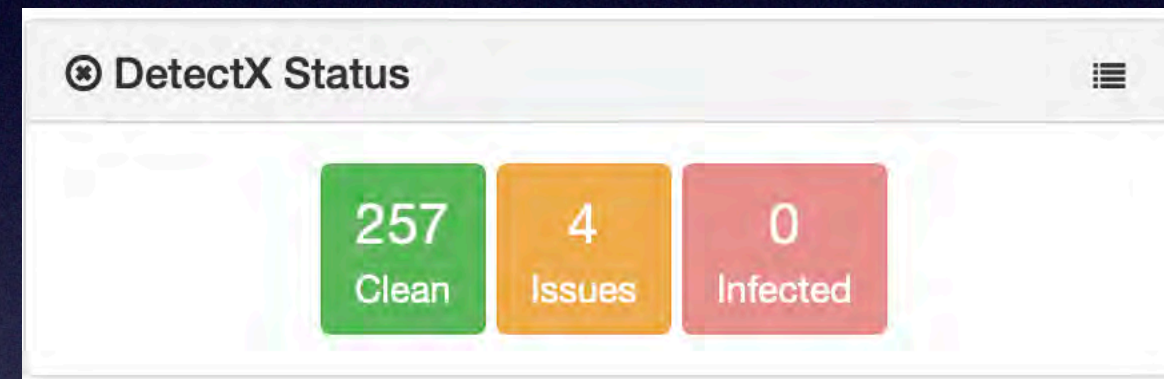
2 Enabled	18 Disabled
-----------	-------------

Hardware Age

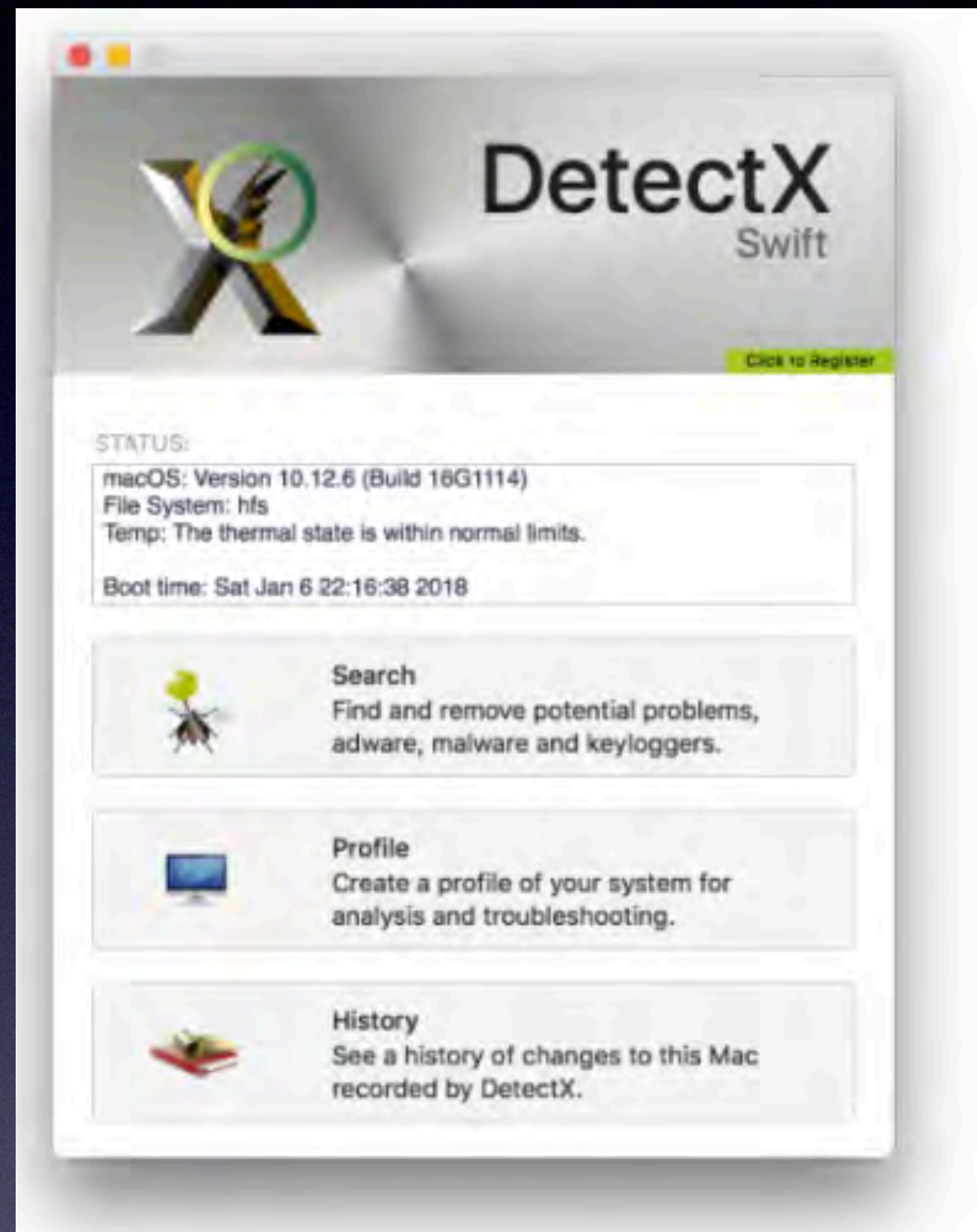
0.0 year	1
1.0 year	2
2.0 year	6
3.0 year	11

Memory Breakdown

32 GB	15
24 GB	1
16 GB	3
8 GB	1



precursor.ca/slides



<https://sqwarq.com/detectx/>

precursor.ca/slides



API

Arjen van Bochoven edited this page on Apr 12, 2017 · 6 revisions

To get data out of Munkireport there is a Datatables API that is also used for the listing reports. To help you get started use the small shell script or python script below. To get the data to CSV format, pipe the output to JSON to CSV parser (see below).

Shell Version

```
#!/bin/sh
#
# Script to run automated queries against the munkireport datatables API
#
# Results are returned in JSON format
# The actual entries are in the 'data' variable
#
# To make this work, set up a regular user in munkireport and adjust the
# proper values below
#
# Author: Arjen van Bochoven
# Date: 2015-11-06

# Retrieve data from munkireport
# DEBUG=1
MR_BASE_URL='https://yourmunkireportserver/index.php?'
MR_DATA_QUERY='/datatables/data'
MR_LOGIN='datauser'
MR_PASSWORD='secretpassword'

CLIENT_COLUMNS=(
    "machine.serial_number"
    "machine.hostname"
```

<https://github.com/munkireport/munkireport-php/wiki/API>

precursor.ca/slides

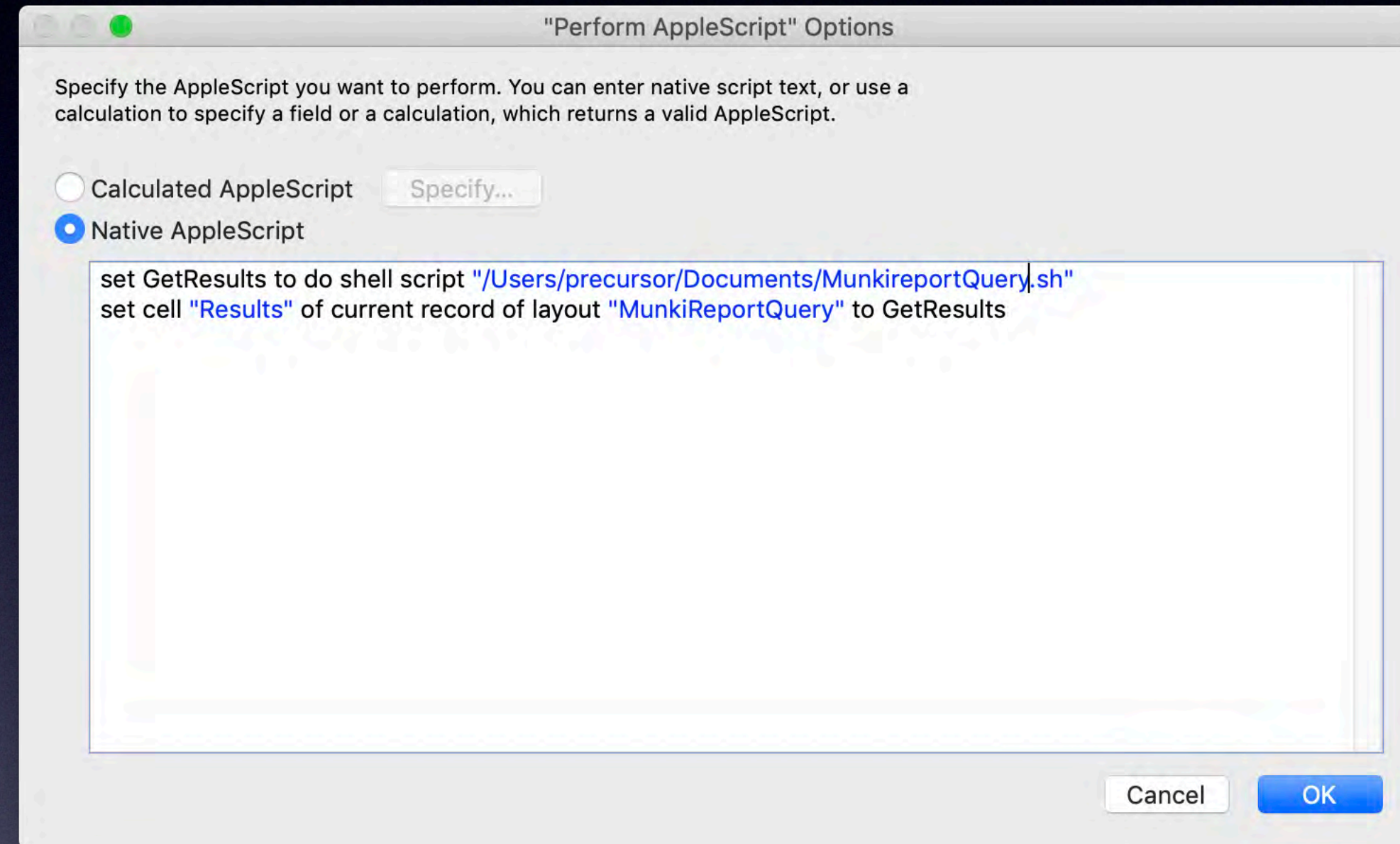


```
CLIENT_COLUMNS=(  
  "machine.serial_number"  
  "machine.hostname"  
  "machine.machine_desc"  
  "reportdata.timestamp"  
  "reportdata.console_user"  
  "machine.os_version"  
  "reportdata.remote_ip"  
  "munkireport.manifestname"  
)
```

“detectx.status”

“detectx.numberofissues”

precursor.ca/slides



precursor.ca/slides



https://munkireport.precursor.ca/index.php?/module/detectx/get_stats/

```
$.getJSON( appUrl + '/module/detectx/get_stats', function( data ) {  
  if(data.error){  
    //alert(data.error);  
    return;  
  }  
  
  $.each(['Clean', 'Issues', 'Infected'], function(index, type){
```

Status: Clean Issues Infected

precursor.ca/slides



```
# Authenticate and capture cookie
if [ $DEBUG ]; then echo 'Authenticating to munkireport..'; fi
COOKIE_JAR=$(curl -s --cookie-jar - --data "login=${MR_LOGIN}&password=${MR_PASSWORD}" --url "${MR_BASE_URL}login.php" --output /dev/null)
SESSION_COOKIE=$(echo $COOKIE_JAR | sed 's/.*PHPSESSID /PHPSESSID=/')

# Retrieve data with session cookie
columns_to_query CLIENT_COLUMNS[@]
if [ $DEBUG ]; then echo 'Retrieving client data..'; fi
echo $(curl -s --cookie "$SESSION_COOKIE" --data $MR_QUERY ${MR_BASE_URL}${MR_DATA})
```



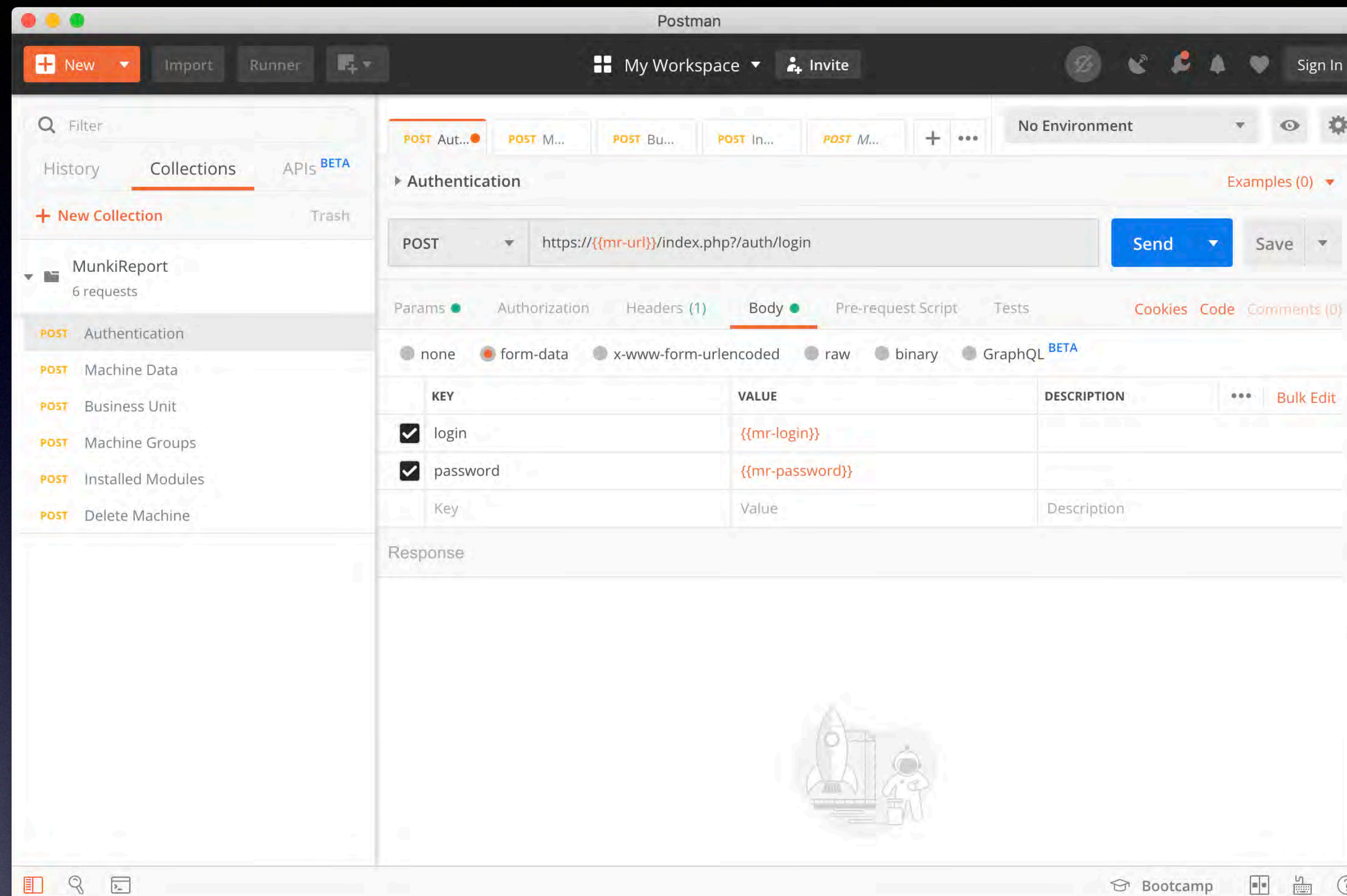

```
// the authentication data we wish to send  
"-d @$data " &amp;  
  
// set the Content type header to JSON  
"--header Content-Type:application/json " &amp;  
  
// create a cookie jar to store cookies in  
"--cookie-jar $$CookieJar "
```

```
Set Variable [ $CookieString ; Value : Substitute ( $$CookieJar ; "¶" ; Char(10) )
```

```
// for debugging ONLY set error reporting on  
"--show-error " &amp;  
"--dump-header $$Headers " &amp;  
  
// set the Content type header to JSON  
"--header Content-Type:application/json " &amp;  
  
// send the cookies with the request  
"--cookie $CookieString"
```

