**Lab Guide**

Exploring the RPA API

Nigel T. Crowther

Hands-on Lab

Version 1.0 for General Availability



NOTICES

This information was developed for products and services offered in the USA.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing

IBM Corporation

North Castle Drive, MD-NC119

Armonk, NY 10504-1785

United States of America

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

**TRADEMARKS**

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

IT Infrastructure Library is a Registered Trade Mark of AXELOS Limited.

ITIL is a Registered Trade Mark of AXELOS Limited.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

© Copyright International Business Machines Corporation 2020.

This document may not be reproduced in whole or in part without the prior written permission of IBM.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

**Table of Contents**

[1 Introduction 4](#_Toc109742603)

[2 RPA API Overview 5](#_Toc109742604)

[2.1 Prerequisites 6](#_Toc109742605)

[2.1.1 Complete RPA toolkit Lab 6](#_Toc109742606)

[2.1.2 Download Lab Zip 6](#_Toc109742607)

[2.1.3 Download SoapUI 7](#_Toc109742608)

[2.1.4 Download Cygwin Unix terminal 7](#_Toc109742609)

[3 Exploring the RPA API 9](#_Toc109742610)

[3.1 Test the RPA API from PowerShell 9](#_Toc109742611)

[3.1.1 Step 1 – Host Selection 9](#_Toc109742612)

[3.1.2 Step 2 – Enter your tenant credentials 10](#_Toc109742613)

[3.1.3 Step 3 – Select the tenant 10](#_Toc109742614)

[3.1.4 Step 4 – Select the process 11](#_Toc109742615)

[3.1.5 Step 5 – Edit the payload 11](#_Toc109742616)

[3.1.6 Step 6 - Run the bot 12](#_Toc109742617)

[3.1.7 Step 7 – View the Result 12](#_Toc109742618)

[3.1.8 View the RPA API Curl 14](#_Toc109742619)

[3.1.9 Run curl 14](#_Toc109742620)

[3.2 Testing the API from SoapUI 16](#_Toc109742621)

[3.2.1 Open SoapUI 16](#_Toc109742622)

[3.2.2 Setting Project Variables 17](#_Toc109742623)

[3.2.3 Run the test suite 18](#_Toc109742624)

[3.2.4 Examine the test suite flow. 20](#_Toc109742625)

[3.3 Generating an Open API Specification 22](#_Toc109742626)

[3.3.1 Start Eclipse 22](#_Toc109742627)

[3.3.2 Optional - run curl against your SaaS tenant 30](#_Toc109742628)

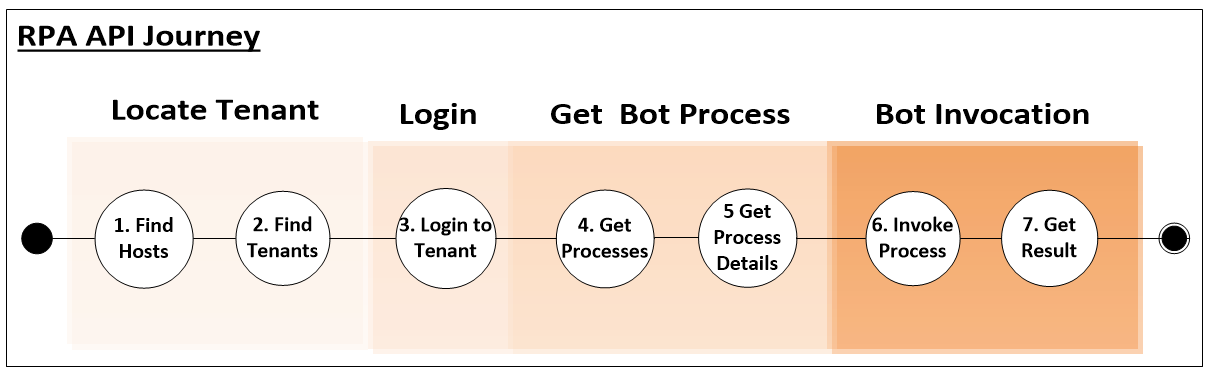
# Introduction

In this lab you will Explore the RPA API. The lab is based on the IBM RPA API documentation:

