RSA ARCHER
SUMMIT 2018

# HELLO AGAIN, WORLD!
# API BASICS LAB


Scott Hagemeyer

Patrick Rausch

Kyle Rickert

Brijesh Gupta

# LAB GOALS

## WHAT WILL WE ACCOMPLISH TODAY?

- Introductions
- Slides
  - RSA Archer APIs Overview
  - RSA Archer REST API Client Overview
- Coding
  - Hello, World!
  - Login
  - Get Applications
  - Get Fields
  - Create/Get Content
  - Logout
- Questions

# WHAT ARE THE APIS?

## NOTHING TO BE AFRAID OF

- **API** stands for **A**pplication **P**rogramming **I**nterface
  - Set of routines, protocols, and tools
  - Specifies how you authenticate, and request / receive data from the system

- **SOAP** stands for **S**imple **O**bject **A**ccess **P**rotocol
  - Requests require programmatic envelope
  - Responses are automatically parsed by the SOAP **WSDL** (**W**eb **S**ervice **D**efinition **L**anguage)

- **REST** stands for **RE**presentational **S**tate **T**ransfer
  - Stateless protocol with standardized operations
  - Lends itself to performance, reliability, and scale
  - Responses parsed by the caller

# WHAT'S THE DIFFERENCE?

## THREE APIS – THREE USES

- **Web Services API (SOAP)**
  - ../Archer/ws virtual directory
  - XML used for request/response
  - Uses HTTP POSTs & GETs
  - Values passed like Excel
  - Focus on back office objects
  - Includes search capability
  - Available since the 4.x days
- **RESTful API**
  - ../Archer/api IIS application
  - JSON used for request/response
  - Simpler alternative to SOAP
  - Lighter load = better performance
  - Values passed in HTTP Request header & body
  - Focus on back office objects
  - No search capability
  - Available since 5.4.1.x
- **Content API**
  - ../Archer/contentapi IIS application
  - JSON used for request/response
  - Values passed in HTTP Request header & body
  - Focus on front office objects; specifically content
  - Data extraction from applications and solutions
  - Available since 6.4.x

# HOW ARE THEY THE SAME?

- Allow clients (users or applications) to access platform data and application content
- Every request must include a session token so permissions are respected
- Accessible via most programming languages (.NET, Java, PHP, PowerShell, Python, C#, etc)
- IIS Authentication set to Anonymous on API directories/applications
- For Single Sign-On (SSO), exclude the /ws, /api, and /contentapi paths for authentication (SIteMinder, ClearTrust, etc)

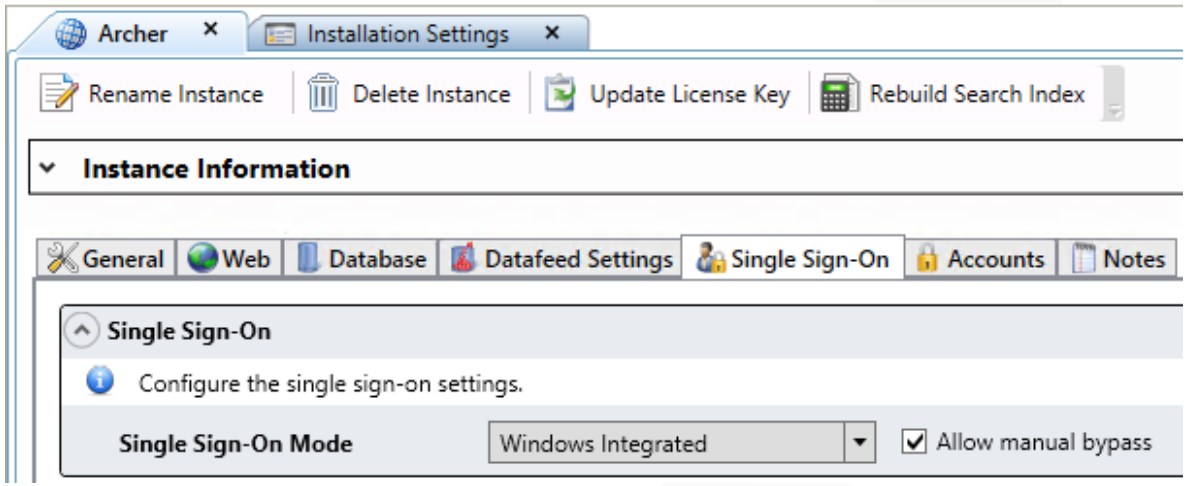
# ALLOW MANUAL BYPASS

## THOU SHALL NOT PASS; UNLESS YOU HAVE A SESSION TOKEN

- To create a Session Token using the REST API with SSO enabled, the Allow Manual Bypass options must be enabled

