How to Instantiate SCAIFE API Calls:

Using SCAIFE Code, the SCAIFE API, Swagger-Editor, and Eventually Using Your Tool with Auto-Generated Client Code

Copyright

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM20-0550

Contents

Prerequisites

You need a Linux machine (or VM) that has:

- around 10 GB extra space (2.5 GB for the expanded files, 6.5GB virtual space for the containers but prior to any activity they use less than 1GB on our test machine, per `docker ps --size`)
- docker installed, from the Docker website https://www.docker.com/resources/what-container (Docker Community Edition, Docker version 19.03.8, build afacb8b7f0 is a tested and working version installed on an Ubuntu version 20.0 LTS (Note that it's best to download the latest stable version for your machine directly from Docker's own website, not to use the default installed docker version on the Linux operating system.

The screenshots in this document are large. **To be able to see them, please view this document in "Web mode" (not "Print mode", which will cut off much of the screenshots).** To switch to Web view in Microsoft Word, select tab View then Icon "Web Layout". Alternatively, click the Web Layout view icon: The only view icons in the bottom-right corner of the Microsoft Word 2019 document window are Read Mode, Print Layout, and web Layout.

To use this document, you need SCAIFE code developed by the SEI (not auto-generated), and provided by the SEI.

For Section *Using Auto-Generated Client Code to Implement SCAIFE API Calls from Your Tool*, some of the instructions refer to using code auto-generated by the SEI. However, that code can alternatively be auto-generated by the reader, as described in that section.

Prepare SCAIFE Code, Start Containers, and Install Swagger-Editor

On the above Linux machine, do the following:

1. Unzip and decrypt the file using the decryption passphrase provided by SEI.
2. Start up the docker containers for SCAIFE by:
    a. changing directories to the `model_api` directory (base directory of the unzipped download)
    b. In the `model_api` directory, run the following command to start the SCAIFE containers:
    ```
    docker-compose -f docker-compose.yml up --build
    ```
3. Install swagger-editor. There are docker containers for swagger-editor. The explanation below of how to install and use was copied from the GitHub swagger-editor README (with slight edit to point user to the directory with the swagger.yaml files): https://github.com/swagger-api/swagger-editor

    There is a docker image published in DockerHub.
    To use this, run the following:
    ```
    docker pull swaggerapi/swagger-editor
    docker run -d -p 80:8080 swaggerapi/swagger-editor
    ```
    This will run Swagger Editor (in detached mode) on port 80 on your machine, so you can open it by navigating to `http://localhost`in your browser.

    Open one of the five `swagger.yaml` definition files on your host, from the folder (substitute a server name `datahub`, `priority`, `ui`, `registration`, or `stats` for `/$SERVER_NAME`), from a filepath that

ends
`$BEGINNING_OF_FILEPATH/model_api/$SERVER_NAME_server_stub/swagger_server/swagger/swagger.yaml`
as follows (edit to substitute the filepath in purple font where you actually placed the `model_api` directory for the SCAIFE code you unzipped.

EXAMPLE command for registration server:

```
docker run -d -p 80:8080 -v $(pwd):/tmp -e
SWAGGER_FILE=/home/username/model_api/registration_server_stub/swagger_server/swagger/swagger.yaml
swaggerapi/swagger-editor
```

Populate Initial Data in SCALe and Other SCAIFE Servers

Use the following command from the `model_api` directory. (The purpose of this command is to prep the system before you start testing your first API calls from your tool or the swagger-editor.) The following command uses a script in the SCALe container that creates a SCALe project, adjudicates (true or false) some meta-alerts, uploads the project to the SCAIFE datahub, then creates a classifier in the SCAIFE stats module, runs the classifier, and then receives the confidence values from the classifier in SCALe:

```
docker exec scale python
scripts/automation/create_manual_test_project.1.microjuliet.py
```

Check to make sure the command completed successfully as expected:

Open your browser to view the new project at `http://localhost:8083/projects/1`
(Browser versions that have been tested and known to work: Firefox v 76.0.1 has been tested and works, other versions may also work. Google Chrome has not been thoroughly tested but version 81.0.4044.138 works for at least this step:.

Scroll to the right in the Meta-alerts list, until you can see the column header "Confidence". There should be values in that field. (In the blue-backed rectangle at top for selecting filtering options, you can select "Sort by:" to be "Confidence", then select the "Filter" button. If you select "Sort Direction" to be ascending ("asc") then the top Confidence value will be 0.0. If you select "Sort Direction to be descending ("desc") then select the "Filter" button again, the top confidence value will be 0.75.

If your check above showed the expected values, then the Project is now successfully uploaded in SCAIFE to be used with SwAT and/or the swagger-editor.

Start to Make API `curl` Calls, Via the Swagger-Editor GUI

All of the following API call requests will be shown via `curl` request statements, so your machine needs `curl` to be installed (e.g., for machines that can install using `apt`, `sudo apt-get install curl` would install `curl`). `curl` is used in command lines or scripts to transfer data. For more information about `curl`, see https://curl.haxx.se/) To begin with, you should run the API calls from the swagger-editor GUI.