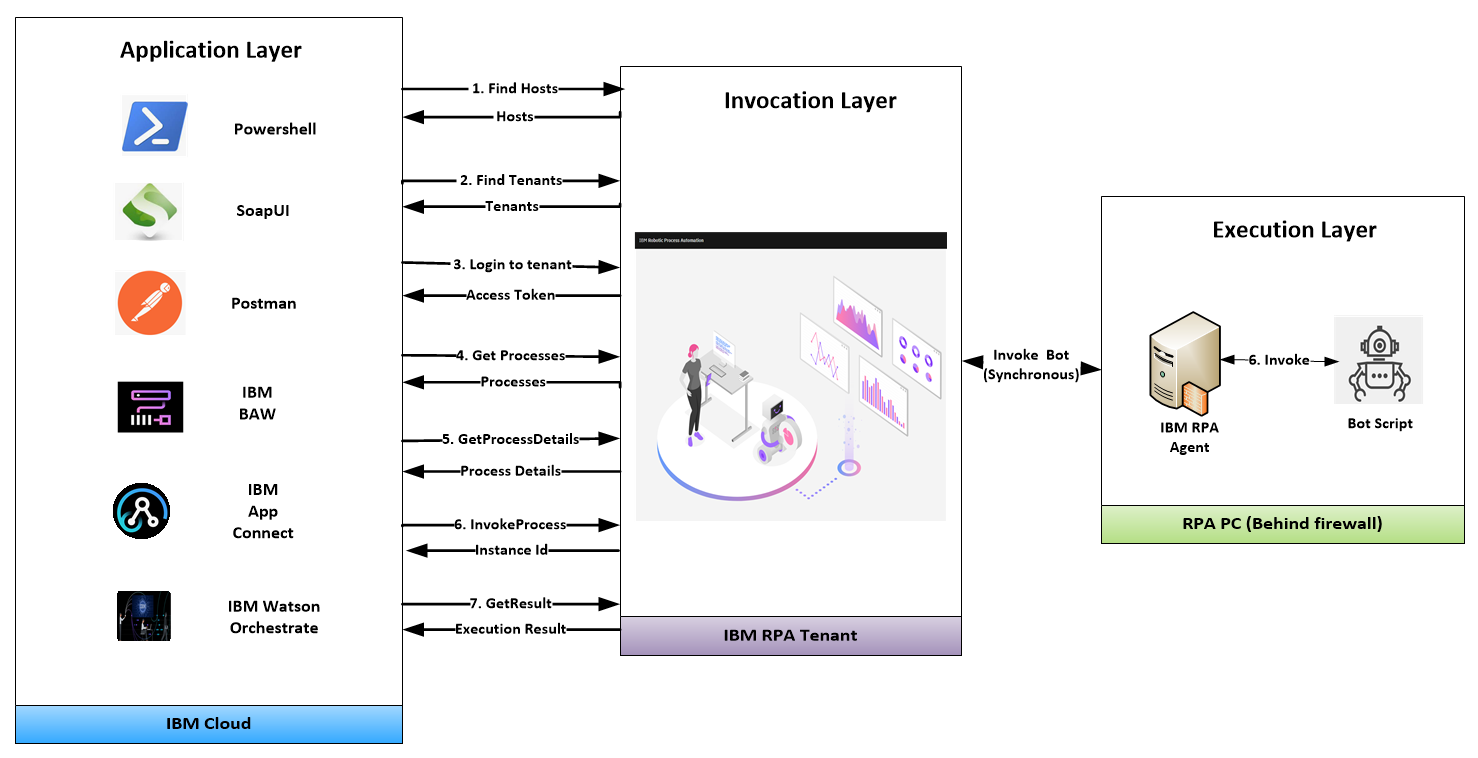
[**https://ww****w.ibm.com/docs/en/rpa/21.0?topic=automation-rpa-api-reference**](https://www.ibm.com/docs/en/rpa/21.0?topic=automation-rpa-api-reference)

# RPA API Overview

To run a bot using the RPA API, you follow a journey. The journey starts with finding a host and ends with bot execution. The following flow diagram describes each step:



The API is invoked against the RPA tenant. The tenant authenticates the caller, locates the process in which the bot resides and then permits invocation of the bot. It is at this point that the RPA tenant finds the designated computer and invokes the internal synchronous API to run the bot. Once the bot is run, the result is retrieved asynchronously by invoking *GetResult*. This interaction is depicted below:



This lab will now examine this API flow in detail.

## Prerequisites

### Partially Complete RPA toolkit Lab

Complete steps 2.1 to 2.4 in the lab:

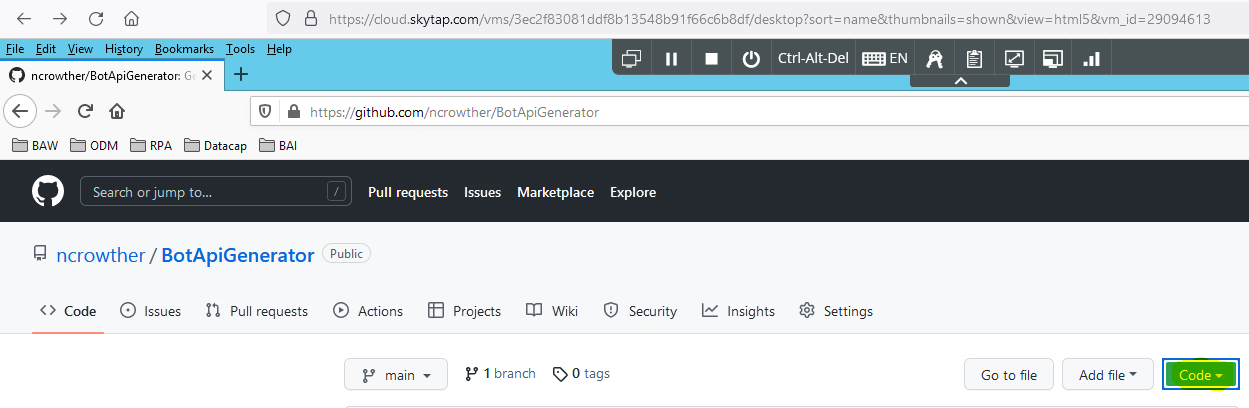
<https://github.com/juseljuk/IBM-RPA-Toolkit-for-BAW/blob/master/downloads/Using%20IBM%20RPA%20with%20IBM%20BAW%201_1.pdf>

### Download Lab Materials

The prerequisite material for this lab is stored in **Github**. Using a browser, navigate to the folder:

<https://github.com/ncrowther/BotApiGenerator>

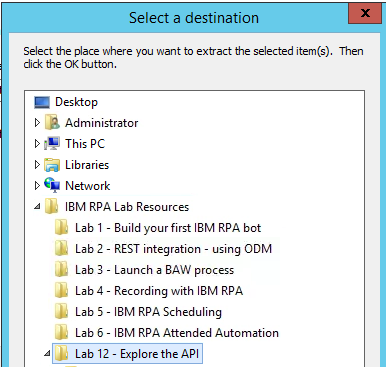
You should see the following:



Click on the  button, and then select *Download Zip*. Once downloaded, right-click the zip and select *extract all….* Then extract the zip to a new folder. If using the Skytap image, we recommend:

C:\Users\Administrator\Desktop\IBM RPA Lab Resources\Lab 12 – Explore the API

See below:



Optional – Download Cygwin

Open <https://cygwin.com/setup-x86_64.exe> in your browser and download and run exe.

Apply all default install settings.

### Download SoapUI

This step is not required if you are using the Skytap image - SoapUI is already installed.

You can download an open source version of SoapUI from the following location:

<https://www.soapui.org/downloads/soapui/>

Click on *Download SoapUI Open Source*.

Save the zip, extract and install the software using the default settings.