<https://msdev.co.uk/adventures-in-filemaker-curl-the-cookie-jar/>

precursor.ca/slides

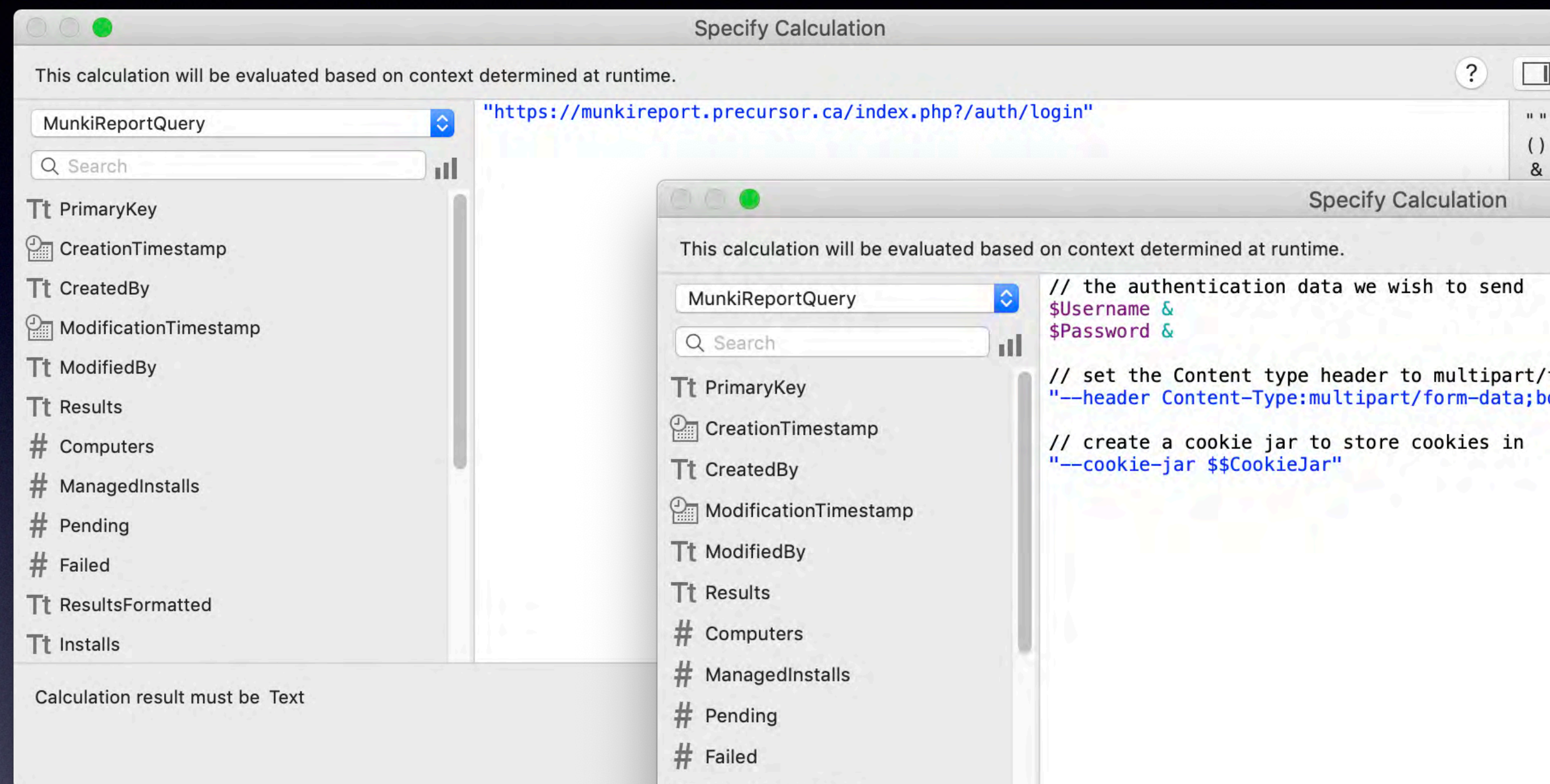


<https://github.com/joncrain/munkireport-postman-collection>

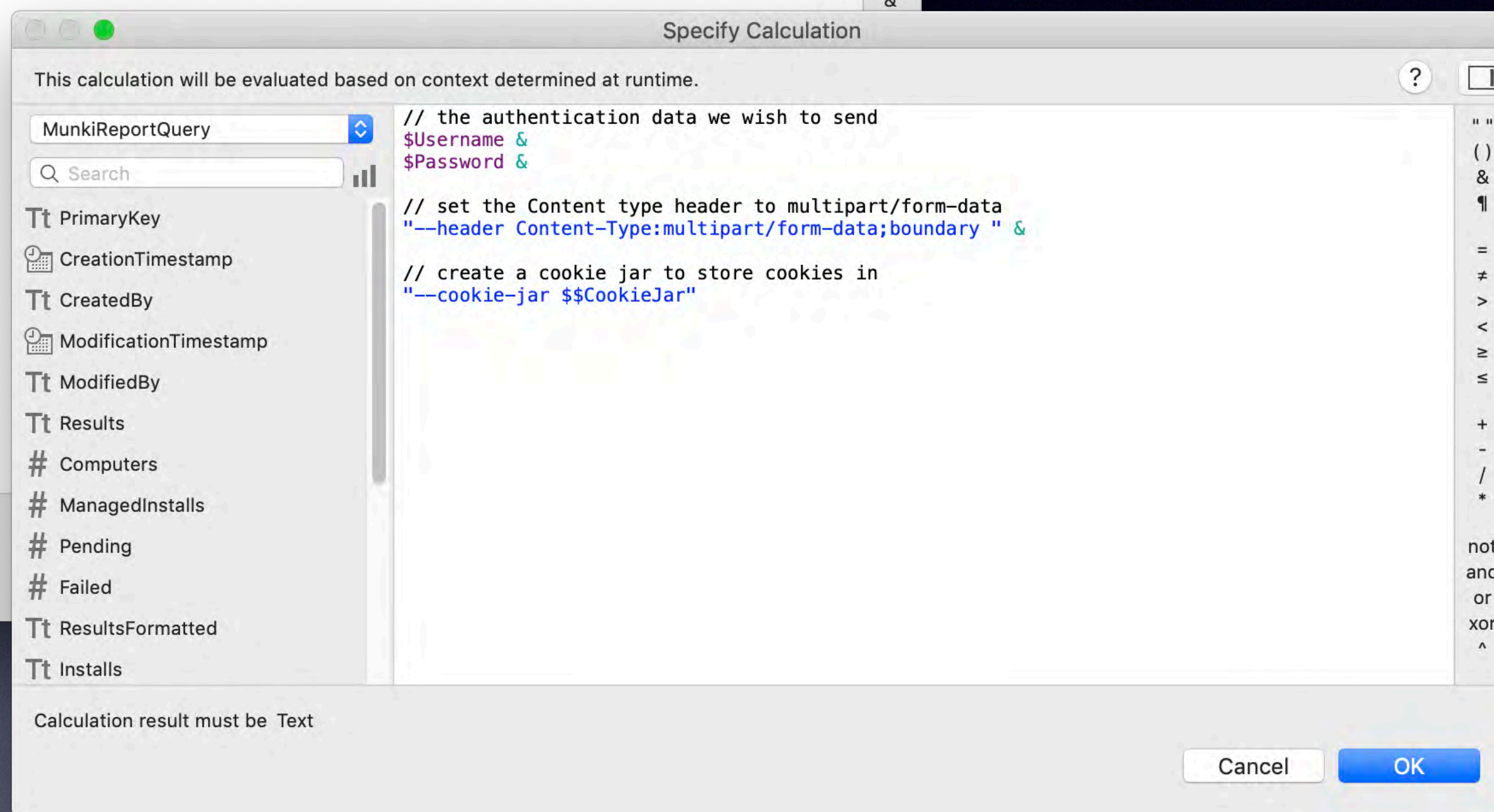
precursor.ca/slides



Specify URL



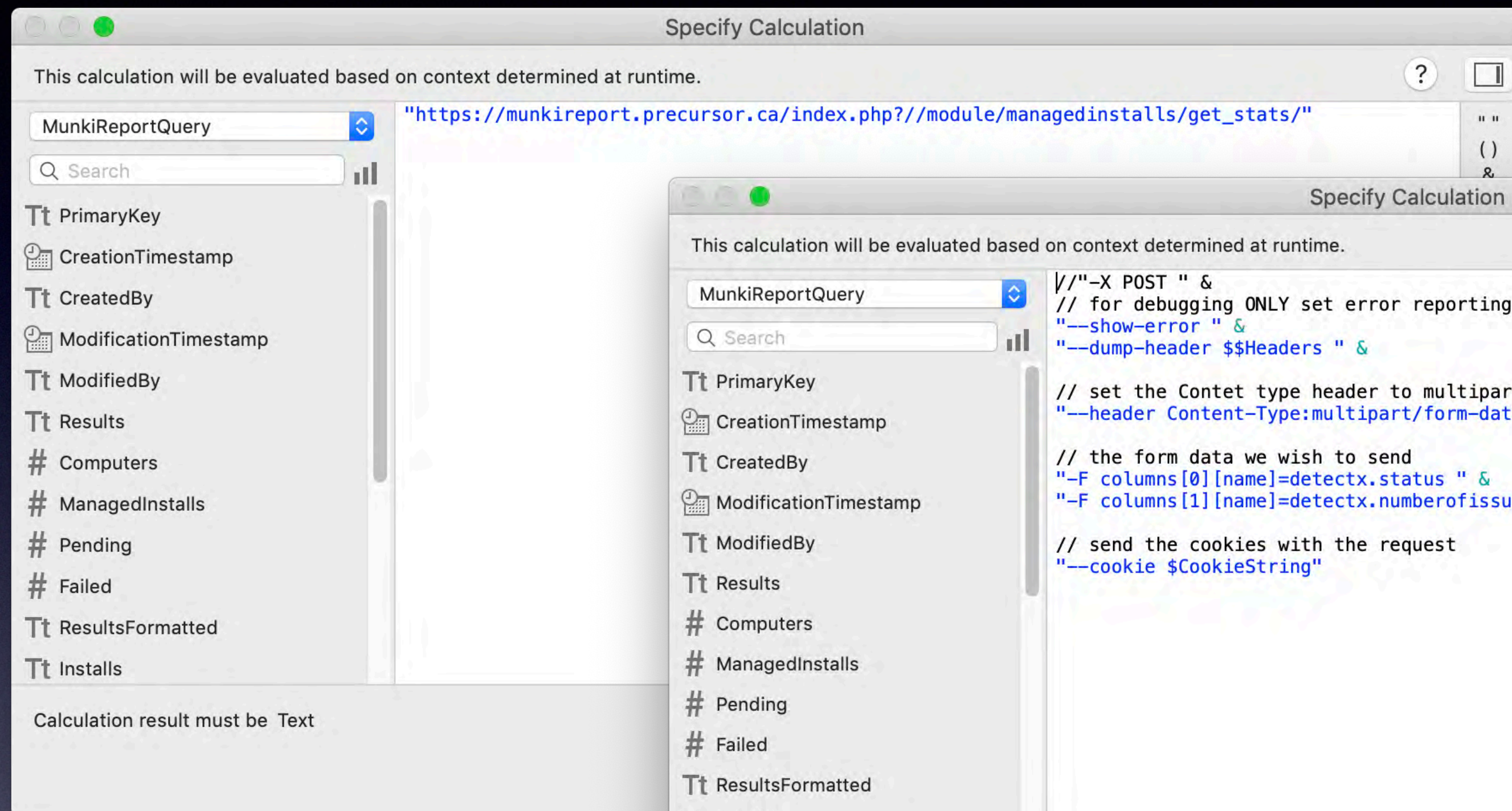
cURL options



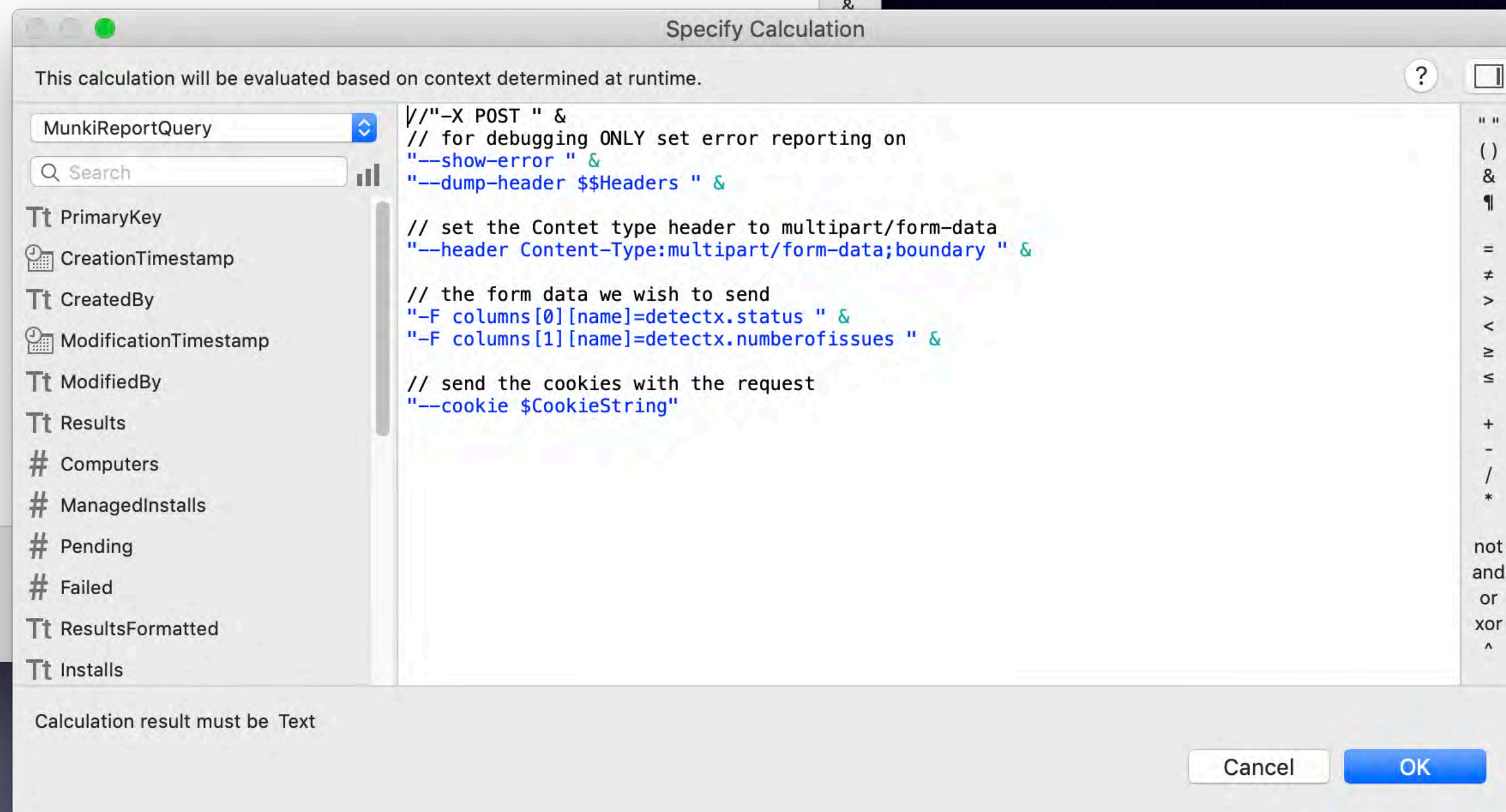
precursor.ca/slides



Specify URL



cURL options



precursor.ca/slides



Business Units

Nathan Perkins edited this page on Jan 30 · 4 revisions

Business Units

Business Units are a way to use one munkireport instance for more than one business/group.

