Simio

Forward Thinking

Simio Portal WebAPI Note: RunSimioPortalExpConsole

Last Update: 27 July 2022

Contents
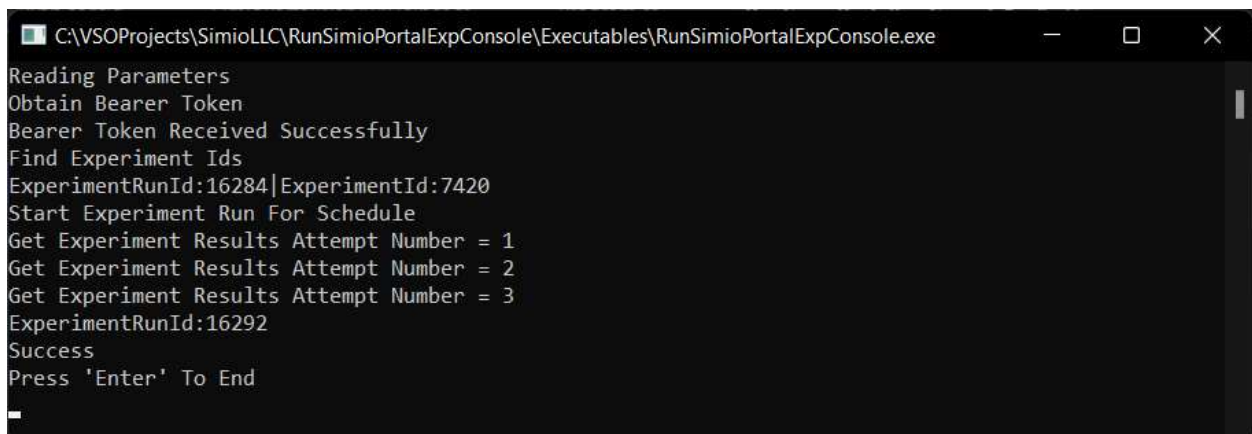
RunSimioPortalExpConsole

Overview

This .NET console application example illustrates external programs can use the Simio Portal WebAPI to create automated workflows for importing running, exporting, and publishing experiments and plans. Please see the Simio Portal API Help for more details on the Simio Portal Web API. See:

https://<yourSimioPortalURL>/Home/APIHelp

Installation Of Executables and Running RunSimioPortalExpConsole

Download from the Executables folder and copy the files into a folder.   You might need to right-click on every file, select properties and make sure the files are unlocked.

Next, open a command prompt.   From the command prompt, navigate to the folder.   To run the RunSimioPortalExpConsole, just type "RunSimioPortalExpColsole" and press Enter.   The RunSimioPortalExpConsole will read the values in the RunSimioPortalExpConsole.exe.config when running the executable to determine what actions to take based on its settings.   Here is an example of a successful schedule run.

Note: you will need to update the URL and Personal Access Token in the RunSimioPortalExpConsole.exe.config. See Settings section below. You will also need to upload the SchedulingLaborEfficiencies.spfx example from C:\Program Files\Simio LLC\Simio\Examples onto portal to have the RunSimioPortalExpConfig to run successfully.
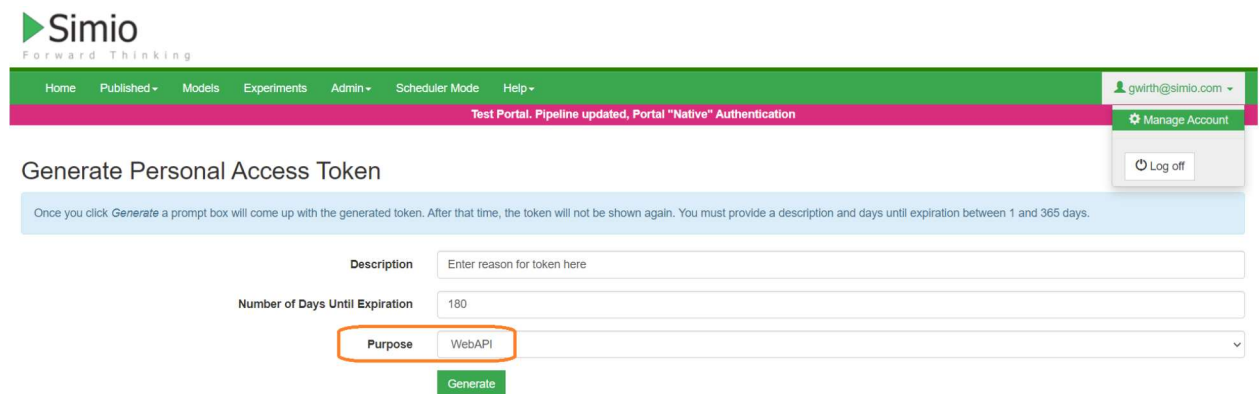
Settings

The settings are used to control the workflow of the RunSimioPortalExpConsole.   These settings are configured in the RunSimioPortalExpConsole.exe.config file. The RunSimioPortalExpConsole.exe.config must be in the same folder as the RunSimioPortalExpConsole.exe.