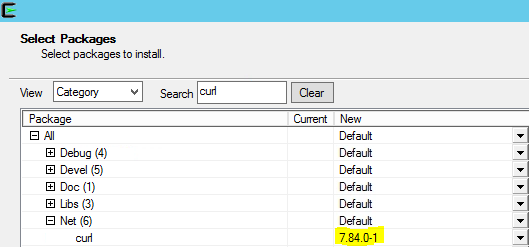
### Download Cygwin

Using a browser, download and run:

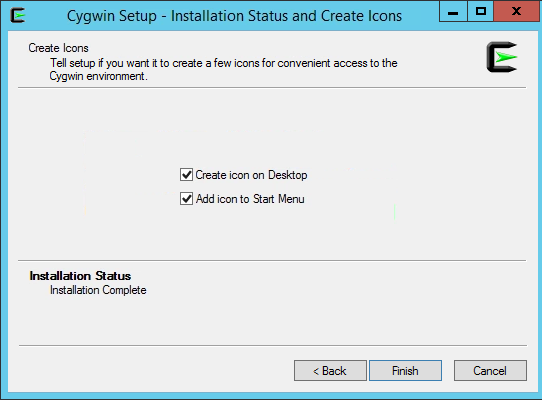
<https://www.cygwin.com/setup-x86_64.exe>

Select your local ftp server.

When selecting packages, search for curl and change ‘Skip’ to the latest version. See below:



Continue pressing defaults until the installation progress window starts. After a minute you should see this:



Press Finish. The install is complete.

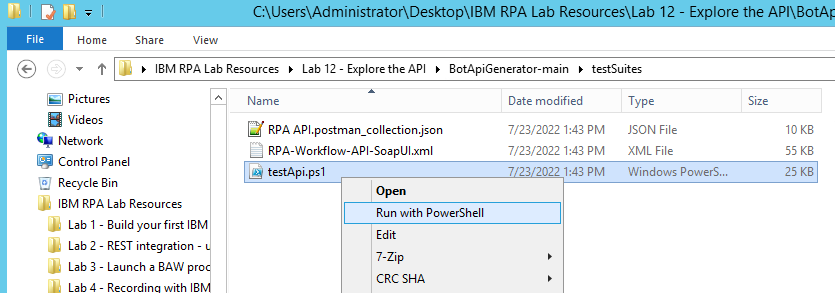
Once done, let’s get started with exploring the APIs in this lab!

# Exploring the RPA API

## Test the RPA API from PowerShell

Navigate to the Folder in which you unzipped the contents in the previous step:

Right-click *testApi.ps1* and select *Run with PowerShell*:

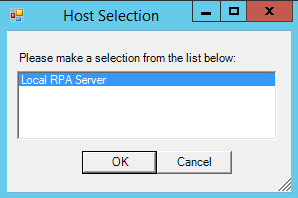


Ignore the security warning and press *Open*. A series of dialog boxes should appear. Let’s follow the steps.

### Step 1 – Host Selection

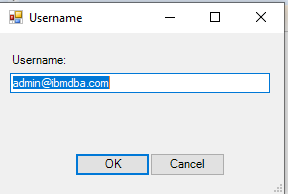
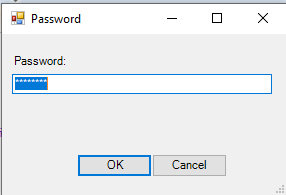
If you are running the Skytap image you will not be able to see the SaaS tenants. Ignore the message “Unable to find SaaS tenants”.

Select *Local RPA Server*



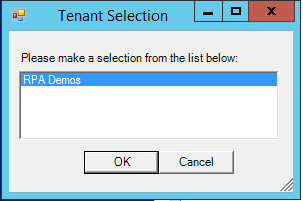
### Step 2 – Enter your tenant credentials

The username is [admin@ibmdba.com](mailto:admin@ibmdba.com), the password is *passw0rd.* These should already be the default values:

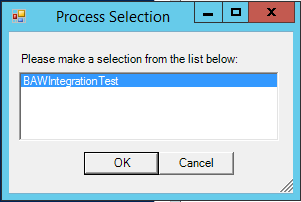
### Step 3 – Select the tenant

Select tenant. If you are running within the Skytap image you only have one option. Select *RPA Demos:*



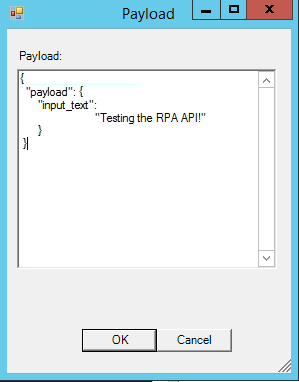
### Step 4 – Select the process

Select the process you defined in the earlier step:



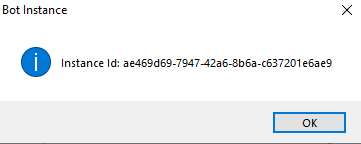
### Step 5 – Edit the payload

The payload will be generated from the bot input parameters. You can modify the input value if you wish:



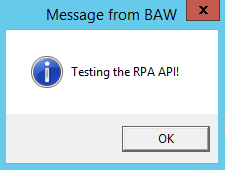
### Step 6 - Run the bot

The bot runs and a dialog box appears showing the instance id of the running process

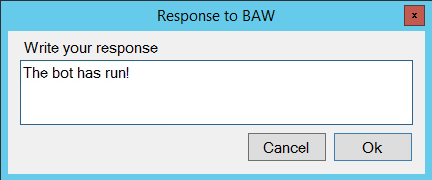


### Step 7 – View the Result

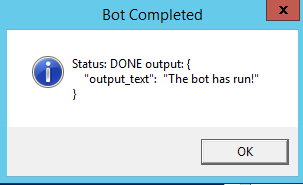
After a second, the bot runs, and you will see the following message:



Click OK and type your response:



The bot may take several seconds to run, in which case it will either be in *new* or *processing* state. When completed, the state will have a status of *done* and you will see the following dialog:



Press OK to finish.