Business Units can have:

- One or more Machine Groups (which reference Machines with the same machine_group)
- One or more Users (who can visit the unit pages)
- One or more Managers (who can add/delete users, add/delete Machine Groups)

Configure Business Units

If you want to enable Business Units, add the following line to `.env` :

```
ENABLE_BUSINESS_UNITS=TRUE
```

<https://github.com/munkireport/munkireport-php/wiki/Business-Units>

precursor.ca/slides



```
Set Variable [ $theCount ; Value: JSONListValues ( MunkiReportQuery::Results ; "data" ) ]  
Set Field [ MunkiReportQuery::DetectXclean ; PatternCount ( $theCount ; "Clean" ) ]  
Set Field [ MunkiReportQuery::DetectXissues ; PatternCount ( $theCount ; "Issues" ) ]  
Set Field [ MunkiReportQuery::DetectXinfected ; PatternCount ( $theCount ; "Infected" ) ]
```




DetectX
MALWARE SCANNING

27 Clean	0 Issues	0 Infected
-------------	-------------	---------------

precursor.ca/slides



Managed Software Center



precursor.ca/slides



MunkiReport Dashboard

Client Activity: 5382 (Total Clients: 6553 | Clients per hour: 1588)

Events:

- parinaz | 2 Munki warnings (2 minutes ago)
- MacBook Pro van Farzanah | 2 Munki warnings (3 minutes ago)
- Raoul Eireiners MacBook Air | 2 Munki warnings (3 minutes ago)
- MacBook Air van Anniek | Munki warning: Download of ATLAS.ti.license failed: HTTP result 403: forbidden (4 minutes ago)
- MacBook Air van Jeyam | Munki error: Could not retrieve managed install primary manifest. (4 minutes ago)
- Floris's MacBook Air | Munki warning: Download of ATLAS.ti.license failed: HTTP result 403: forbidden (4 minutes ago)
- MacBook Air van Desiree | Munki warning: Download of endnote_x8_license failed: HTTP result 403: forbidden (5 minutes ago)

New Clients (31):

- Yash's MacBook Pro (an hour ago)
- MacBook Air van Titia (an hour ago)
- Niels's MacBook Pro (3 hours ago)
- Zhaoxin's MacBook Pro (4 hours ago)
- MacBook Pro van Max (18 hours ago)

Pending Apple Updates:

- Security Update 2018-005 10.12.6 (8)
- Safari 12.0.1 (4)
- iTunes 12.8 (3)
- Security Update 2018-004 10.12.6 (2)
- Security Update 2018-004 10.11.6 (1)

Pending Installs:

- Microsoft Word for Mac 16.16.18111001 (78)
- Microsoft Excel for Mac 16.16.18111001 (49)
- Microsoft Outlook for Mac 16.16.18111001 (46)
- Google Chrome 70.0.3538.110 (44)
- Microsoft PowerPoint for Mac 16.16.18111001 (31)

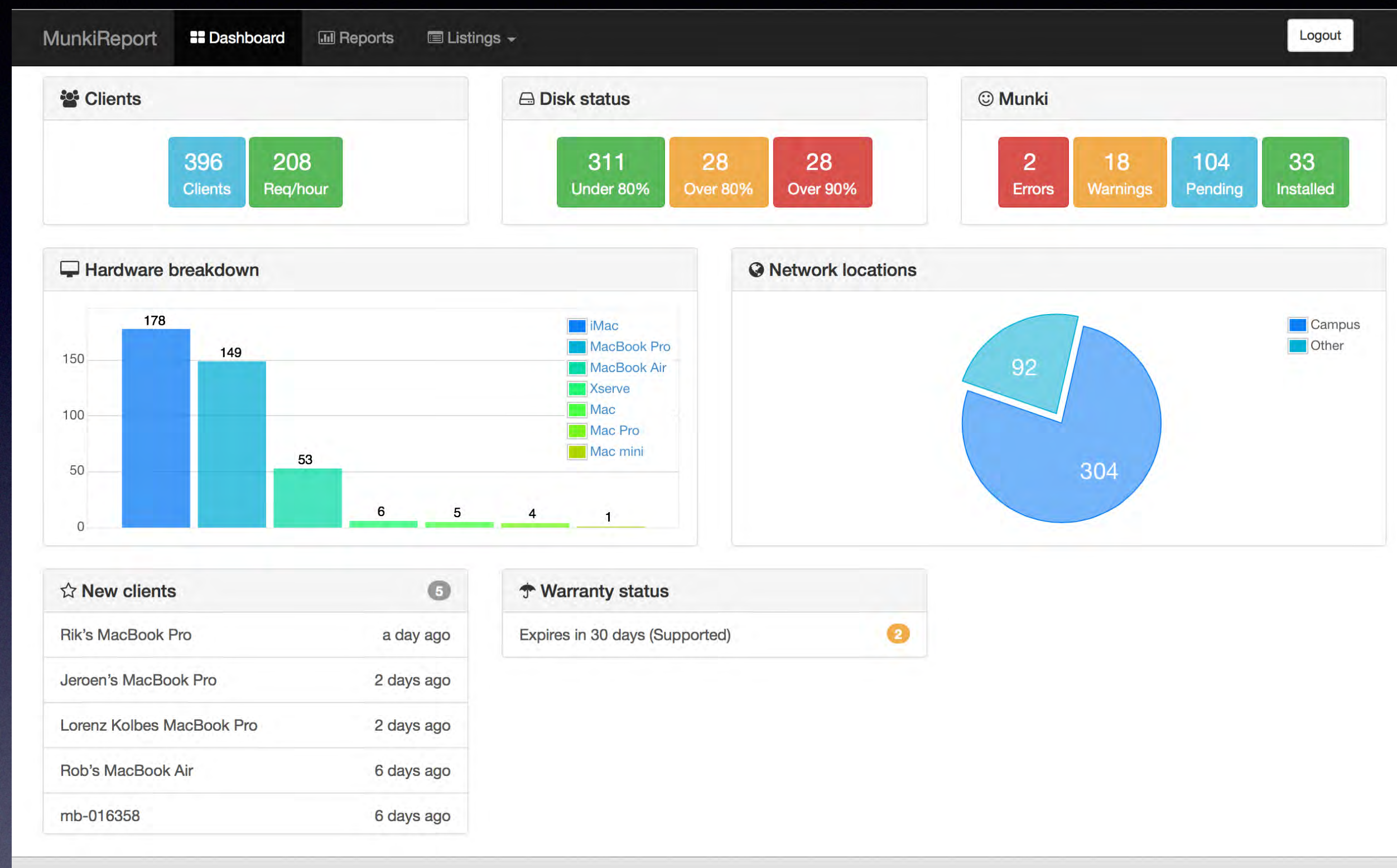
Munki: 50 Errors, 250 Warnings

Free Disk Space: 411 < 5GB, 801 < 10GB, 5744 10GB +

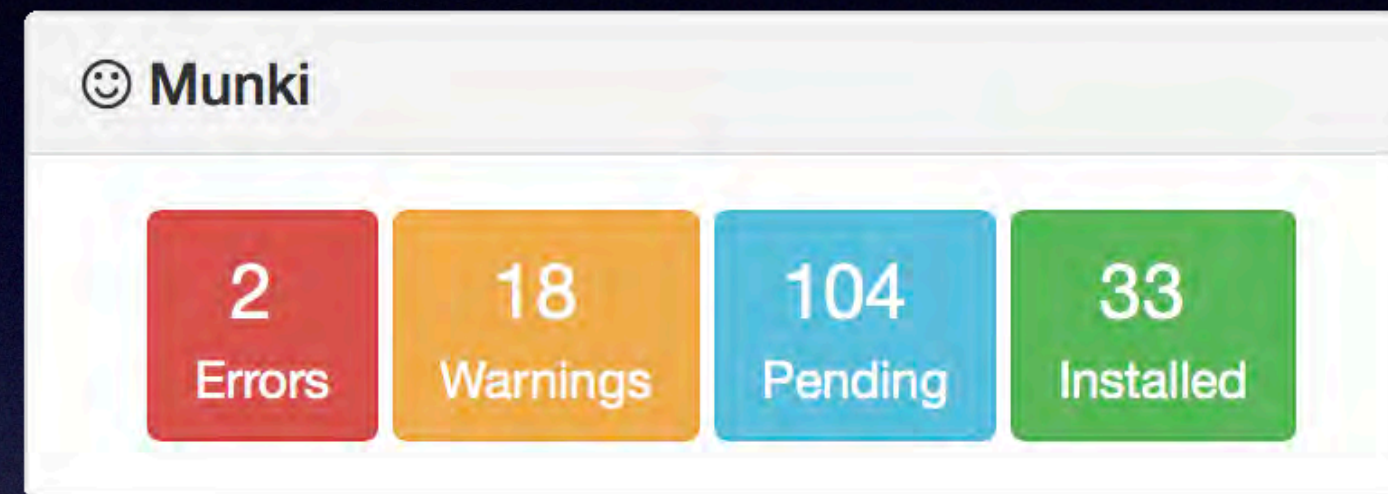
Uptime: 3131 (7 Days +), 1849 (< 7 Days), 1573 (< 1 Day)

MunkiReport Version 3.4.0.3657

precursor.ca/slides



precursor.ca/slides



precursor.ca/slides



<https://github.com/precursorca/munki-stats-widget>

Thanks to Arjen van Bochoven!