Here is list of the settings contained in the RunSimioPortalExpConsole.exe.config.   The values contained in the parentheses are command line overrides.   The settings that are in *italic* require Simio Portal to be configured to allow private web API calls…. See Appendix A for steps to configure Simio Portal to allow private web API calls. The next section will cover how to use the command line overrides.

General Settings

> URL (-u) – string – This is the URL so Simio Portal (e.g. https://<YourSimioPortalURL>/).

> Personal Access Token (-pat) – string – This is the personal access token used to authenticate the calling of the Simio Portal Web API.   The personal access tokens are setup under your username under "Manage Account".   From there, select "Generate new token".   Then, enter Description, Number of Days Until Expiration and Purpose. You must choose "WebAPI" as Purpose to use the RunSimioPortalExpConsole.

> Once generated, copy the token text and copy it into this setting.

Copy this token for use in the Simio application, it will not be shown again.

eyJ1IjoiZ3dpcnRoQHNpbWlvLmNvbSIsInQiOiIxMGxpVG9vaXIzSHpSTk82a2h

OK   Cancel
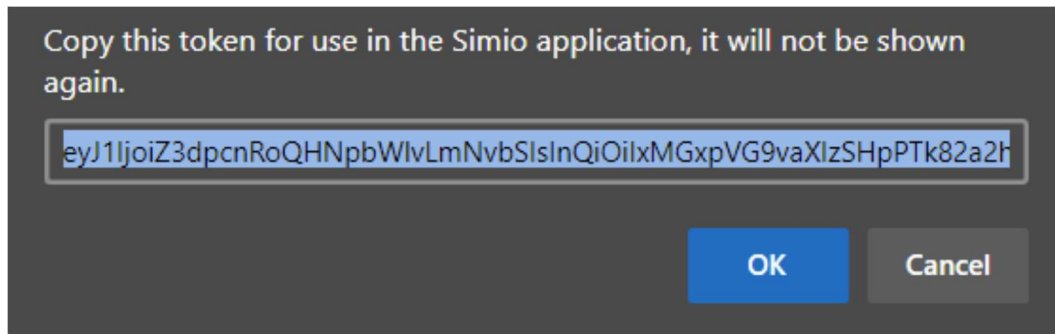
Retry Results Interval Seconds (-ris) – integer – The RunSimioPortalExpConsole will poll the Simio Portal Web API until a run completes. This parameter is the interval that the Simio Portal Web API is polled for results.

Retry Results Max Attempts (-rma) – integer – The number of attempts that the RunSimioPortalExpConsole will try to get results before it returns in error.

WaitAtEnd (-w) – boolean – used to make the console app wait at the end of the run.

Authentication Settings

Authentication Type (-at) – string – the valid types are None and Basic.

Domain (-d) – string – optional parameter to include the domain as part of the authentication.

UserName (-u) – string – optional parameter for user name.

Password (-p) – string – optional parameter for password.   This value is not encrypted.

Bearer Token Refresh Interval Minutes (-btr) – integer – number of minutes that the Simio Portal Bearer Token will be used prior to refreshing.   The default value is 15.   The Simio Portal Bearer Token is valid for 30 minutes.

Project Settings

Project Name (-prj) – string – project name in portal.

Scheduling Only Settings

Run Schedule Plan Scenario Name (-rsn) – string – existing plan name that is used to run the schedule.   You need to have an existing plan name to run the plan. The user must be the owner of the plan to access it.

Run Schedule Control Values Array (-rsc) – string – name values pairs that will set the control value prior to running the plan.   The control value settings are defined as follows "ABC=123|EFG=456" where equals "=" separates the name from the value and pipe "|" separates the control value settings from one another.

Run Schedule (-rs) – boolean -determines whether to run plan.

Run Risk Analysis (-rra) – boolean – determine whether to run risk analysis.

Publish Schedule Run (-psr) – boolean – determines when to publish plan. This can be run independently from running the schedule.

Publish Schedule Name (-psn) – string – name of the publish plan.

Publish Schedule Description (-psd) – string – description of the publish plan.

Run Length Days (-rdl) – integer – number of days to run the model. This value will only be used if the Start Time Selection is set to a valid string and this value is greater than zero.