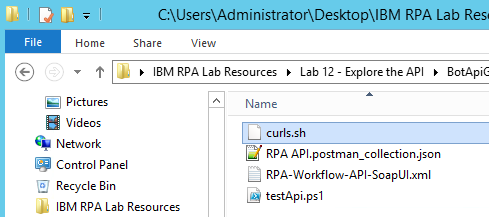
Congratulations, you have run your first bot using PowerShell to drive the asynchronous API!

Now we will view the actual RPA API commands executed.

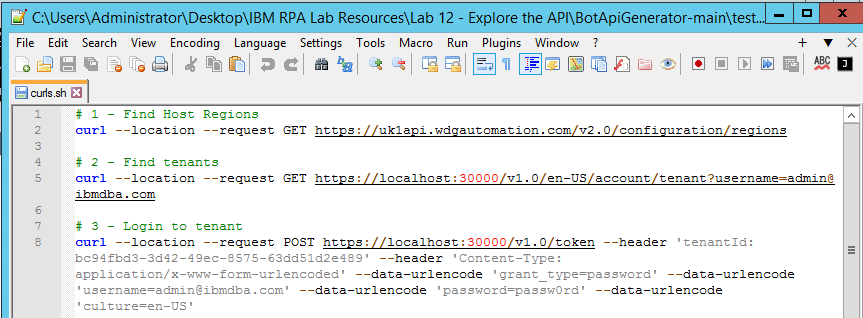
### View the RPA API Curl

In this section we will use curl to test the RPA API. Curl is the de facto command line tool to test APIs.

Navigate to the directory in which you ran PowerShell. Right click *curls.sh* and edit using *Notepad++.*



You will see the curl executed in the previous step.



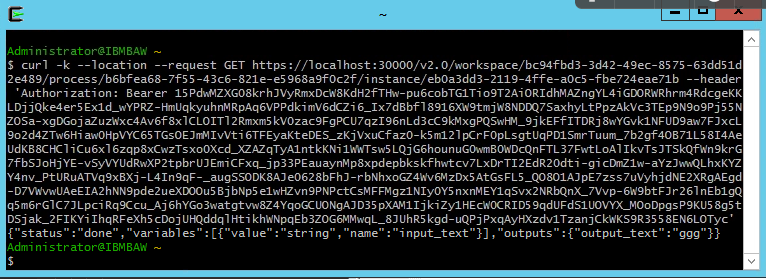
### Run curl

From the desktop, open



A Unix terminal should appear.

Paste each curl command into the terminal and hit return. At the end of the execution of the curl commands, your terminal should look like this:



Congratulations, you have now run a bot using curl!

## Testing the API from SoapUI

SoapUI is an open source graphical tool for testing APIs. In this section we will use SoapUI to test the RPA external API.

### Open SoapUI

Open SoapUI by double-clicking:

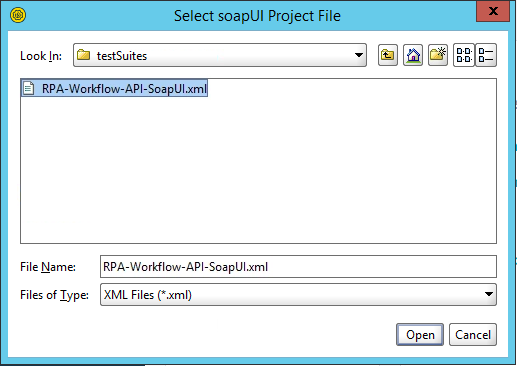
SoapUI

Once open, click the Import button. Navigate to the following folder:

C:\Users\Administrator\Desktop\IBM RPA Lab Resources\Lab 12 - API\BotApiGenerator-main\testSuites

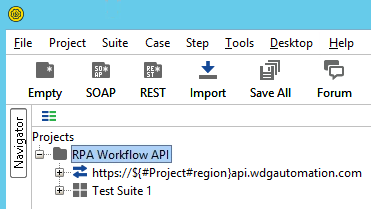
**Note**: This location may vary depending on where you downloaded your lab in step 2.1.2

Select *RPA-Workflow-API-SoapUI.xml* and press *Open*:



Ignore the version warning.

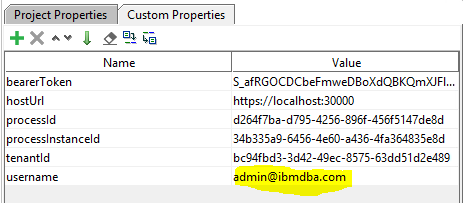
Verify the file is imported:



### Setting Project Variables

**Note:** If you are using Skytap you can skip this step as the variables have been pre-configured.

The test suite retrieves information from project variables. Click on the project *RPA Workflow API* and select *custom properties* tab.



You can edit these variables to your own tenant and processes. The two key variables are *username* and *hostURL*.

The default for *username* is

[admin@ibmdba.com](mailto:admin@ibmdba.com).

If you are not using Skytap, you need to set this to your RPA user.

The default for *hostURL* is

<https://ibmbaw:30000>

If you are not using Skytap, you will need to set this to your RPA server. If you are using SaaS, you can use the tenant below:

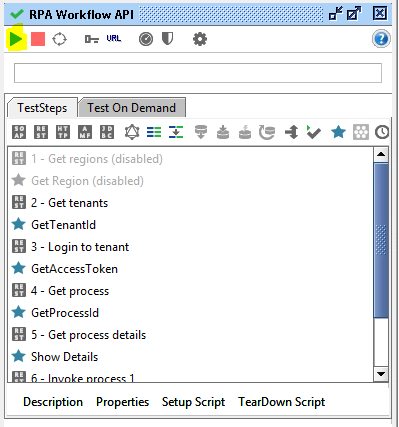
<https://us1api.wdgautomation.com>

### Run the test suite

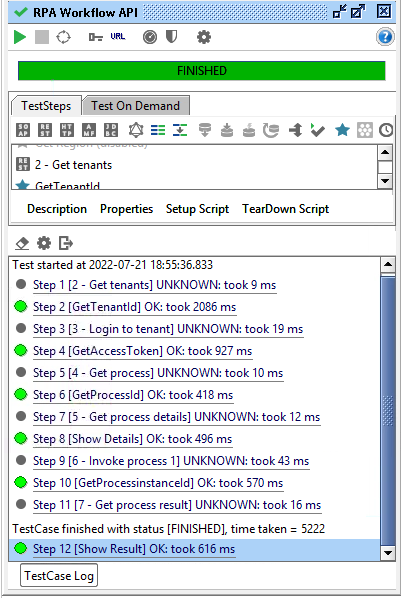
In the Projects tree, navigate to *Test Steps*. See below:



Double-click *Test steps* to open the *RPA Workflow API* test runner. Click *Run* to invoke the RPA API test suite. See below:

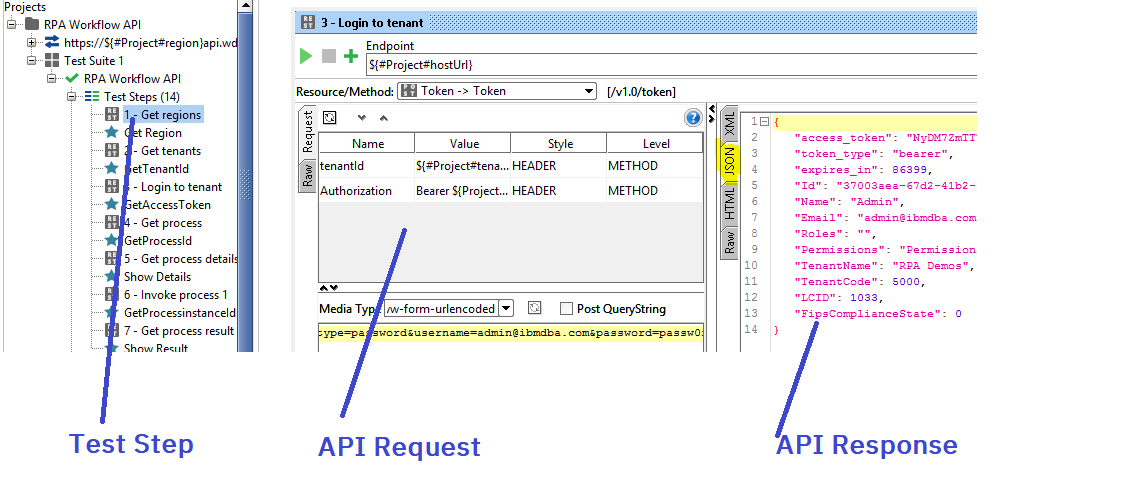


When running, you will be prompted for a series of dialog boxes showing you the API journey. When complete, you should see the following:



### Examine the test suite flow.

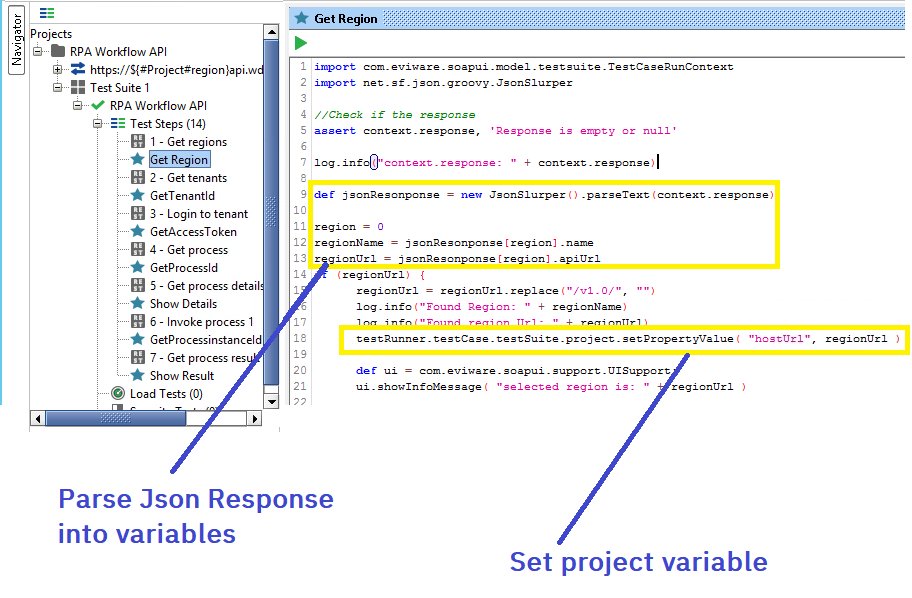
The SoapUI test suite orchestrates the flow of execution by setting variables in one step and reading them in the next. You can verify this by going to **step 1 – Get Regions:**



You can see the invocation request in the center panel and the invocation result on the left most panel.

**Note:** The result is JSON, so you need to select the JSON tab.

Now select \* **Get Region**. This step is written in *Groovy Script*. It parses the response from previous step and sets a project variable to be used in the next step. See below:



Now follow the remainder of the test suite to understand the flow of execution.

## Generating an Open API Specification

Up until now we have used the raw REST RPA API calls. Modern applications usually provide a way of interacting with REST APIs through an *OpenAPI* specification. This specification defines a definition file which the application then uses to generate the underlying REST calls.

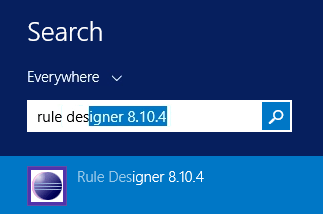
The advantage of using an OpenAPI specification are:

* It’s a universal definition language understood by most modern applications
* It can be viewed and understood in tools such as <https://editor.swagger.io/>

Let’s go ahead and generate an OpenAPI spec for the bot we created.