precursor.ca/slides



munkireport.example.com/index.php?//show/dashboard/stats

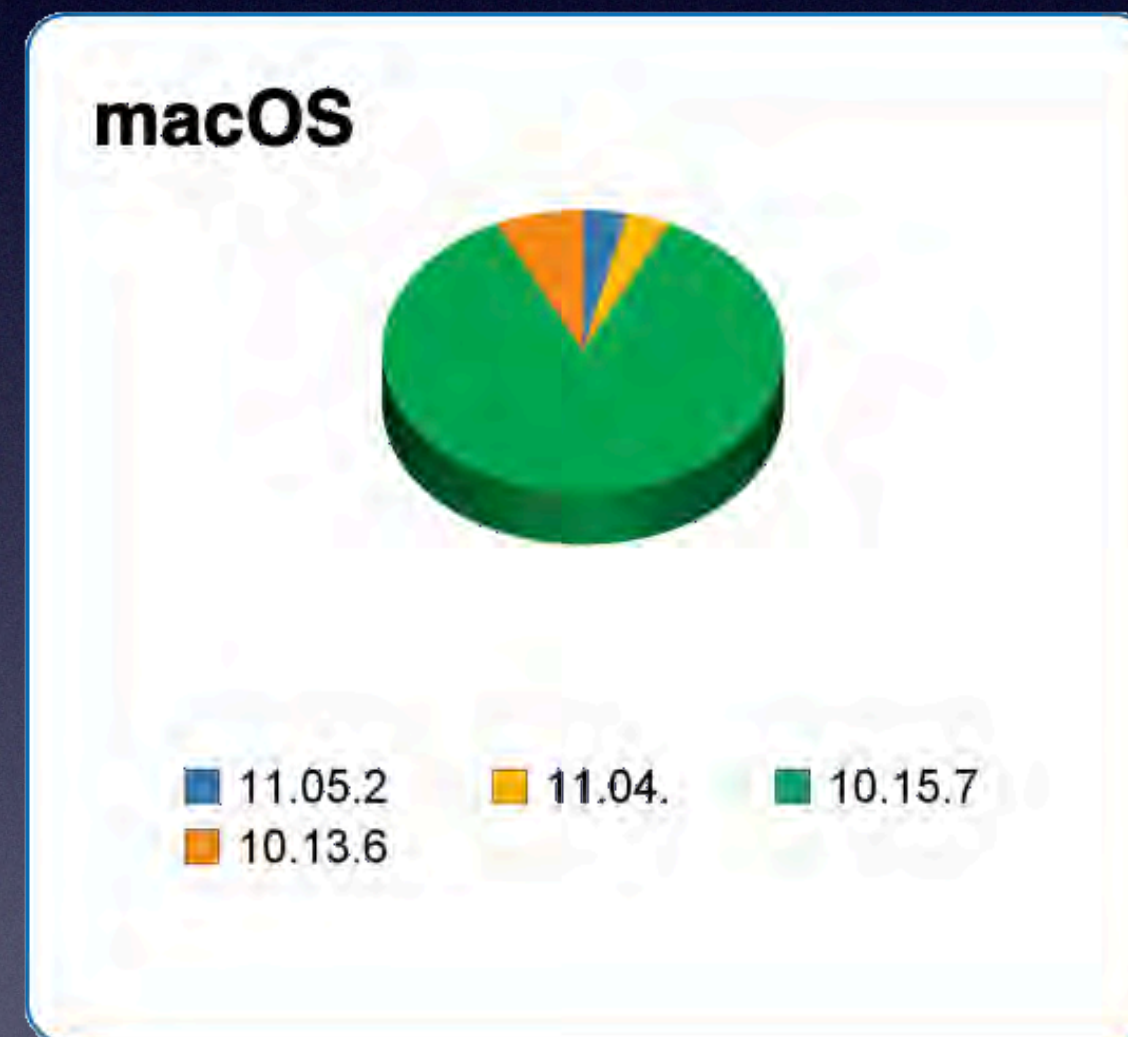


Display as a FileMaker Buttons

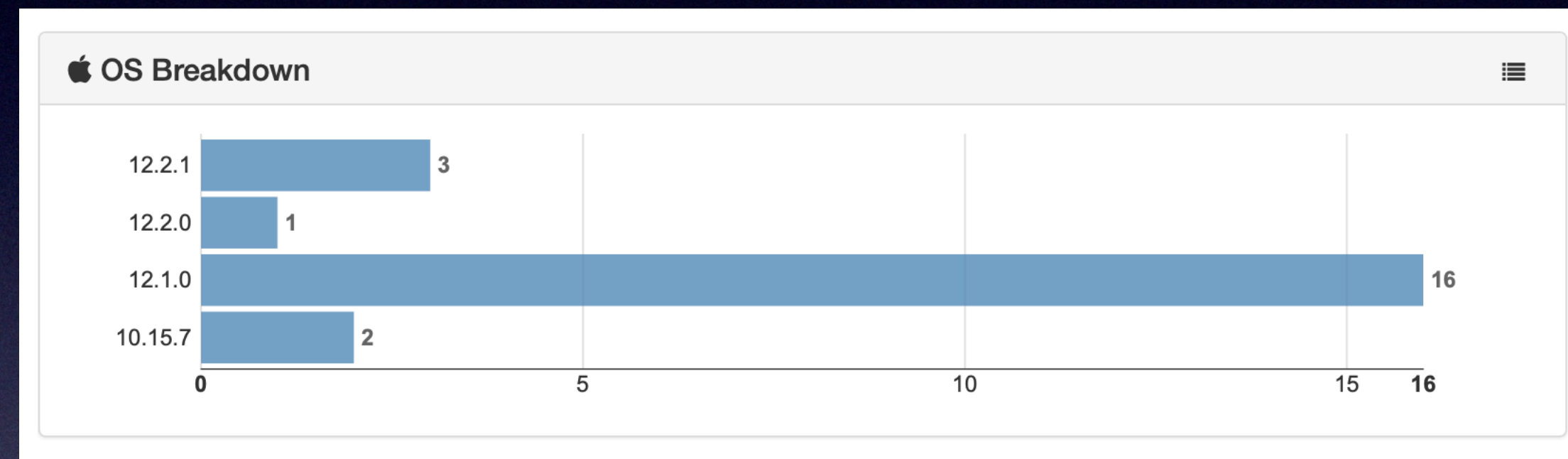
precursor.ca/slides



macOS

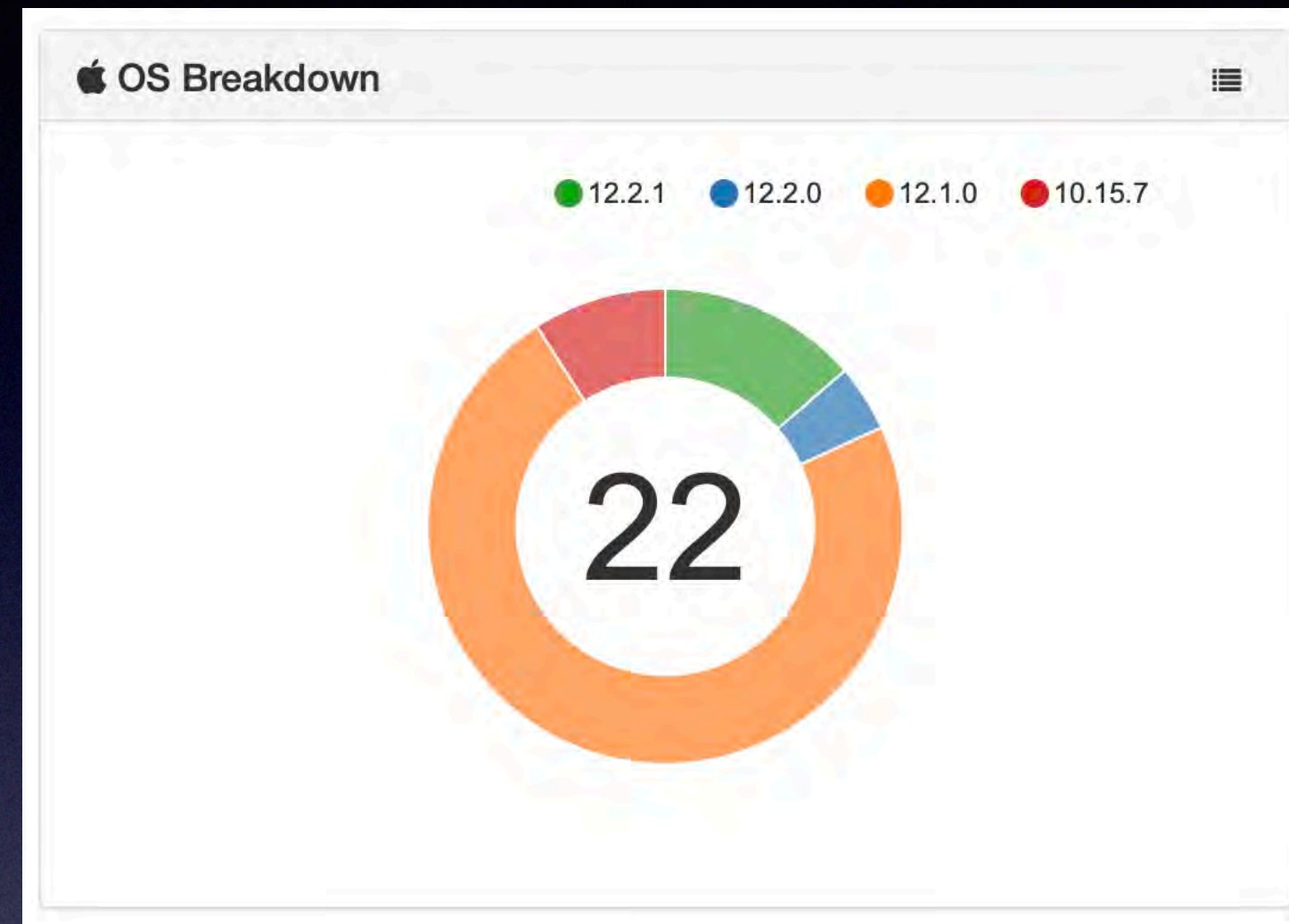


precursor.ca/slides



Munkireport OS Breakdown

precursor.ca/slides



Munki report OS Breakdown Donut

https://github.com/precursorca/os_donut_widget

precursor.ca/slides



`munkireport.example.com/index.php?//module/machine/os`



Display as a FileMaker Chart

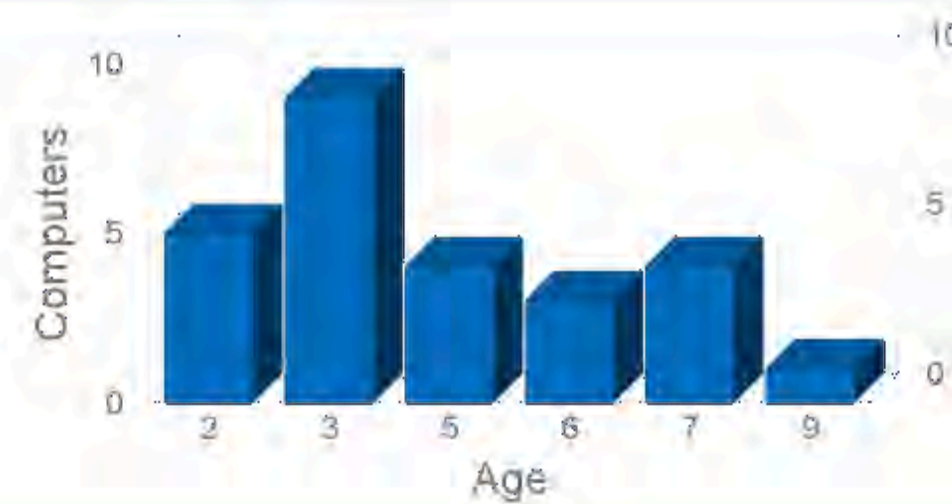
precursor.ca/slides



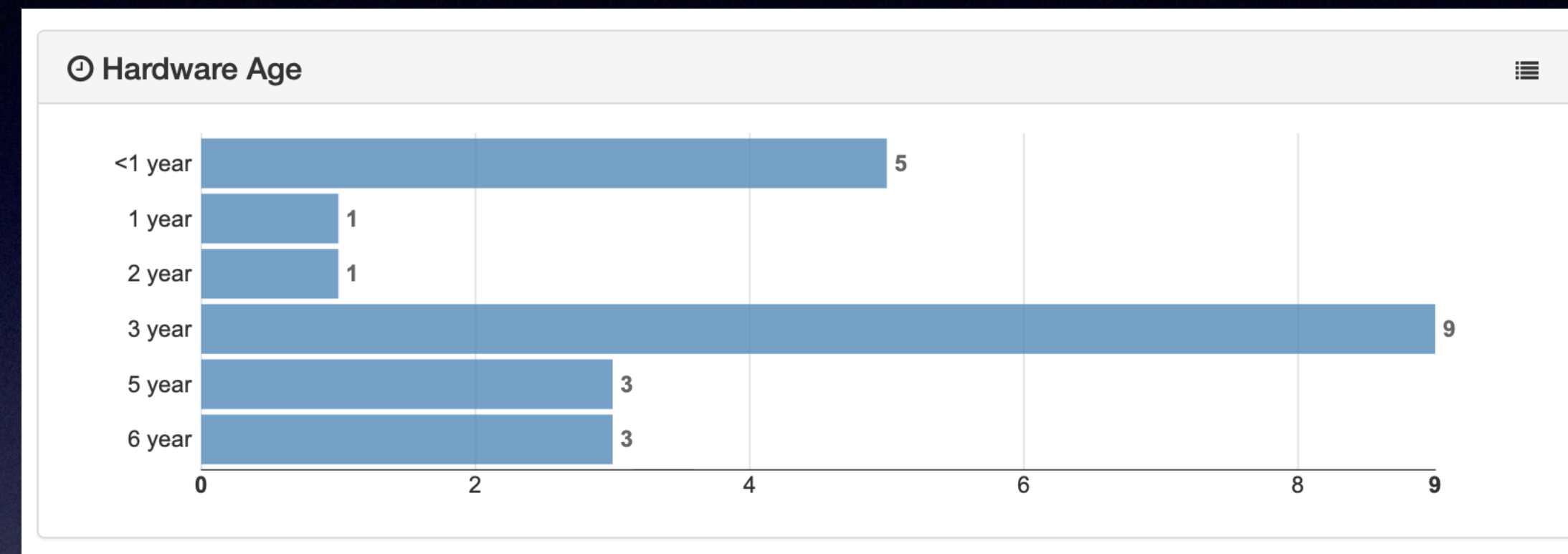
Asset Age

Asset Age

Suggest replacement budget for
this year:
\$24,000.00



precursor.ca/slides



Munkireport Hardware Age

precursor.ca/slides

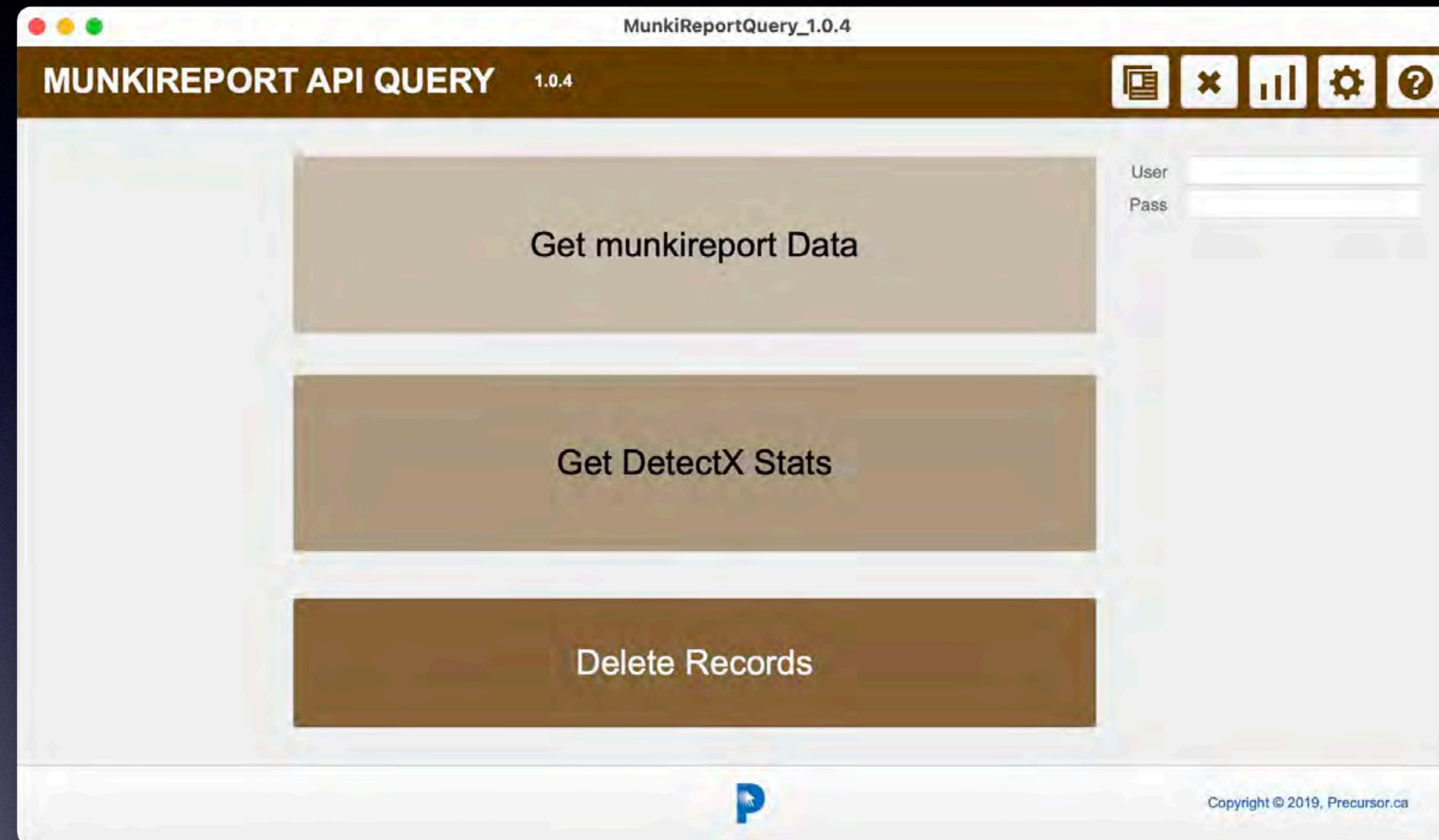


munkireport.example.com/index.php?//module/warranty/age



Display as a FileMaker Chart

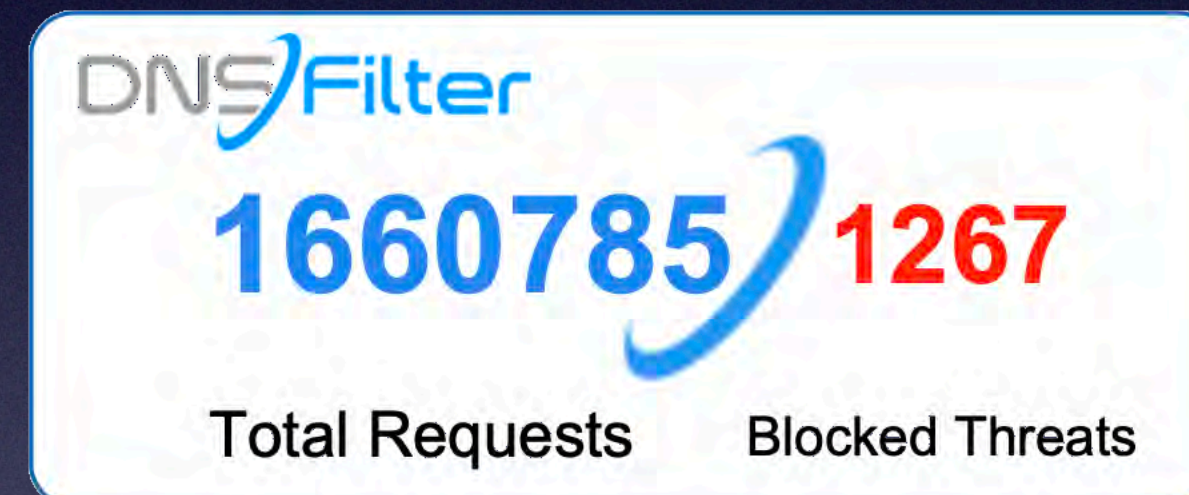
precursor.ca/slides



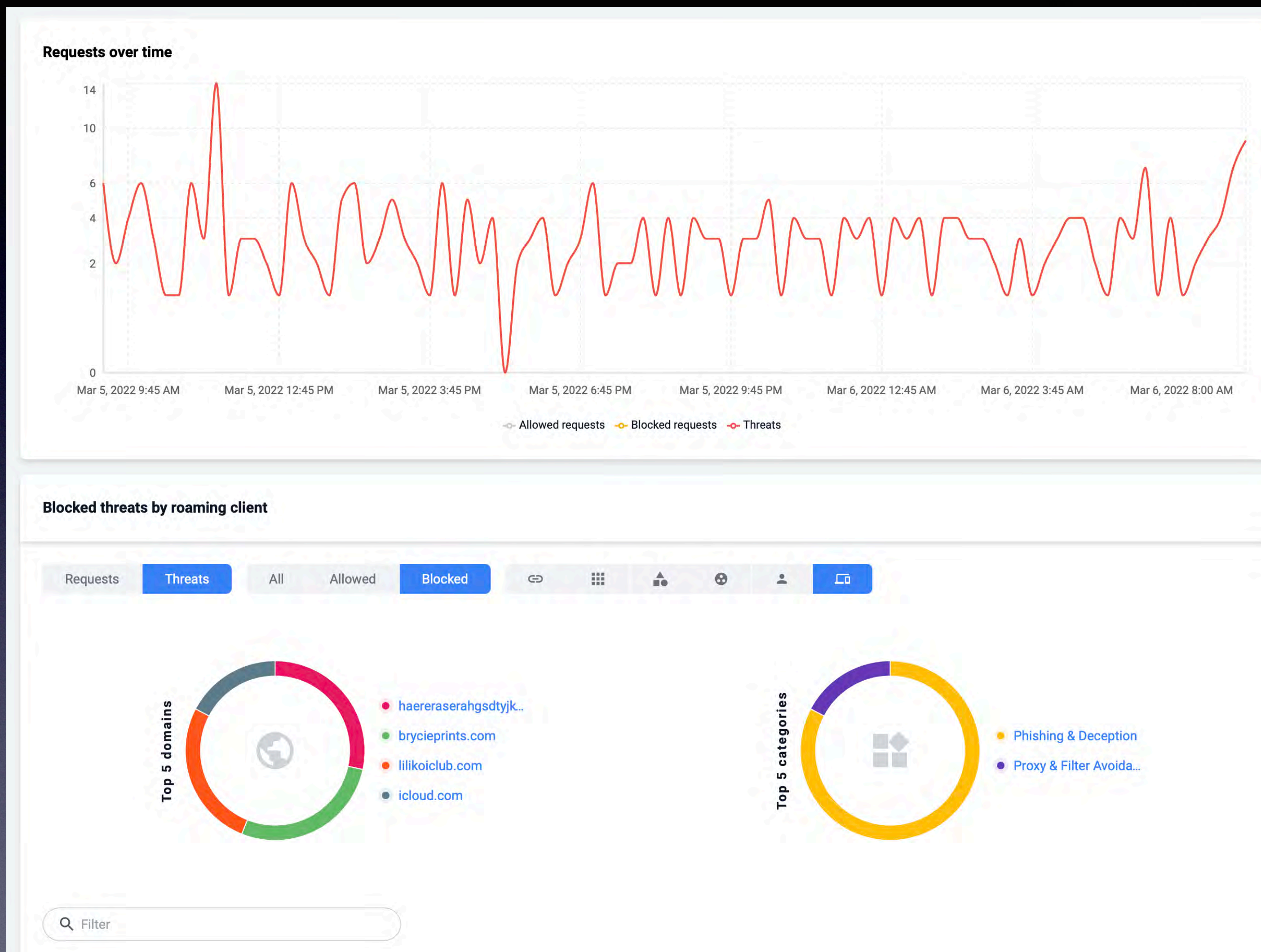
precursor.ca/slides



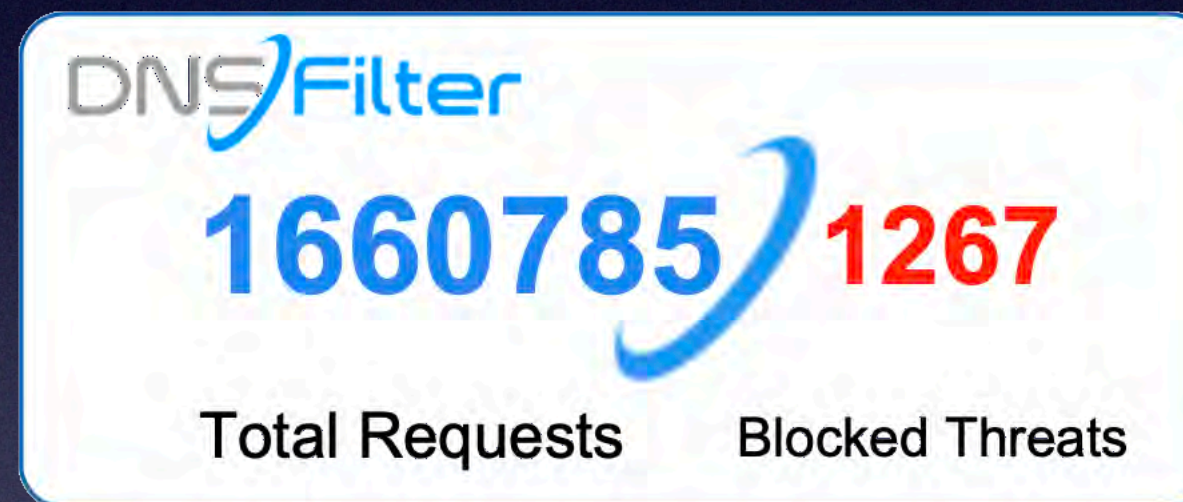
DNSFilter



precursor.ca/slides



precursor.ca/slides



precursor.ca/slides



The screenshot shows the SwaggerHub interface for the DNSFilter API (version 1.0.1). The page includes a navigation menu on the left with sections like 'AgentLocalUser', 'Auth0', and 'Authentication'. The main content area displays the API title 'DNSFilter API' and version '1.0.1'. A note states: 'NOTE: Our API is fairly stable, but we are continuing to make fixes before we finalize our version 1 API. We will do our best to communicate any breaking changes to the API before they happen.' Below this, it mentions the API uses standard HTTP status codes and provides a JSON response format for errors: `{ "error": "Not Authorized" }`. A code editor in the center shows the Swagger JSON specification for the API, including the title, description, and error handling details.

https://app.swaggerhub.com/apis/DNSFilter/dns-filter_api/1.0.1

precursor.ca/slides



- 1) Obtain a key for authentication**
- 2) Get the Organization IDs (clients)**
- 3) Set a time period (30 days back)**
- 4) Get total number of requests (aggregate 30 buckets)**
- 5) Get total number of blocked threats (aggregate 30 buckets)**

precursor.ca/slides



Obtaining a Token

Authentication tokens can be obtained from Auth0 using their [Authenticate API](#) `/oauth/token` endpoint with the following parameters:

- `client_id` : `zJ1WJHavuUFx89cConwlipxo0c2J3TVQ`
- `realm` : `Username-Password-Authentication`
- `audience` : `https://dnsfilter.auth0.com/mfa/`
- `grant_type` : `http://auth0.com/oauth/grant-type/password-realm`
- `scope` : `enroll read:authenticators remove:authenticators
offline_access openid picture name email`
- `username` : `<your username>`
- `password` : `<your password>`



Networks ^	
GET	/v1/networks
POST	/v1/networks
GET	/v1/networks/all
GET	/v1/networks/msp
GET	/v1/networks/msp/all
GET	/v1/networks/lookup
GET	/v1/networks/{id}
PUT	/v1/networks/{id}
DELETE	/v1/networks/{id}
POST	/v1/networks/{id}/secret_key
DELETE	/v1/networks/{id}/secret_key
PATCH	/v1/networks/{id}/secret_key

Organizations ^	
GET	/v1/organizations
POST	/v1/organizations
GET	/v1/organizations/all
GET	/v1/organizations/{id}
PUT	/v1/organizations/{id}
DELETE	/v1/organizations/{id}
POST	/v1/organizations/{id}/cancel
POST	/v1/organizations/promote_to_msp/

<https://api.dnsfilter.com/v1/clients>

<https://api.dnsfilter.com/v1/organizations>

precursor.ca/slides



GET

<https://api.dnsfilter.com/v1/organizations>

```
{
  "data": [
    {
      "id": "string",
      "type": "string",
      "attributes": {
        "id": 0,
        "name": "string",
        "stripe_customer_id": "string",
        "trial_days": 0,
        "update_payment": true,
        "billing_contact_name": "string",
        "billing_contact_phone": "string",
        "billing_contact_email": "string",
        "address": "string",
      }
    }
  ]
}
```

precursor.ca/slides




Loop

New Record/Request

```
Set Field [DNSFilterNetworks::networkNAME ;
```

```
JSONGetElement ( DNSFilterNetworks::Results ; "data[" & $i & "]attributes.name" ) ]
```

```
Set Field [DNSFilterNetworks::networkID ;
```

```
JSONGetElement ( DNSFilterNetworks::Results ; "data[" & $i & "]id" ) ] 
```

precursor.ca/slides



```
"https://api.dnsfilter.com/v1/traffic_reports/total_requests?  
organization_ids="  
& DNSFilterStats::ORGANIZATION_ID  
& "&from="  
& $$DNSFilterDATE  
& "T01:01:01" & "\""
```

precursor.ca/slides




bucket_size

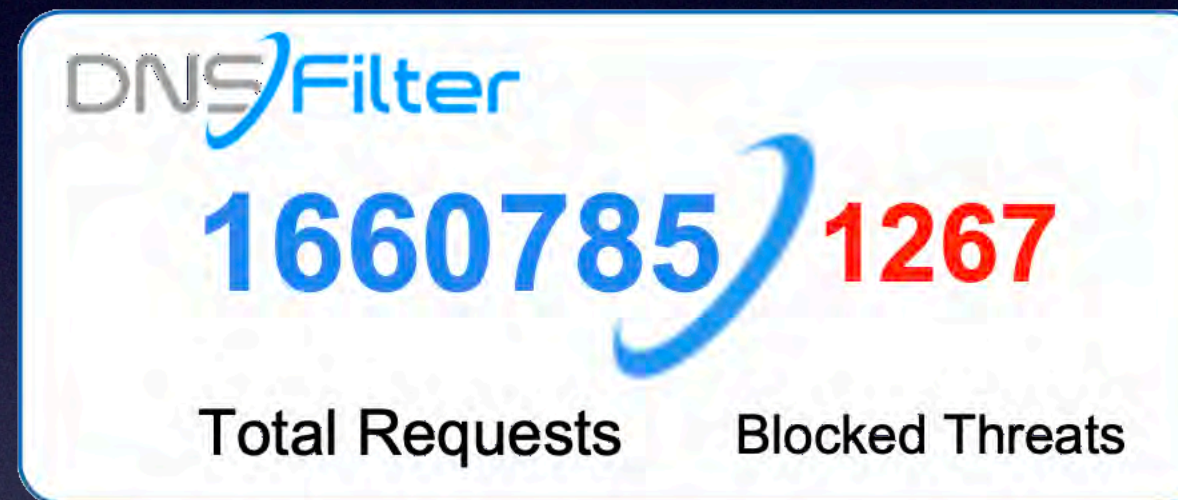
string

(query)

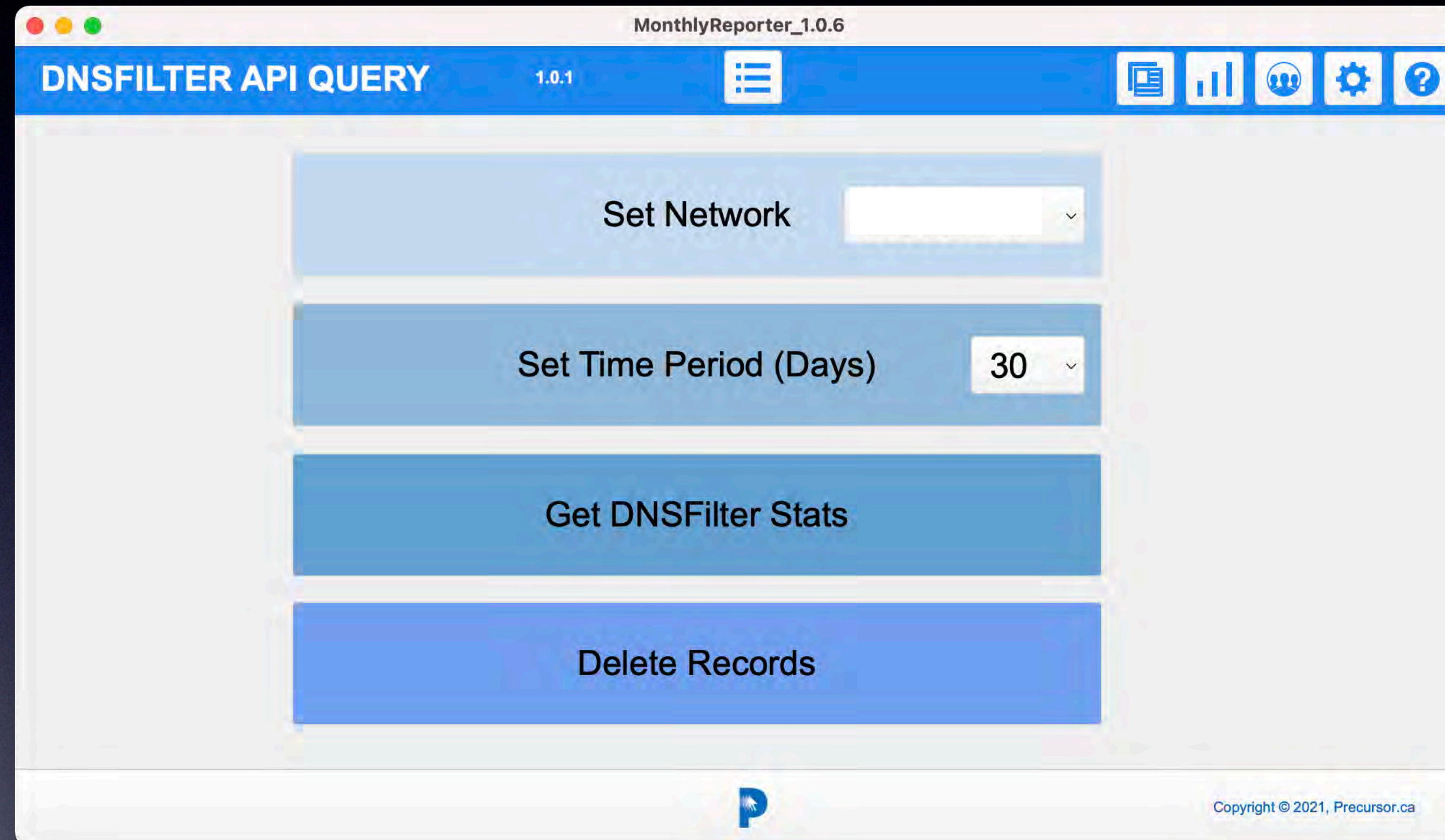
- Desired bucket size, possible values: *auto*, *15min*, *1day*, defaults to *auto*
- *Auto:*
- If report from/to range is less than 24 hours (ex.: "Today" and "Yesterday"), bucket size is fifteen minutes
- If report from/to range is greater than 24 hours (ex.: "Last 7 days" and "Last 14 days"), bucket size is one day



```
If [ $BucketCount > 0 ]  
  Loop  
    Set Variable [ $Requests ; Value: $Requests + JSONGetElement ( DNSFilterStats::Results ; "data.values[" & $i & "].total" ) ]  
    Set Variable [ $i ; Value: $i + 1 ]  
    Exit Loop If [ $i ≥ $BucketCount ]   
  End Loop  
Set Field [ DNSFilterStats::TotalRequests ; $Requests ]
```

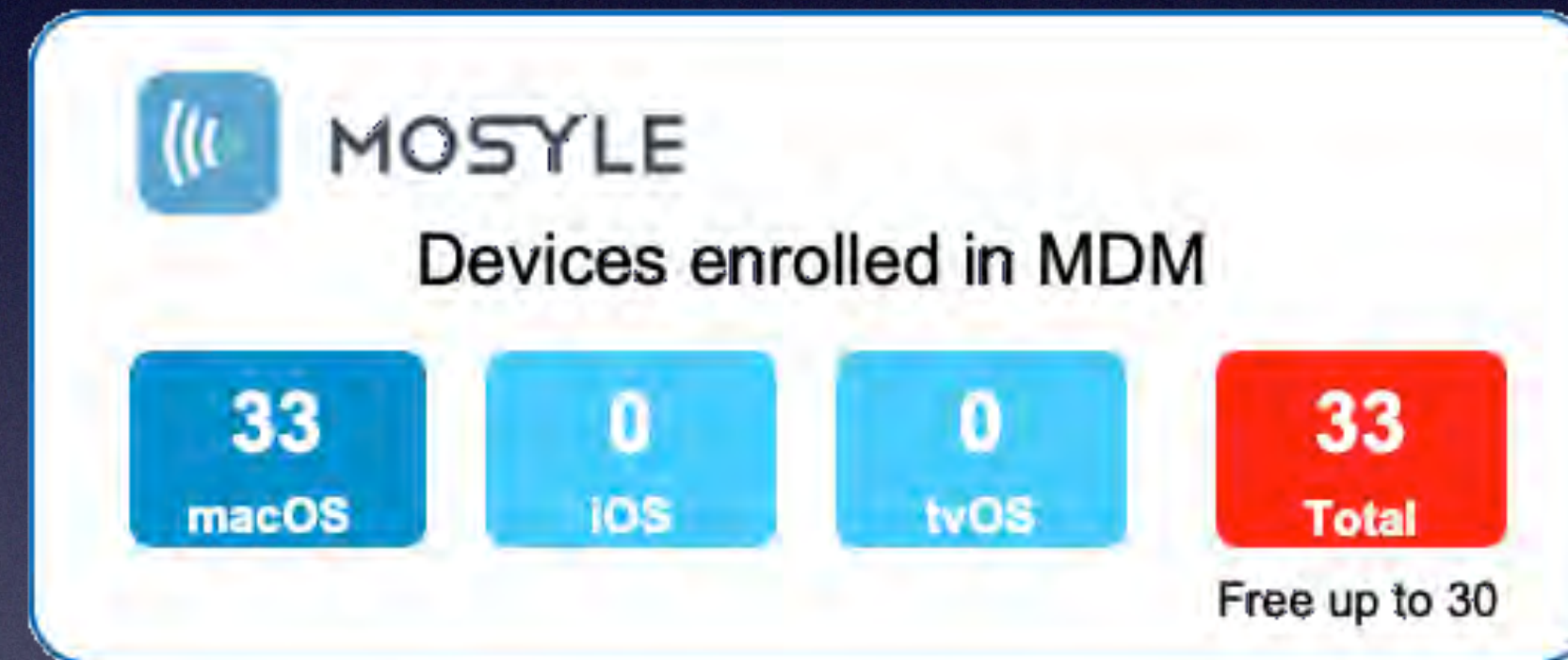
precursor.ca/slides



precursor.ca/slides



Mosyle



precursor.ca/slides



mybusiness.mosyle.com

Alex Narvey

Settings Logout

Your Trial ends in 300 day(s). [Click here to upgrade now.](#)

Apple Basic Setup

Users & Groups

- Administrators
- End Users
- User Groups
- Device Assignment
- Users Photos

Integrations

- Mosyle API Integration

Preferences

Mosyle API Integration

Configure and use the access token to make requests through the API.

Deactivate Add new token

To learn more about our API, [check our documentation.](#)

Search by name...

MOSYLEapi	<input checked="" type="checkbox"/>
-----------	-------------------------------------

precursor.ca/slides



mybusiness.mosyle.com

Precursor Systems | Apple Certified Support in Winnipeg, MB

Precursor Systems

Alex Narvey

Dashboard Organization Management Security Support Settings Logout

Your Trial ends in 300 day(s). Click here to upgrade now.

Apple Basic Setup

Users & Groups

Administrators

End Users

User Groups

Device Assignment

Users Photos

Integrations

Activate New Integration

Mosyle API Integration

Preferences

Mosyle API Integration

Configure and use the access token to make requests through the API.

Deactivate Add new token

To learn more about our API, [check our documentation](#).

Search by name...

MOSYLEapi

API Information

Profile Name
MOSYLEapi

Workspace
Precursor Systems

Access Token
*****7c5f

Revoke and Renew

Require User Credentials
No

Access Method
Public

Allowed Endpoints
Allow all current and future endpoints

Edit

Remove token

precursor.ca/slides



API DOCS

NB. THIS IS JUST A COPY AND PASTE OF VARIOUS MODAL DIALOGS FROM THE MOSYLE BUSINESS INTERFACE AND AS SUCH MAY NOT BE UP-TO-DATE.

First of all, to use the Mosyle API you need to enable this feature in the API profile page (Organization > API Integration > enable the profile)

Once enabled you will be able to see your access token and make requests to the endpoint "http://businessapi.mosyle.com/v1", every request will have some required parameters as well as optional parameters. You can send these parameters like key => value using the form-data format or a JSON format. You can specify this in the header of your request using the key **Content-Type** and the value **application/x-www-form-urlencoded** or **application/json**. Check the list below with the required parameters:

Key	Type	Required	Description
Authorization	string	<i>Required only if checkbox 'Require an user credential' is checked</i>	The string 'Basic ' followed by the base64 encoded 'username: password'
accesstoken	string	<i>Required</i>	Your token in API profile page
Content-Type	string	<i>Required</i>	application/x-www-form-urlencoded
			application/json

For more examples with other languages and to test the API, [click here to download the API sample file](#). (Compatible with Postman and Insomnia).

Attention: You must **change the environment variables** and provide your **API Token Access, Email** and **Password** when you open the sample archive on Postman or Insomnia.