Start Time Selection (-sts) – string – determines when to start the schedule based on current time type. The valid strings are Second, Minute, Hour, Day, Week, Month and Year. A valid string needs to be entered and the Run Length Days need to be greater than zero

Import All Tables (-iat) – boolean – determines whether to import all tables before scheduling run. Run schedule must be set to true to use this setting.

Export All Tables And Logs (-eat) – determine whether to export all tables. This can be run independently from running the schedule.

Experiments Settings

Run Experiment Run Description (-erd) – string – description of the experiment run. Currently, the RunSimioPortalExpConsole is only setup to run from an existing experiment run.

NOTE: In the RunSimioPortalExpConole.exe.config, the experiment run name is OptQuestRun. This was created from the OptQuest experiment SchedulingLaborEfficiencies example. When run from the RunSimioPortalExpConsole, it will not run OptQuest. This capability is currently not available from this app. It will run the experiment based on the detail in the Run Experiment Scenarios JSON setting.

Run Experiment Scenarios JSON (-rej) – string – JSON string containing the experiment run detail.

Run Experiment (-re) – boolean - determines whether to run experiment.

Publish Experiment Run (-psr) – boolean - determines whether to publish experiment. This can be run independently from running the experiment.

Publish Experiment Run Name (-pen) – string - name of the published experiment.

Publish Experiment Run Description (-ped)- string – description of the published experiment.

Specifying Command Line Settings

Any of the setting can also be set by specifying them as part of the command line run. In addition to the Settings section, you can find the setting from running "RunSimioPortalExpConsole -?". Here are

the results. The results also show the current setting values in the
RunSimioPortalExpConsole.exe.config.

```
Command Prompt - runsimioportalexpconsole  -?                                      —    □    X

Reading Parameters
-url = Portal url  (default = https://mdt.external.simioportal.com/)
-pat = Personal Access Token  (default = eyJ1IjoiZ3dpcnRoQHNpbWlvLmNvbSIsInQiOiIwWXU1eHloOGhtMFhsdXpkM1BJV
jFnWTRpQVJvdVAvOWRCU1ZFVk9Kd0J0UmxWek51OVhCK3kvUU0xMjNLRU1YLzE4NXdvL1pQOFc5KzJxUzlWN3kwZz09In0=)
-ris = Retry Results Interval Seconds  (default = 5)
-rma = Retry Results Max Attemps  (default = 500)
-at = Authentication Type  (default = None)
-d = Domain  (default = )
-u = User Name  (default = )
-p = Password  (default = )
-prj = Project Name  (default = SchedulingLaborEfficiencies)
-rsn = Run Schedule Plan Scenario Name  (default = _ModelValues)
-rsc = Run Schedule Control Values Array  (default = )
-rs = Run Schedule  (default = True)
-rra = Run RiskAnalysis  (default = False)
-psr = Publish Schedule Plan Run (default = False)
-psn = Publish Schedule Name  (default = SchedRunPub)
-psd = Publish Schedule Description  (default = SchedRunPubDesc)
-erd = Run Experiment Run Desc  (default = OptQuestRun)
-rej = Run Experiment Scenareios JSON  (default = { "Scenarios": [{ "Name": "Scenario1", "ReplicationsRequ
ired": 10, "ControlValues": [{ "Name": "ToolQty", "Value": 1}]},{ "Name": "Scenario2", "ReplicationsRequir
ed": 10, "ControlValues": [{ "Name": "ToolQty", "Value": 2}]},{ "Name": "Scenario3", "ReplicationsRequired
": 10, "ControlValues": [{ "Name": "ToolQty", "Value": 3}]},{ "Name": "Scenario4", "ReplicationsRequired":
 10, "ControlValues": [{ "Name": "ToolQty", "Value": 4}]},{ "Name": "Scenario5", "ReplicationsRequired": 1
0, "ControlValues": [{ "Name": "ToolQty", "Value": 5}]}]})
-re = Run Experiment  (default = False)
-per = Publish Experiment Run (default = False)
-pen = Publish Experiment Run Name  (default = ExpPub)
-ped = Publish Experiment Run Description  (default = ExpPubDesc)
-rld = Run Length Days (Scheduling Only) (default = 180)
-iat = Import All Tables (Scheduling Only)  (default = False)
-eat = Export All Tables And Logs (Scheduling Only)  (default = False)
-sts = Start Time Selection (Scheduling Only)  (default = None)
-btr = Bearer Token Refresh Interval Minutes (default = 15)
-w = wait (pause) at end  (default = True)
Press 'Enter' To End
```

To run the existing configuration as an experiment instead of schedule, first you need to create an
experiment run from the OptQuest experiment called "OptQuestRun". Next, the following command
line setting would be applied. "-rs False" will disable the schedule run and "-re True" will enable the
experiment run. The existing configuration already has the Run Experiment Scenarios JSON that will
define the scenarios.