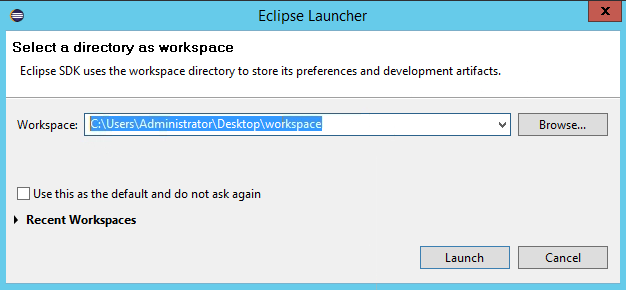
### Start Eclipse

We will be running Java from our favorite IDE, Eclipse. If you are running in the Skyap image, you can access Eclipse by running Rule Designer. Search for *Rule Designer* in the Windows search and then double click to start.



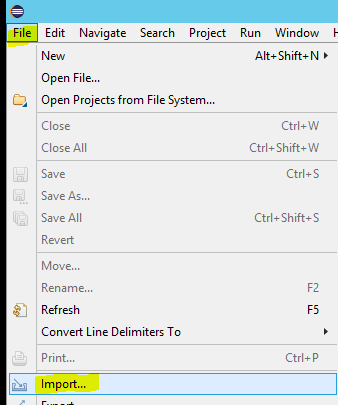
Create a workspace of

C:\Users\Administrator\Desktop\workspace

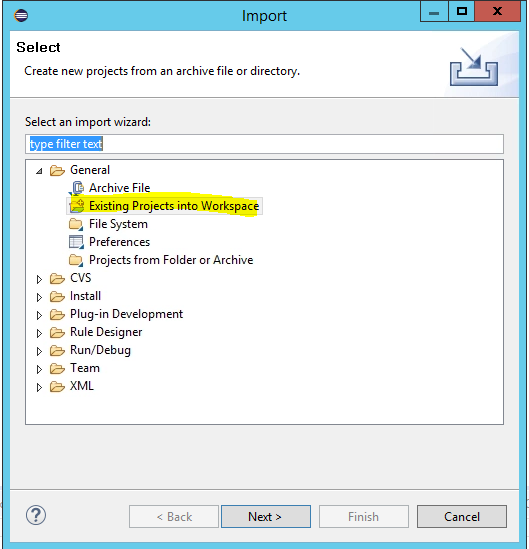


Press *Launch*.

Once Eclipser is open, select *File->Import:*



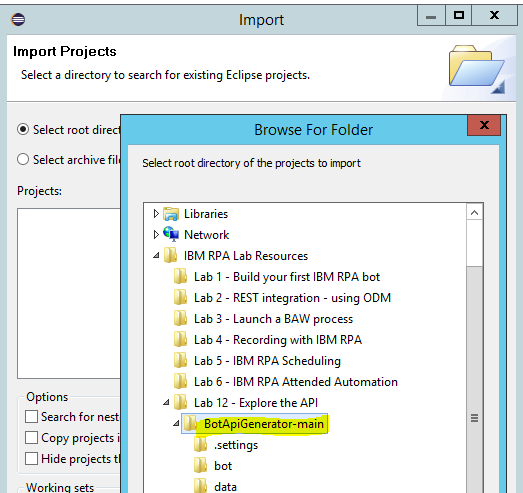
Select *General->Existing Projects into Workspace*:



Select the folder

C:\Users\Administrator\Desktop\IBM RPA Lab Resources\Lab 12 - API\BotApiGenerator-main

**Note**: This location may vary depending on where you downloaded your lab in step 2.1.2



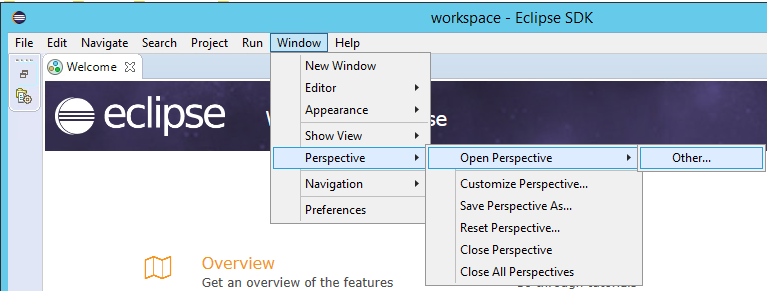
Press *OK.*

Select *BotApiGenerator* and then press Finish:

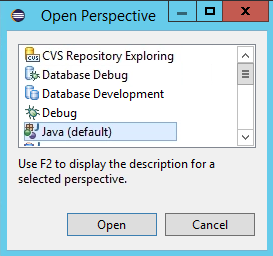


**Set the perspective to Java**

Navigate to Window->Perspective->Open Perspective->Other

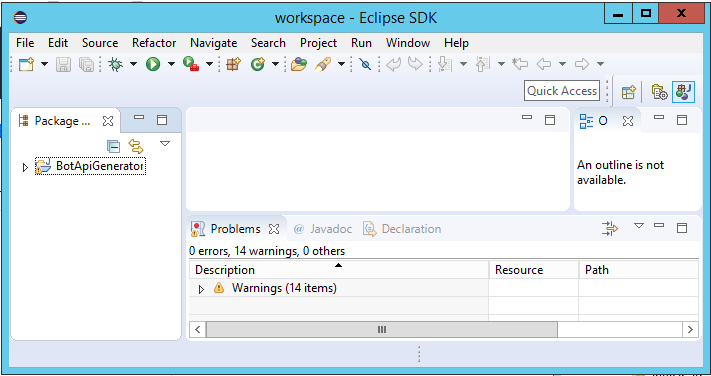
****

Then choose *Java (default):*

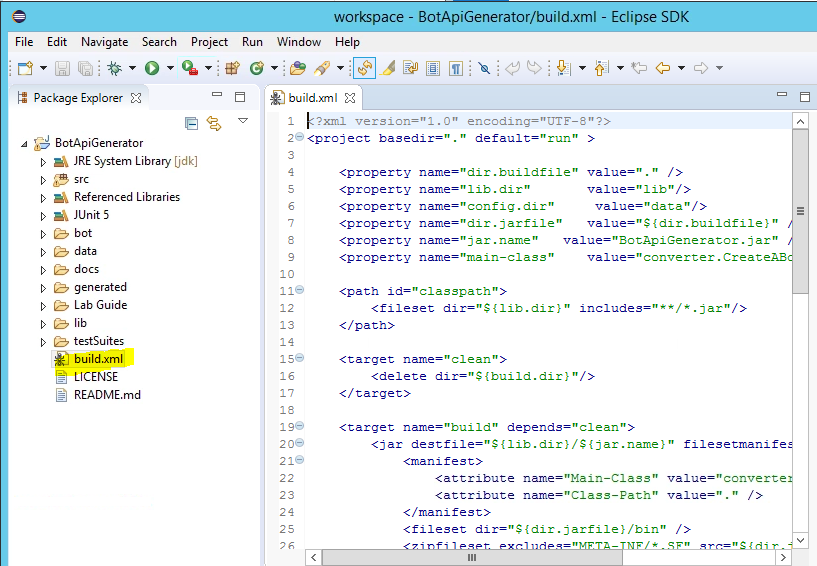


Press *Open*.

Close the Welcome tab. You should see this:



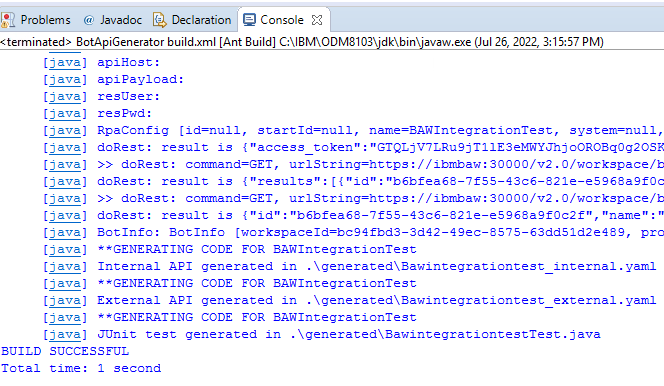
Expand the Package hierarchy on the left panel until you find *build.xml*. Double click to open. You should see this:



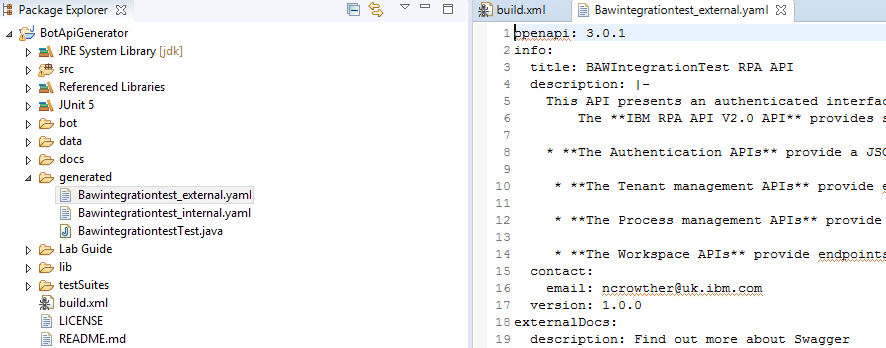
Now execute the *build.xml* by selecting the *External Tools* icon, then select *Run As->Ant Build*. Alternatively, you can press *Alt+Shift+X, Q*.



This should run the API generator. You should see the following in the Console:



Now see what has been generated. Open *Bawintegrationtest\_external.yaml*

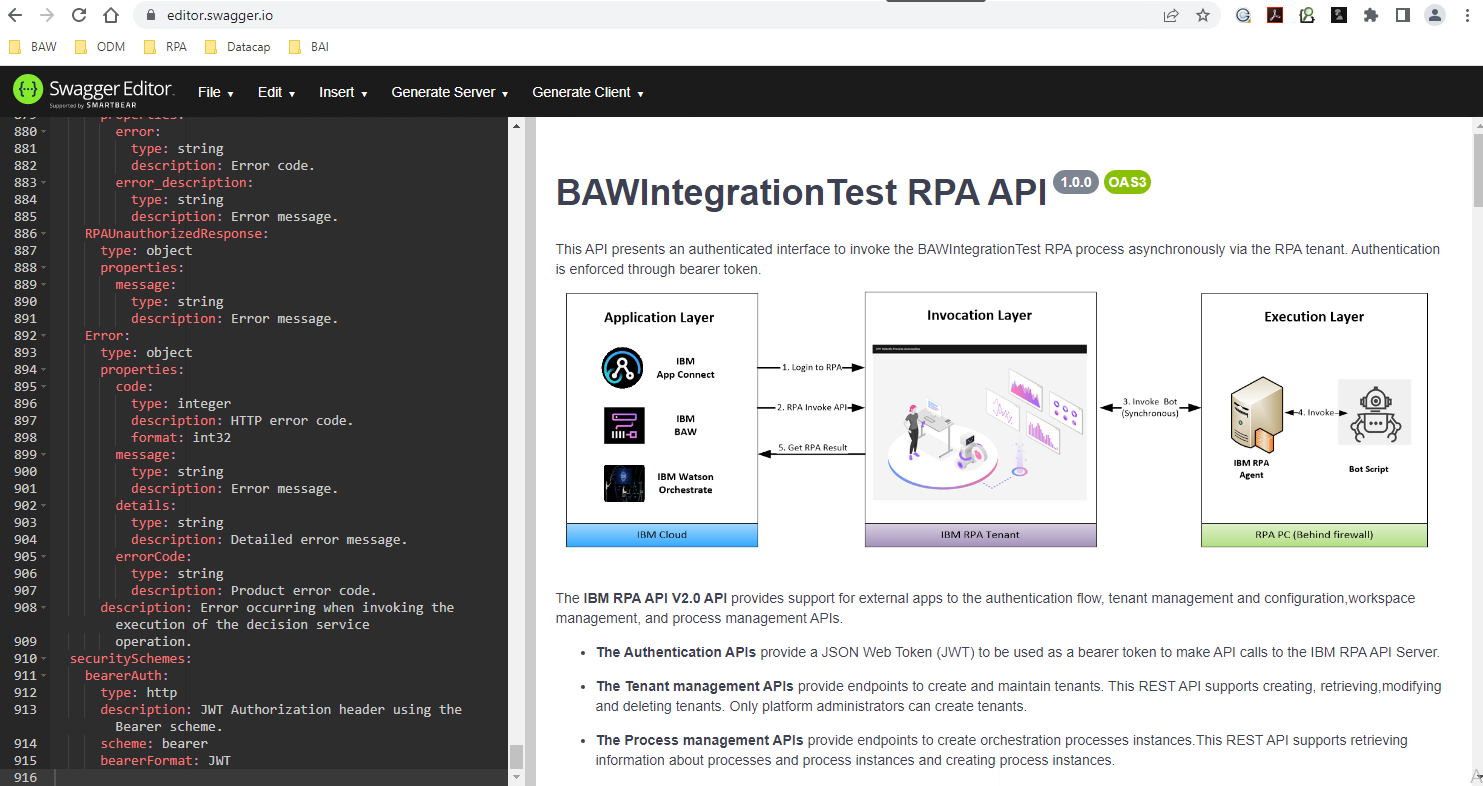


Copy the contents of the yaml into the clipboard.

Open URL

<https://editor.swagger.io/>

Paste the contents of the yaml into the left panel. You should see the Open API specification for your bot:

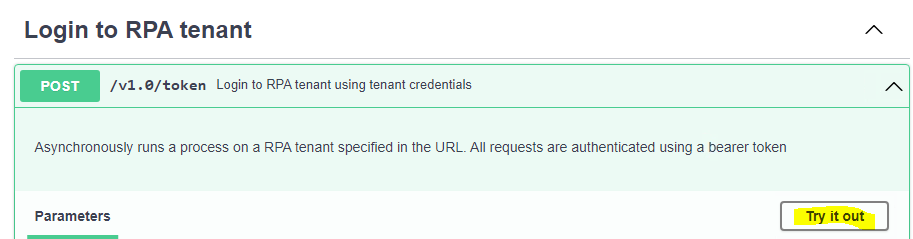


Take a moment to explore the API.

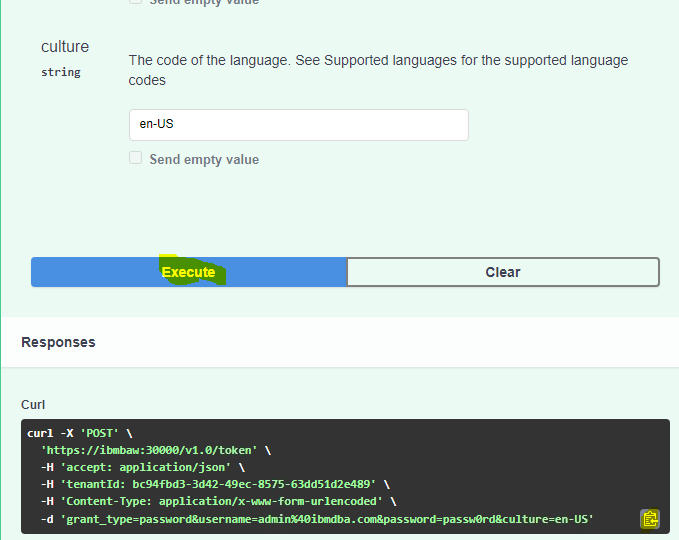
### Run curl from OpenApi

Using the *Swagger Editor*, you can generate curl commands.

Navigate to *Login to RPA* in the Swagger Editor. You should see this:



Press “*try it out*”. Then press *Execute*. See below:

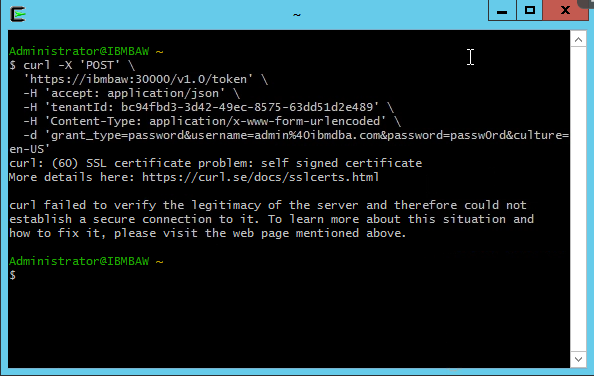


Copy the generated curl command (using the clipboard icon)

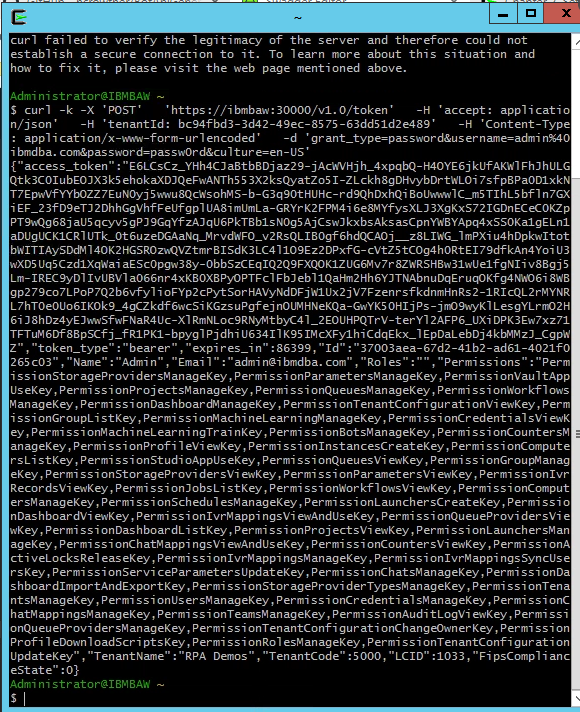
From the desktop, open a Unix terminal:



Paste the curl command into the terminal and hit return. If you are using Skytap, then you will see the following error:



This is not a problem! As we are in a Skytap VM, we can safely ignore certificate errors by adding the **-k** flag to the curl as shown below:



Now the curl runs successfully!

Nicely done! This concludes the lab.