The following code snippet will be used in all of your requests.

```
<?php
//PHP EXAMPLE
$token = 'your token';
$username = 'email that will be authenticated in mosyle business';
$password = '*****';
$headers = array();
$headers[] = 'Content-Type: application/x-www-form-urlencoded';
$headers[] = 'Authorization: Basic '.base64_encode($username.':'.$password);
$headers[] = 'accesstoken: '.$token;
```




DEVICES

Results

status CurrentConsoleManagedUser deviceudid

idusermosyle AutoSetupAdminAccounts

ActiveManagedUsers

device_type	<input type="text"/>	isActivationLockEnabled	<input type="text"/>	carrier	<input type="text"/>	appleTVid	<input type="text"/>
device_name	<input type="text"/>	isDeviceLocatorServiceEnab	<input type="text"/>	roaming_enabled	<input type="text"/>	asset_tag	<input type="text"/>
model_name	<input type="text"/>	isDoNotDisturbInEffect	<input type="text"/>	isroaming	<input type="text"/>	status_login	<input type="text"/>
device_model	<input type="text"/>	isCloudBackupEnabled	<input type="text"/>	imei	<input type="text"/>	date_profiles_info	<input type="text"/>
serial_number	<input type="text"/>	IsNetworkTethered	<input type="text"/>	meid	<input type="text"/>	date_lastlogin	<input type="text"/>
userCODE	<input type="text"/>	is_mdm_lostmode_enabled	<input type="text"/>	iTunesStoreHash	<input type="text"/>	date_checkin	<input type="text"/>
os	<input type="text"/>	is_muted	<input type="text"/>	iTunesStoreAccountIsActive	<input type="text"/>	date_enroll	<input type="text"/>
osversion	<input type="text"/>	date_muted	<input type="text"/>	total_disk	<input type="text"/>	date_checkout	<input type="text"/>
BuildVersion	<input type="text"/>	activation_bypass	<input type="text"/>	available_disk	<input type="text"/>	date_kinfo	<input type="text"/>
needosupdate	<input type="text"/>	date_media_info	<input type="text"/>	percent_disk	<input type="text"/>	date_info	<input type="text"/>
productkeyupdate	<input type="text"/>	is_supervised	<input type="text"/>	ManagementStatus	<input type="text"/>	date_app_info	<input type="text"/>
battery	<input type="text"/>	SIPenabled	<input type="text"/>	OSUpdateStatus	<input type="text"/>	date_last_beat	<input type="text"/>
HostName	<input type="text"/>	is_deleted	<input type="text"/>	AvailableOSUpdates	<input type="text"/>	date_last_push	<input type="text"/>
LocalHostName	<input type="text"/>	OSUpdateSettings	<input type="text"/>			date_printers	<input type="text"/>
ethernet_mac_address	<input type="text"/>					LastCloudBackupDate	<input type="text"/>
wifi_mac_address	<input type="text"/>						
bluetooth_mac_address	<input type="text"/>	tags	<input type="text"/>				

precursor.ca/slides



```
"https://businessapi.mosyle.com/v1/devices"
```

```
//The cURL details
```

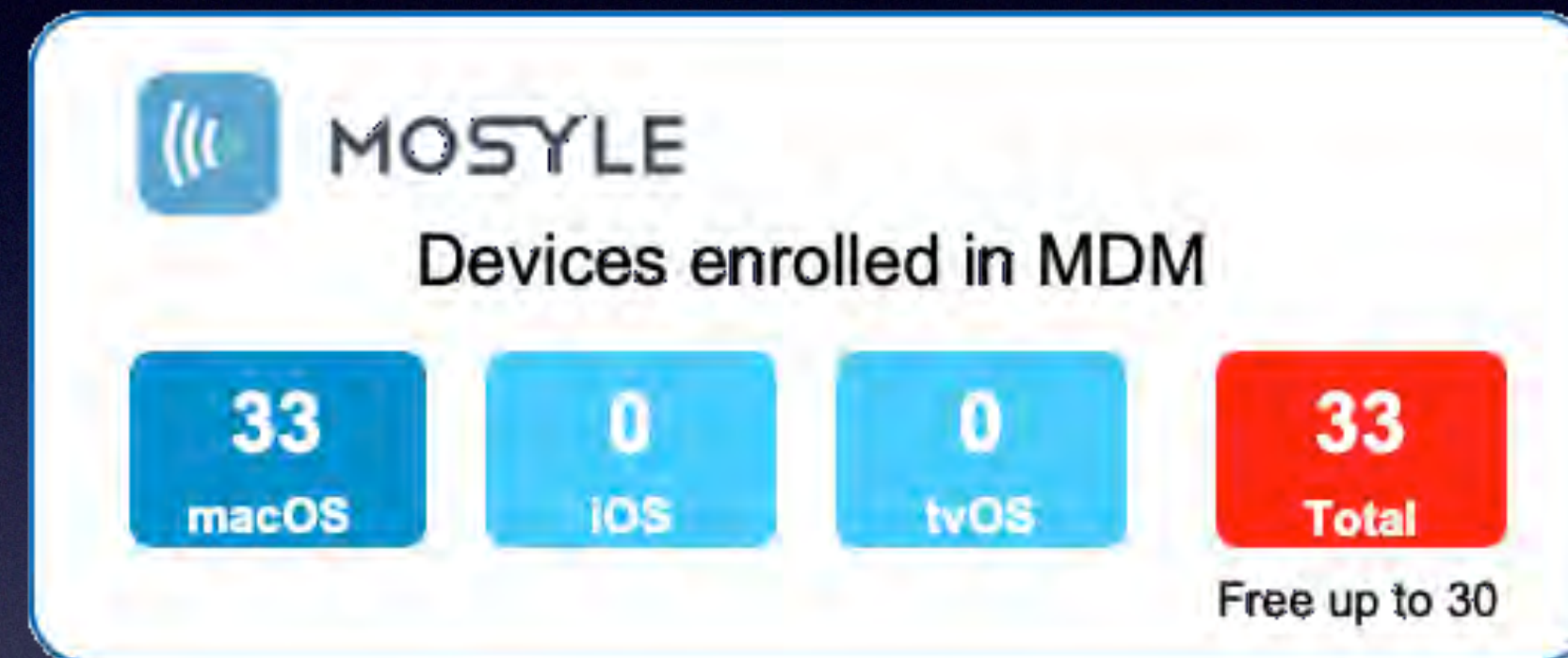
```
" -d @$MacDeviceJSON"
```

```
//" -d {"operation": "list", "options": { "os": "mac" } }"
```

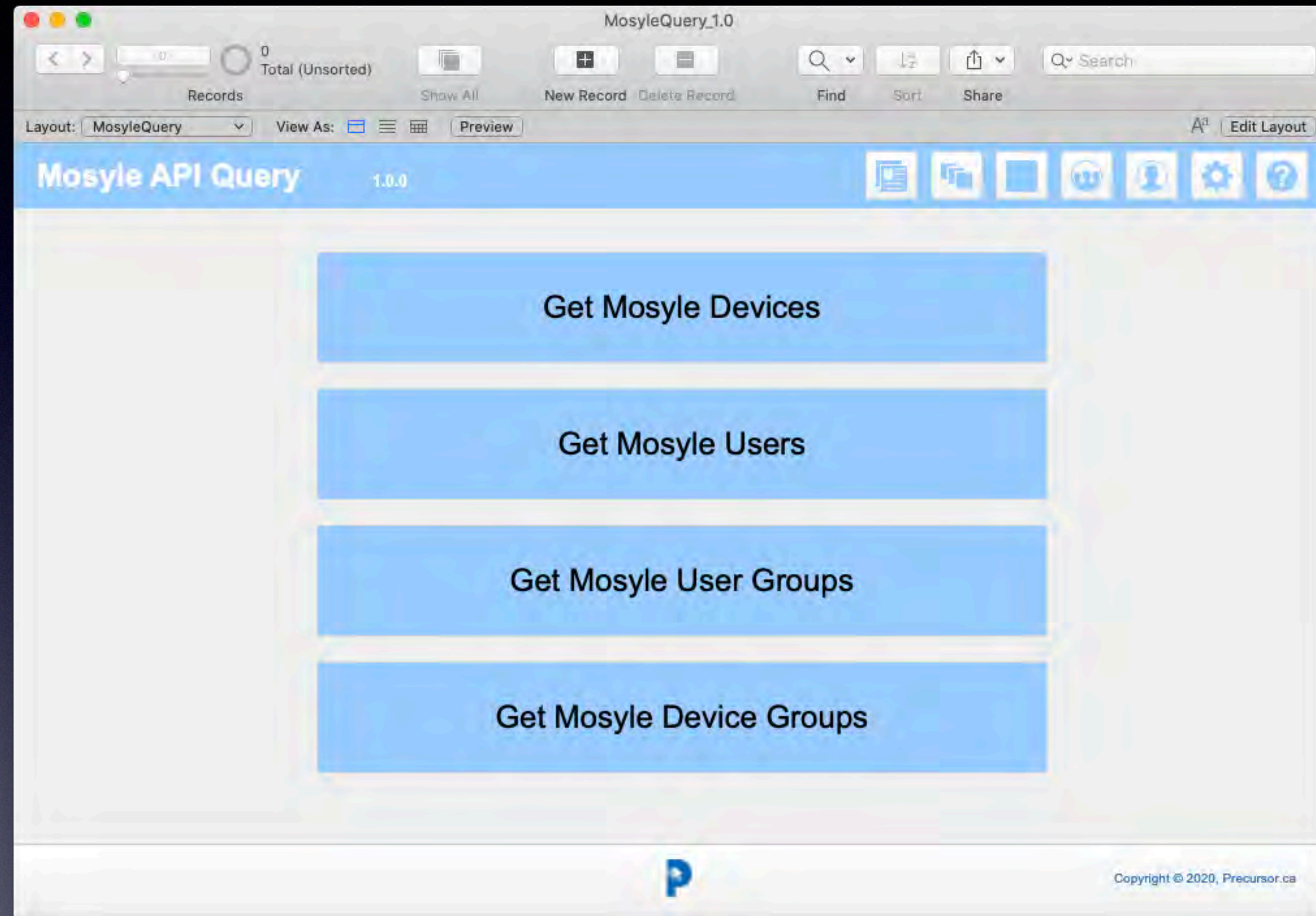
```
//" -d {"operation": "list", "options": { "os": "ios" } }"
```

```
//" -d {"operation": "list", "options": { "os": "tvos" } }"
```

precursor.ca/slides



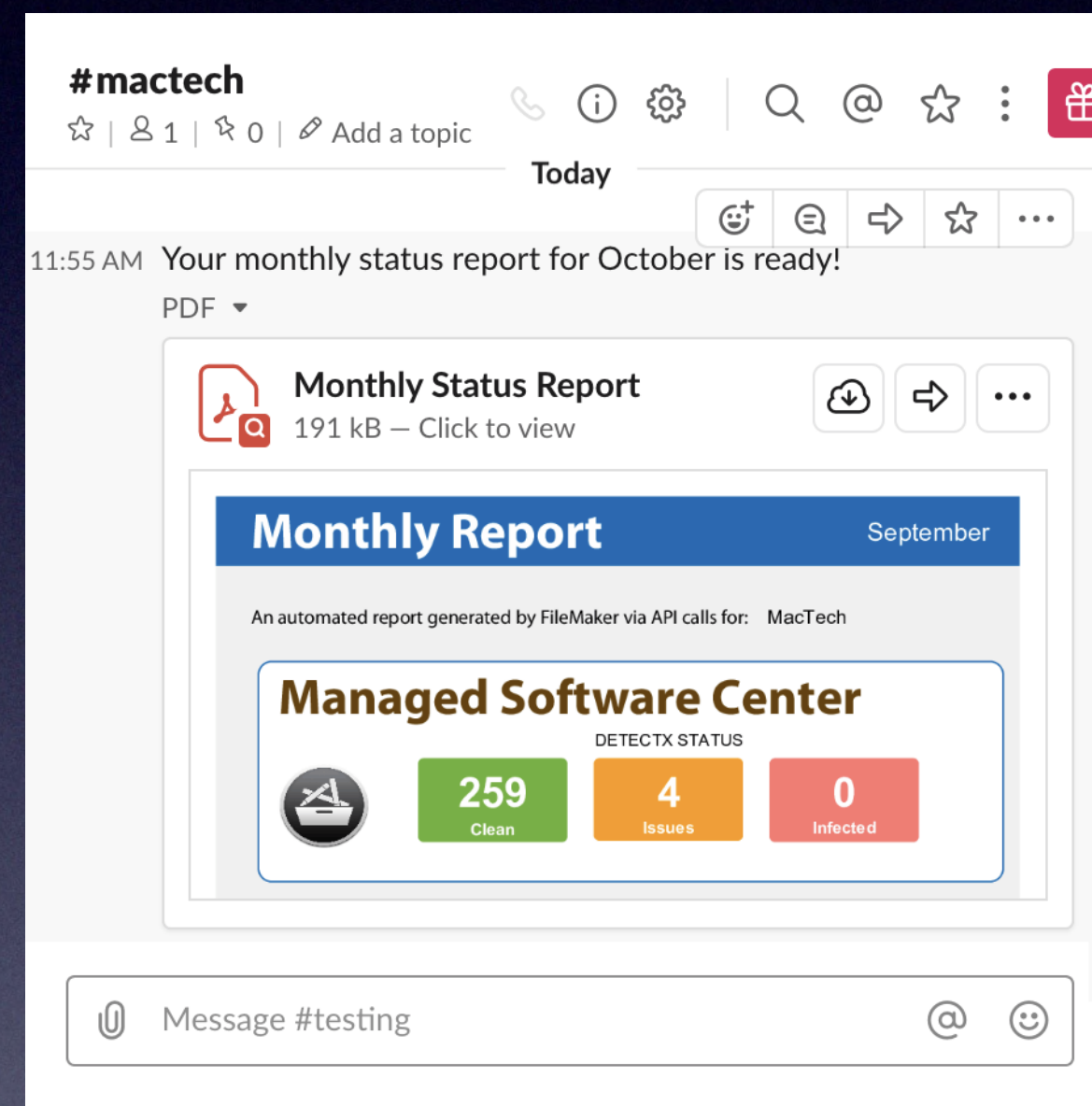
precursor.ca/slides



precursor.ca/slides



Slack



precursor.ca/slides



Web API

Events API

Conversations API

Real Time Messaging API

Methods

Object Types

Event Types

SCIM API

Audit Logs API

<https://api.slack.com>

precursor.ca/slides



Method URL:	<code>https://slack.com/api/chat.postMessage</code>						
Preferred HTTP method:	POST						
Accepted content types:	<code>application/x-www-form-urlencoded</code> , <code>application/json</code>						
Rate limiting:	Special						
Works with:	<table><thead><tr><th>Token type</th><th>Required scope(s)</th></tr></thead><tbody><tr><td>bot</td><td>bot</td></tr><tr><td>user</td><td><code>chat:write:user</code> <code>chat:write:bot</code></td></tr></tbody></table>	Token type	Required scope(s)	bot	bot	user	<code>chat:write:user</code> <code>chat:write:bot</code>
Token type	Required scope(s)						
bot	bot						
user	<code>chat:write:user</code> <code>chat:write:bot</code>						

<https://api.slack.com/methods/chat.postMessage>

precursor.ca/slides



Argument	Example	Required	Description
<code>token</code>	<code>xxxx-xxxxxxxx-xxxx</code>	Required	Authentication token bearing required scopes.
<code>channel</code>	<code>C1234567890</code>	Required	Channel, private group, or IM channel to send message to. Can be an encoded ID, or a name. See below for more details.
<code>text</code>	<code>Hello world</code>	Required	Text of the message to send. See below for an explanation of formatting . This field is usually required, unless you're providing only <code>attachments</code> instead. Provide no more than 40,000 characters or risk truncation .

precursor.ca/slides



```
CURL EXAMPLE </>
curl -X POST -H 'Authorization: Bearer xoxb-1234-56789abcdefghijklmnop'
-H 'Content-type: application/json' \
--data '{"channel":"C061EG9SL","text":"I hope the tour went well, Mr'
https://slack.com/api/chat.postMessage
```

https://api.slack.com/web#posting_json

precursor.ca/slides



Authentication

Authenticate your Web API requests by providing a [bearer token](#), which identifies a single user, bot user, or workspace-application relationship.

[Register your application](#) with Slack to obtain credentials for use with our [OAuth 2.0](#) implementation, which allows you to negotiate tokens on behalf of users and workspaces.

`https://api.slack.com/web#posting_json`

`precursor.ca/slides`



Your Apps

Create New App

i If your app is (or will be) listed in the Slack App Directory, please review our new [Slack App Directory Agreement](#). These terms are in addition to the existing Developer Policy, API TOS, and Brand Guidelines.

By keeping your app in the App Directory or review process, you're confirming your agreement to the Slack App Directory Agreement and to providing additional information for security review, if requested. If you don't agree with this Agreement, please send an email to feedback@slack.com, and we'll remove your app from the App Directory or the review process.

I Agree

<https://api.slack.com/apps>

precursor.ca/slides



Webhook URLs for Your Workspace

To dispatch messages with your webhook URL, send your [message](#) in JSON as the body of an `application/json` POST request.

Add this webhook to your workspace below to activate this curl example.

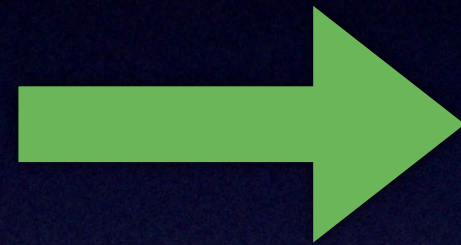
Sample curl request to post to a channel:

```
curl -X POST -H 'Content-type: application/json' --data '{"text":"Hello, World!"}'  
https://hooks.slack.com/services/T0508V0QC/BN3HC102F/00ZcyMkcypx309hsEq8Dhze  
4
```

Copy

<https://api.slack.com/incoming-webhooks>

precursor.ca/slides



Add features and functionality ✓

Choose and configure the tools you'll need to create your app (or review all our [documentation](#)).

- Incoming Webhooks**
Post messages from external sources into Slack.
- Interactive Components**
Add buttons to your app's messages, and create an interactive experience for users.
- Slash Commands**
Allow users to perform app actions by typing commands in Slack.
- Event Subscriptions**
Make it easy for your app to respond to activity in Slack.
- Bots**
Add a bot to allow users to exchange messages with your app.
- Permissions**
Configure permissions to allow your app to interact with the Slack API.

<https://api.slack.com/methods/chat.postMessage>

precursor.ca/slides



✓ Incoming Webhooks

Post messages from external sources into Slack.

App Credentials

These credentials allow your app to access the Slack API. They are secret. Please don't share your app credentials with anyone, include them in public code repositories, or store them in insecure ways.

App ID

XXXXXXXXXXXX

Date of App Creation

August 25, 2019

Client ID

50CXXXXXXXXXXXXX.72XXXXXXXXXXXX

Client Secret

.....

You'll need to send this secret along with your client ID when making your [oauth.access](#) request.

Signing Secret

.....


Slack signs the requests we send you using this secret. Confirm that each request comes from Slack by verifying its unique signature.

Verification Token

LUXXXXXXXXXXXXX

This deprecated Verification Token can still be used to verify that requests come from Slack, but we strongly recommend using the above, more secure, signing secret instead.