- Web Services API can create a Session Token regardless of this setting, and afterwards the token can be used with REST API calls

- From the ACP, open the Instance settings and select the Single Sign-On tab to enable



# REST API GUIDE

## VERBS AND ACTIONS

| Request Verb | Action | Example |
|---|---|---|
| GET | Select record 123456 | GET /api/core/content/123456 |
| GET | Select all applications | GET /api/core/system/application |
| POST | Insert a new record | POST /api/core/content |
| PUT | Update a record | PUT /api/core/content |
| DELETE | Delete user 123 | DELETE /api/core/system/user/123 |

- The standard GET request is unencrypted
- To avoid this inherent vulnerability, the API watches for and responds to the X-HTTP-Method-Override header

# SELECT, FILTER, & SORT IN REST

## GIVE ME WHAT I WANT, THE WAY I WANT IT!

- Limit results with Odata (SQL-like statements)
  - Follow URL with a question mark (?) and combine actions using ampersands (&)
  - Refer to the REST API Guide to see which actions are supported
- $select : return results for the given Property Names (fields)
  - ../api/core/system/user?$select=Id,UserName,FirstName,LastName,AccountStatus,DomainId
- $filter : filter the results using Logical, Arithmetic, and Grouping Operators
  - ../api/core/system/user$filter=Id gt 200
- $orderby : sort results in ascending (asc)(the default) or descending (desc) order
  - ../api/core/system/user?$orderby=LastName desc
- $top : sets the page size of the current request
  - ../api/core/system/user?$top=50
- $skip : selects the correct page
  - Get page 3: ../api/core/system/user?$top=50&$skip=100

# FILTER OPERATORS

| Operator | Description | Example |
|---|---|---|
| Logical Operators | | |
| eq | Equal | /application?$filter=Alias eq 'Patches' |
| ne | Not equal | /application?$filter=Alias ne 'Patches' |
| gt | Greater than | /user?$filter=Id gt 300 |
| ge | Greater than or equal | /user?$filter=Id ge 300 |
| lt | Less than | /user?$filter=Id lt 300 |
| le | Less than or equal | /user?$filter=Id le 300 |
| and | Logical and | /user?$filter=Id le 300 and Id gt 500 |
| or | Logical or | /user?$filter=Id le 300 or Id gt 500 |
| not | Logical negation | /application?$filter=not endswith(Name,'App') |
| Arithmetic Operators | | |
| add | Addition | /Products?$filter=Price add 5 gt 10 |
| sub | Subtraction | /Products?$filter=Price sub 5 gt 10 |
| mul | Multiplication | /Products?$filter=Price mul 2 gt 2000 |
| div | Division | /Products?$filter=Price div 2 gt 4 |
| mod | Modulo | /Products?$filter=Price mod 2 eq 0 |
| Grouping Operators | | |
| ( ) | Precedence grouping | /Products?$filter=(Price sub 5) gt 10 |

# RSA ARCHER REST API CLIENT

## REST API PROGRAMMING – SIMPLIFIED

- Last year we used a prototype version of this
- We now have an official NuGet package in alpha stage
- More work to be done before this goes beta, and finally public release
- Goal is to make API programming more accessible

# CRACK YOUR KNUCKLES – LET'S GET CODING!

# LAB GUIDE CORRECTION

## MEA CULPA

- Please follow the steps on page 1 of the lab guide
- At the top, there is a collapsed region called Fields
  - Expand the region
  - Delete the “Test Instance” fields
  - Uncomment the “Summit Lab VM” fields
- Leaves this at the top:

```
19  namespace API_Basics_Lab_2018
20  {
        0 references
21      class Program
22      {
23          #region Fields
24
25          private const string BASE_URL = "http://archer.local/api";
26          private const string username = "webapi";
27          private const string instance = "apilab";
28          private const string password = "Password$";
29
30          #endregion
```

# THANK YOU

RSA ARCHER
SUMMIT 2018