RunSimioPortal -re True -rs False

Here are the results from the run.

```
C:\VSOProjects\SimioLLC\RunSimioPortalExpConsole\Executables\RunSimioPortalExpConsole.exe
Reading Parameters
Obtain Bearer Token
Bearer Token Received Successfully
Find Experiment Ids
ExperimentRunId:155|ExperimentId:88
Start Experiment Run For Experiment
Get Experiment Results Attempt Number = 1
Get Experiment Results Attempt Number = 2
Get Experiment Results Attempt Number = 3
Get Experiment Results Attempt Number = 4
Get Experiment Results Attempt Number = 5
Get Experiment Results Attempt Number = 6
Get Experiment Results Attempt Number = 7
Get Experiment Results Attempt Number = 8
Get Experiment Results Attempt Number = 9
Get Experiment Results Attempt Number = 10
Get Experiment Results Attempt Number = 11
Get Experiment Results Attempt Number = 12
Get Experiment Results Attempt Number = 13
Get Experiment Results Attempt Number = 14
Success
Press 'Enter' To End
```

Appendix A: Allow Private API Calls

Some of the method used in the RunSimioPortalExpConsole use private Simio Web API call.   The method that required the private API available are noted in the Settings section of this document. To use these method calls, the following setting needs to be added to the Simio Portal Web API.   The setting that needs to be added Wed.Config under appSettings. Once added, just save the Web.Config and the private Web API calls should be available. See screenshot below:

<add key="allowPrivateAPICallsKnowingTheAPIMayChangeAndCallersWillThenFail" value="true"/>

Simio
Forward Thinking

```
C:\inetpub\SimioPortal\Web.config - Notepad++ [Administrator]                                    —  □  ×

File  Edit  Search  View  Encoding  Language  Settings  Tools  Macro  Run  Plugins  Window  ?                X

Web.config

31      <section name="settings" type="DevExpress.Web.SettingsConfigurationSection, DevExpress.Web.v21.2, Version=21.2.6.0, Culture=neutral, PublicKeyToken=b88d1754d700e
32      <section name="errors" type="DevExpress.Web.ErrorsConfigurationSection, DevExpress.Web.v21.2, Version=21.2.6.0, Culture=neutral, PublicKeyToken=b88d1754d700e49a"
33      <section name="resources" type="DevExpress.Web.ResourcesConfigurationSection, DevExpress.Web.v21.2, Version=21.2.6.0, Culture=neutral, PublicKeyToken=b88d1754d70
34      </sectionGroup>
35    </configSections>
36    <!-- Enable the section below to generate a trace file for use when troubleshooting runtime problems -->
37    <!--
38    <system.diagnostics>
39      <trace autoflush="true">
40        <listeners>
41          <add type="System.Diagnostics.TextWriterTraceListener" name="FileListener" initializeData="App_Data\SimioPortalTrace.log" />
42          <remove name="Default" />
43        </listeners>
44      </trace>
45    </system.diagnostics>
46    -->
47    <connectionStrings>
48      <!-- This will get set to something during deployment -->
49      <add name="SimioServerDB" connectionString="server=localhost\SqlExpress;database=SimioPortal;Trusted_Connection=True;" providerName="System.Data.SqlClient"/>
50    </connectionStrings>
51    <appSettings file="WebAdditionalSettingsThatArentCommitted.config">
52      <add key="allowPrivateAPICallsKnowingTheAPIMayChangeAndCallersWillThenFail" value="true"/>
53      <add key="webpages:Version" value="3.0.0.0"/>
54      <add key="webpages:Enabled" value="false"/>
55      <add key="PreserveLoginUrl" value="true"/>
56      <add key="ClientValidationEnabled" value="true"/>
57      <add key="UnobtrusiveJavaScriptEnabled" value="true"/>
58    <!--
59    Setting jwt:autogenerate to true will generate values for both the key and issuer (if they are not otherwise provided).
60    This allows for quick setup in single server environments. However, if multi-server/farm environments, at least the jwt:key setting
61    should be set to the same value across servers.
62    -->
63      <add key="jwt:autogenerate" value="true"/>
64    <!--
65    A value for the "jwt:key" setting lets experiments query for run requests and send results back to the portal
66    A single line powershell to give a unique value for this is:
67
68    $bytes = New-Object "System.Byte[]" 64;$rnd = New-Object System.Security.Cryptography.RNGCryptoServiceProvider;$rnd.GetBytes($bytes);[System.Convert]::ToBase64Stri
69

eXtensible Markup Language file           length : 47,149   lines : 667        Ln : 52  Col : 5  Pos : 4,141        Windows (CR LF)    UTF-8    INS
```