 **Permissions**
Configure permissions to allow your app to interact with the Slack API.

OAuth Tokens & Redirect URLs

Tokens for Your Workspace

These tokens were automatically generated when you installed the app to your team. You can use these to authenticate your app. [Learn more.](#)

OAuth Access Token

xoxp-500[REDACTED]-5[REDACTED]-7[REDACTED] 3-c Copy

[Reinstall App](#)

<https://api.slack.com/docs/oauth>

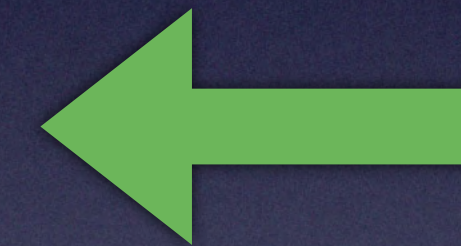
precursor.ca/slides



Add features and functionality ✓

Choose and configure the tools you'll need to create your app (or review all our [documentation](#)).


✓ Incoming Webhooks Post messages from external sources into Slack.	Interactive Components Add buttons to your app's messages, and create an interactive experience for users.
Slash Commands Allow users to perform app actions by typing commands in Slack.	Event Subscriptions Make it easy for your app to respond to activity in Slack.
Bots Add a bot to allow users to exchange messages with your app.	✓ Permissions Configure permissions to allow your app to interact with the Slack API.



`https://api.slack.com/methods/chat.postMessage`

precursor.ca/slides





 **Permissions**
Configure permissions to allow your app to interact with the Slack API.

Select Permission Scopes

Add permission by scope or API method... ▼

CONVERSATIONS

Send messages as user 
chat:write:user

Post to specific channels in Slack 
incoming-webhook

<https://api.slack.com/docs/oauth-scopes>

precursor.ca/slides



SLACK API POST Message 1.0.0

Slack Channel:

Title:

Comment:

Headers

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 225
Connection: keep-alive
Date: Tue, 17 Sep 2019 15:17:46 GMT
```

Trace

```
== Info: Trying 13.249.92.165...
== Info: Connected to slack.com (13.249.92.165) port 443 (#0)
== Info: ALPN, offering http/1.1
-- Info: Cipher selection: ECDHE_RSA_AES128_GCM
```

SlackResult

```
{"ok": true, "channel": "KADMF3ADFL", "ts": "1568760460.000000", "message": {"bot_id": "A0681258287", "type": "message", "text": "Your Monthly Report is now available!", "user": "U0681258287", "ts": "1568760460.000000", "team": "T0681258287"}, "warning": "missing_charset", "response_metadata": {"warnings": ["missing_charset"]}}
```

Copyright © 2019, Precursor.ca



SLACK API POST Message 1.0.0

Slack Channel:

Title:

Comment:

Headers

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 225
Connection: keep-alive
Date: Tue, 17 Sep 2019 15:17:46 GMT
```

Trace

```
== Info: Trying 13.249.92.165...
== Info: Connected to slack.com (13.249.92.165) port 443 (#0)
== Info: ALPN, offering http/1.1
-- Info: Cipher selection: ECDHE_RSA_AES128_GCM
```

SlackResult

```
{"ok": true, "channel": "KADMF3ADFL", "ts": "1568814466.004100", "message": {"bot_id": "U01234567", "type": "message", "text": "Your Monthly Report is now available!", "user": "U01234567", "ts": "1568814466.004100", "team": "T01234567"}, "warning": "missing_charset", "response_metadata": {"warnings": ["missing_charset"]}}
```

Copyright © 2019, Precursor.ca

#mactech

☆ | 👤 1 | ✨ 0 | ✎ Add a topic

Today

Precursor Systems 12:16 PM

Your Monthly Report is now available!

precursor.ca/slides



Method URL:	<code>https://slack.com/api/files.upload</code>	
Preferred HTTP method:	POST	
Accepted content types:	<code>multipart/form-data</code> , <code>application/x-www-form-urlencoded</code>	
Rate limiting:	Tier 2	
Works with:	Token type	Required scope(s)
	bot	bot
	user	<code>files:write:user</code>

<https://api.slack.com/methods/files.upload>

precursor.ca/slides



</> Present arguments as parameters in `application/x-www-form-urlencoded` querystring or POST body. This method does not currently accept `application/json`.



```
curl -F "file=@$FilePath" -F "initial_comment=Your monthly status report is ready!"  
-F "title=Monthly Status Report" -F "filename=Monthly_Report.pdf" -F channels=CMC4LCXUJ  
-H "Authorization: Bearer xoxp-1234567890-██████████-453627128-██████████788372829f9"  
https://slack.com/api/files.upload
```




SLACK API POST file.upload 1.0.1

Slack Channel:

Title:

Comment:

FileType:

FileName:

File:

SlackResult:

```
{
  "ok": true,
  "file": {
    "id": "FNX3AJ4GZ",
    "created": 1570391902,
    "timestamp": 1570391902,
    "name": "theFile",
    "title": "MacTech_Monthly_Report_October.pdf",
    "mimetype": "application/vpdf",
    "filetype": "pdf",
    "pretty_type": "PDF",
    "user": "U0508V0R4",
    "editable": false,
    "size": 193334,
    "mode": "hosted",
    "is_external": false,
    "external_type": "",
    "is_public": true,
    "public_url_shared": false
  }
}
```

Headers:

```
HTTP/1.1 100 Continue
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 557
Connection: keep-alive
Date: Sun, 06 Oct 2019 21:19:12 GMT
Server: Apache
```

Trace:

```
== Info: Trying 99.84.174.165...
== Info: Connected to slack.com (99.84.174.165) port 443 (#0)
== Info: ALPN, offering http/1.1
-- Info: Cipher selection: ECDHE_RSA_AES128_GCM
```

Copyright © 2019, Precursor.ca



SLACK API POST file.upload 1.0.1


Slack Channel: [redacted]

Title: [redacted]

Comment: Your Monthly Report is now available!

FileType: [redacted]

FileName: [redacted]

File: 

Post files.upload

SlackResult

```
{
  "ok": true,
  "file": {
    "id": "FNX3AJ4GZ",
    "created": 1570391902,
    "timestamp": 1570391902,
    "name": "theFile",
    "title": "MacTech_Monthly_Report_October.pdf",
    "mimetype": "application/vpdf",
    "filetype": "pdf",
    "pretty_type": "PDF",
    "user": "U0508V0R4",
    "editable": false,
    "size": 193334,
    "mode": "hosted",
    "is_external": false,
    "external_type": "",
    "is_public": true,
    "public_url_shared": false
  }
}
```

Headers

```
HTTP/1.1 100 Continue
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Content-Length: 557
Connection: keep-alive
Date: Sun, 06 Oct 2019 21:19:12 GMT
Server: Apache/2.4.18 (Ubuntu)
```

Trace

```
== Info: Trying 99.84.174.165...
== Info: Connected to slack.com (99.84.174.165) port 443 (#0)
== Info: ALPN, offering http/1.1
== Info: Cipher selection: ECDHE_RSA_AES128_GCM
```

Copyright © 2019, Precursor.ca

#mactech

11:55 AM Your monthly status report for October is ready!

PDF

Monthly Status Report
191 kB — Click to view

Monthly Report September

An automated report generated by FileMaker via API calls for: MacTech

Managed Software Center

DETECTX STATUS

259 Clean	4 Issues	0 Infected
-----------	----------	------------

Message #testing

precursor.ca/slides



SLACK API POST 1.0.0

Post a Message using "chat.postMessage"

Upload a file using "files.upload"

Copyright © 2019, Precursor.ca

precursor.ca/slides



Triggers & Webhooks

precursor.ca/slides



Webhooks > Webhooks

Webhooks

Webhooks let you listen for support events in real time and easily pass data to use in services and integrations. [Learn about webhooks](#)

Search by name

Name	Endpoint	Status
Trello Descriptio...	https://api.trello.com/1/card/{{ticket.ticket_field_360027632...	Active



Actions

Actions that will occur if global conditions are satisfied

Notify active webhook Trello Description Modify

Endpoint
https://api.trello.com/1/card/{{ticket.ticket_field_360027[REDACTED]}}?customFieldItems=true&key=e925aed[REDACTED]7efd

Description
-

Method
PUT

JSON body

```
1 { "idList": "{{ticket.requester.organization.custom_fields.trello_list_id}}",  
2   "desc": "{{ticket.description}}"  
3 }
```

[View available placeholders](#)



Conditions

Conditions that must be met for the trigger to run

Meet ALL of the following conditions

Ticket Updated

Trello List ID

Status Solved

Tags

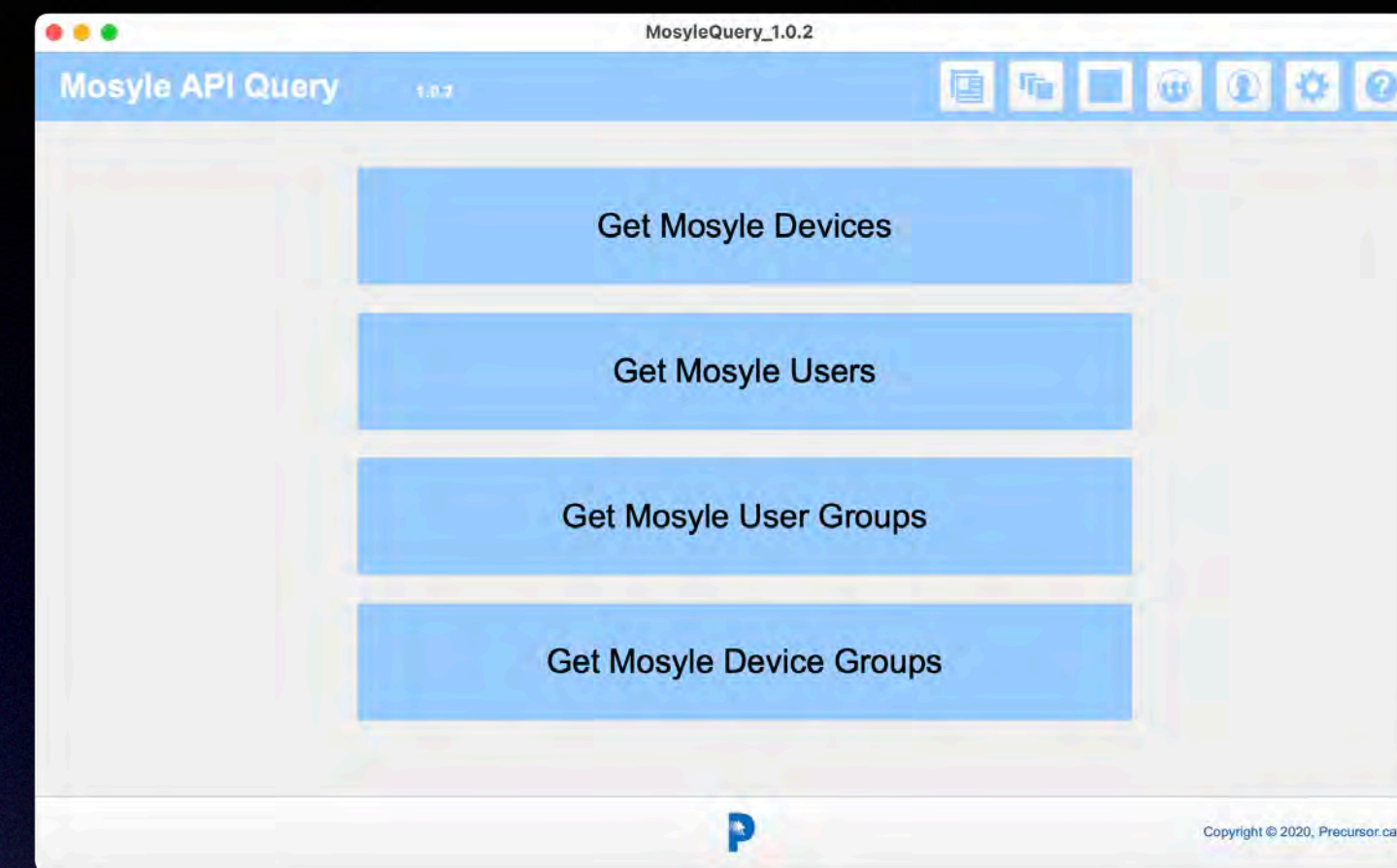
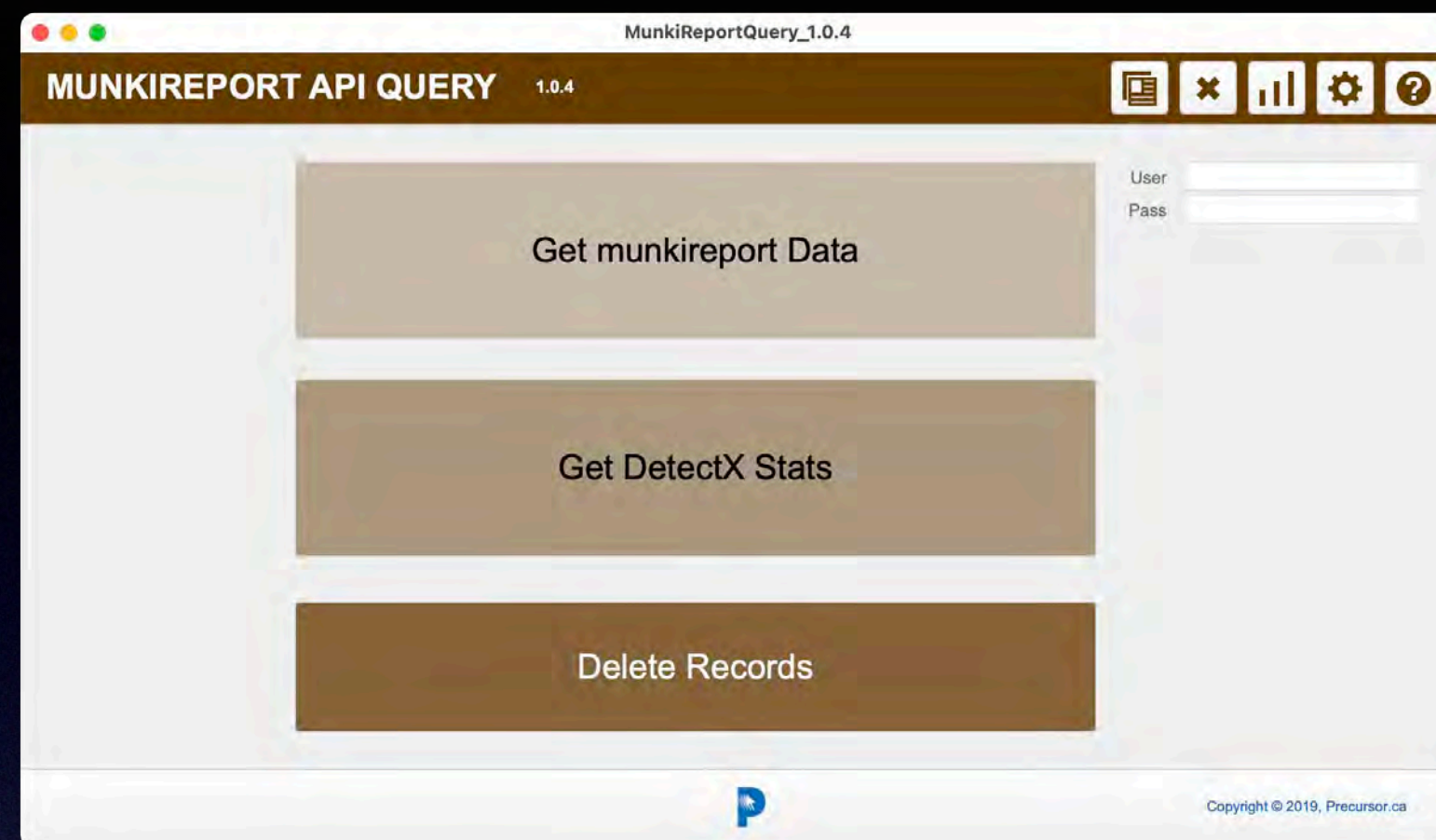
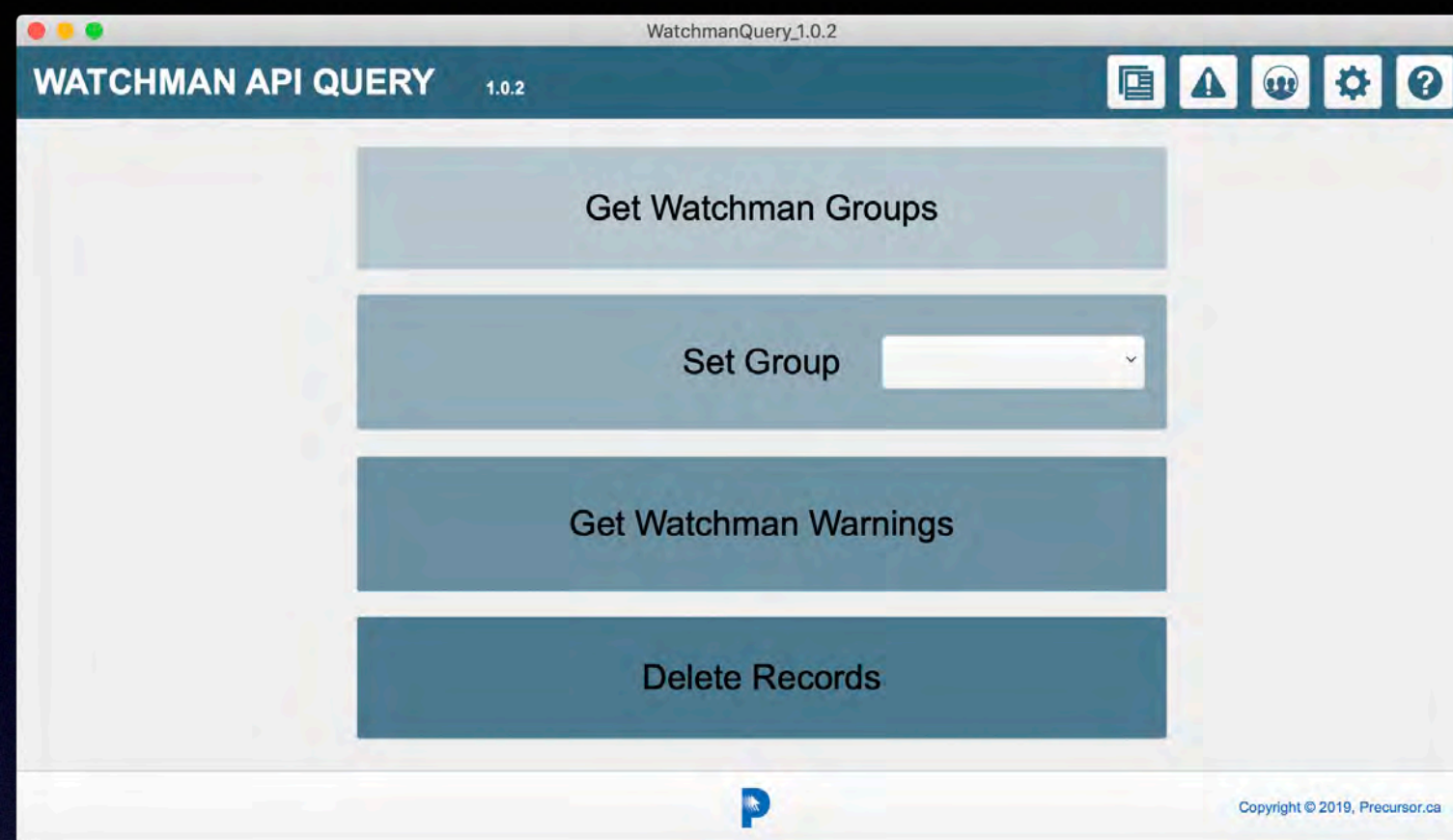
[Add condition](#)

precursor.ca/slides

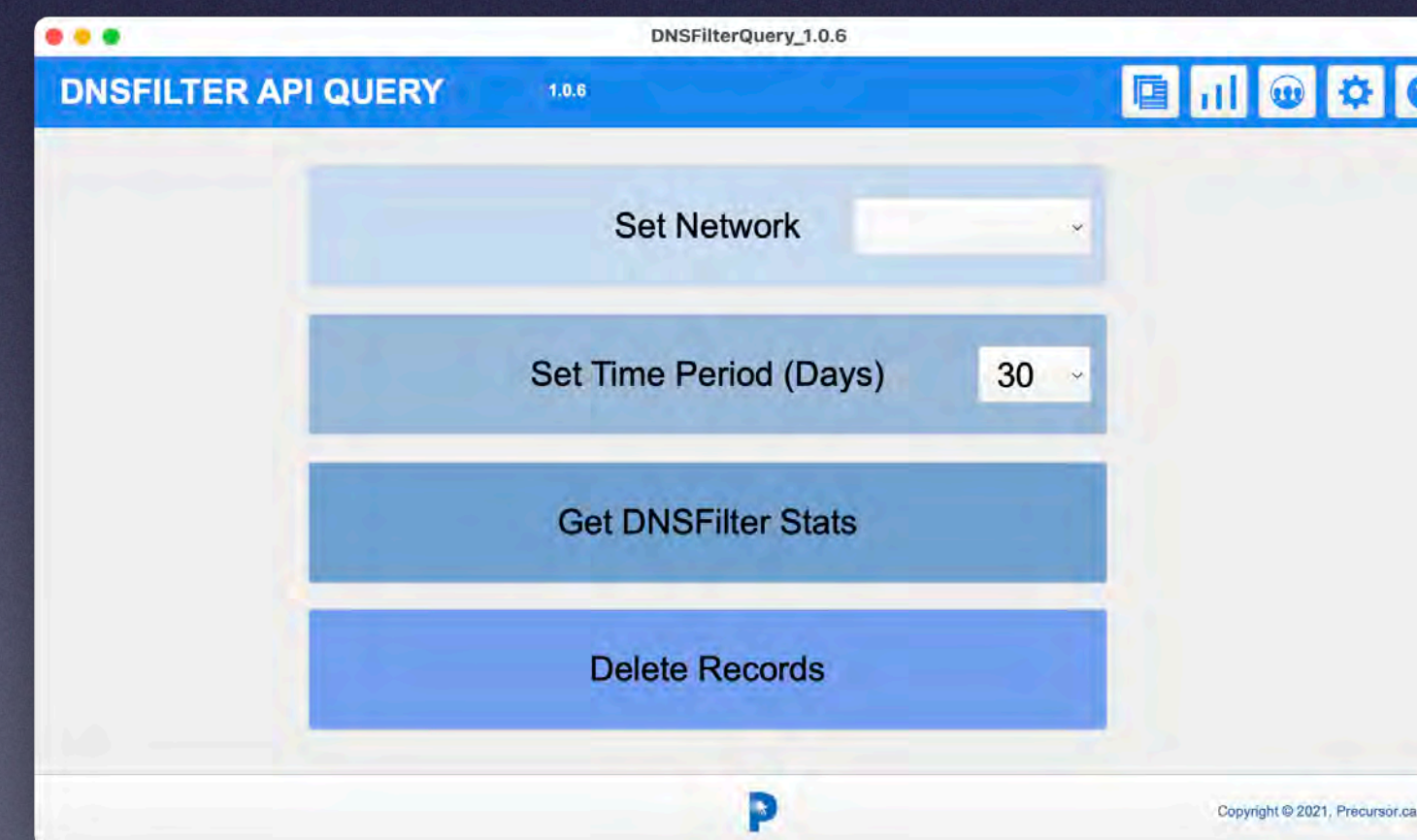
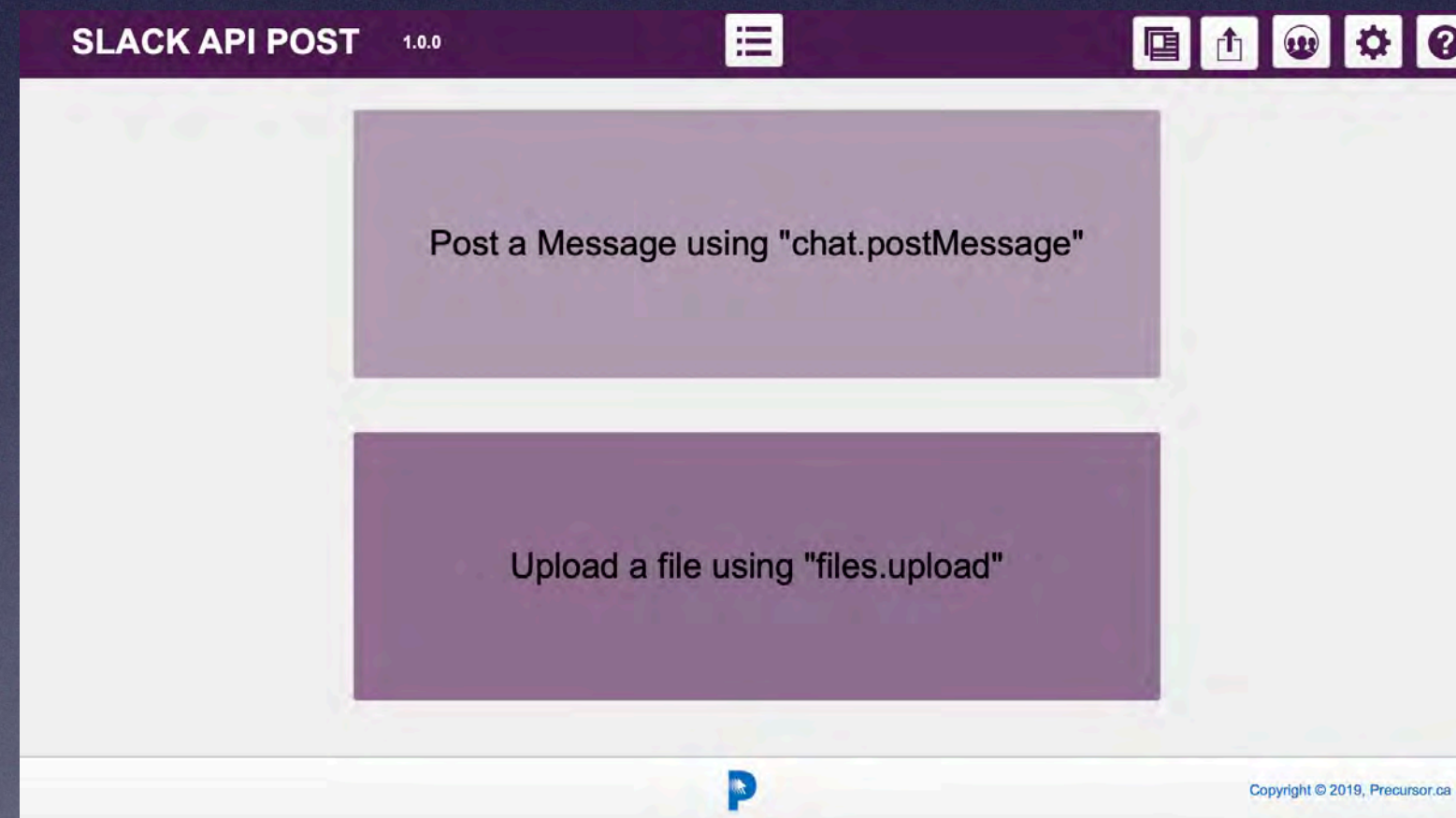
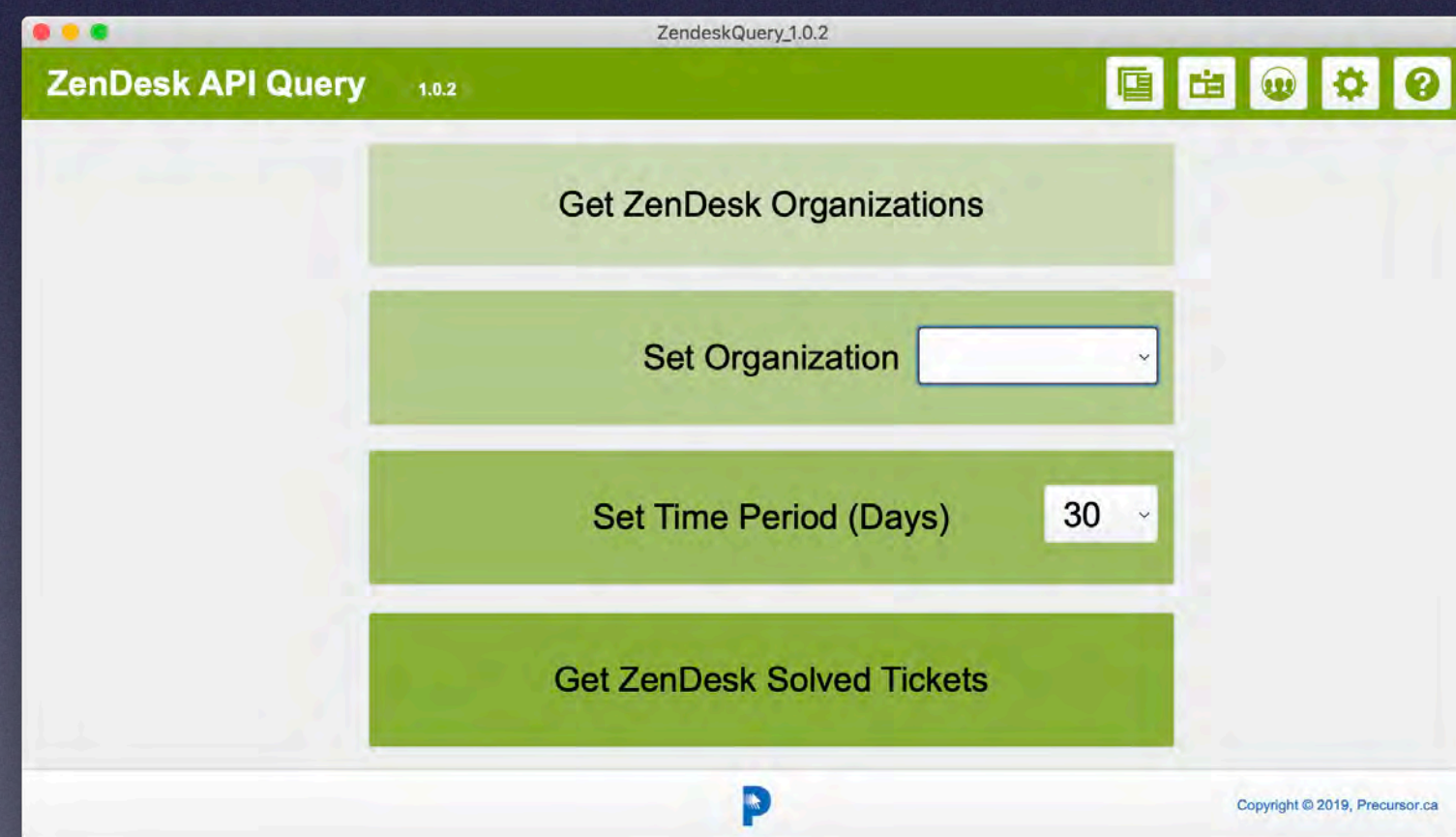


Templates

precursor.ca/slides



<https://precursor.ca/APItemplates>



precursor.ca/slides



Claris FileMaker

45 day demo

<https://www.claris.com/trial/>



ACN FileMaker NFR

<https://consultants.apple.com>

Claris Engage

<https://community.claris.com/en/s/engage>

FileMaker Partners (free to partners)

<https://www.claris.com/partners/become-a-partner.html>

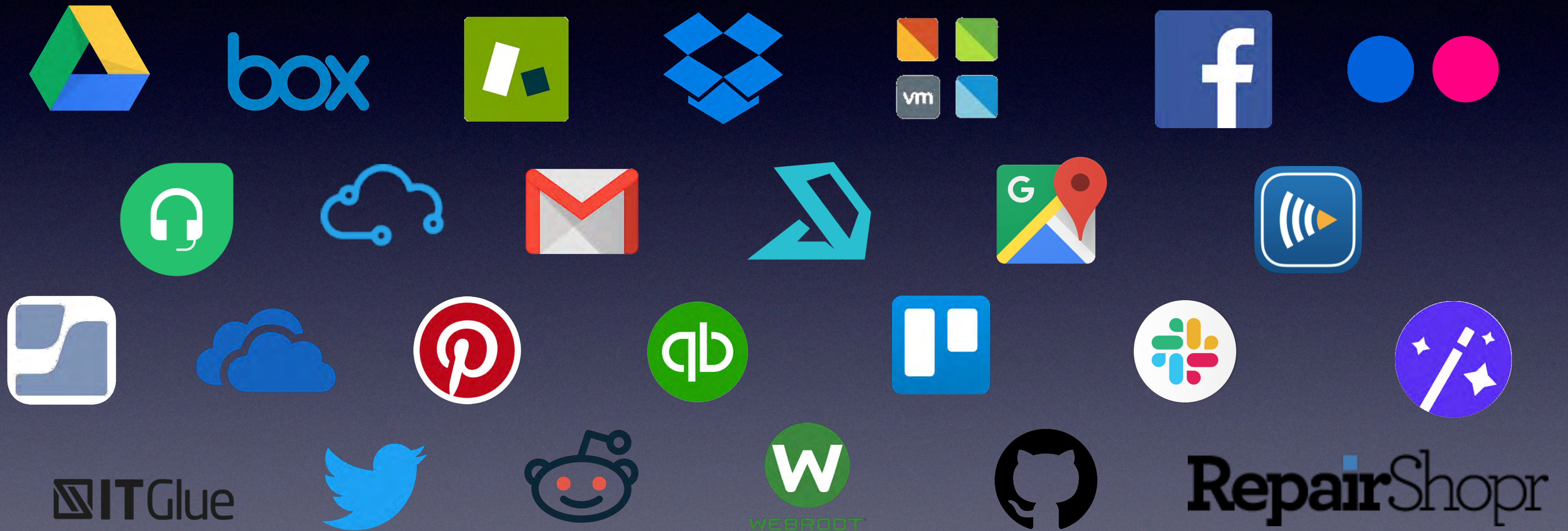
Purchase at Retail

<https://store.claris.com>

precursor.ca/slides



REST APIs



precursor.ca/slides



Alex Narvey



<https://Precursor.ca>



@precursorca

precursor.ca/APItemplates

precursor.ca/slides