Use the Registration Module's API with the Swagger-Editor GUI

First, you will need to load the registration's `swagger.yaml` file into the swagger-editor, via the GUI or with the startup command example above. Your view will look like the following:

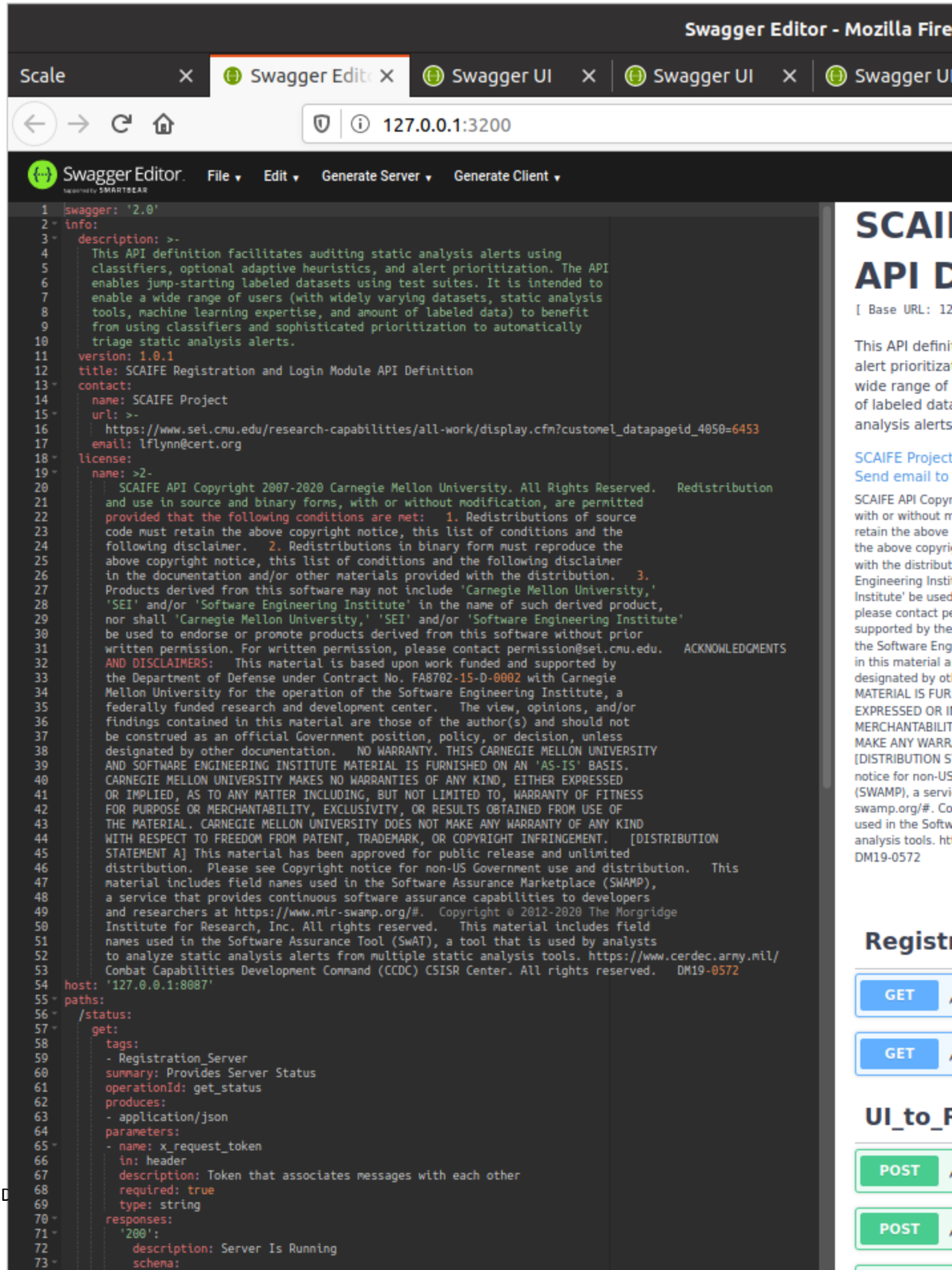

Swagger Editor - Mozilla Fire
Scale
Swagger Edit
Swagger UI
Swagger UI
Swagger UI
127.0.0.1:3200
Swagger Editor
File
Edit
Generate Server
Generate Client
swagger: '2.0'
info:
description: >-
This API definition facilitates auditing static analysis alerts using
classifiers, optional adaptive heuristics, and alert prioritization. The API
enables jump-starting labeled datasets using test suites. It is intended to
enable a wide range of users (with widely varying datasets, static analysis
tools, machine learning expertise, and amount of labeled data) to benefit
from using classifiers and sophisticated prioritization to automatically
triage static analysis alerts.
version: 1.0.1
title: SCAIFE Registration and Login Module API Definition
contact:
name: SCAIFE Project
url: >-
https://www.sei.cmu.edu/research-capabilities/all-work/display.cfm?customel_datapageid_4050=6453
email: lflynn@cert.org
license:
name: >2-
SCAIFE API Copyright 2007-2020 Carnegie Mellon University. All Rights Reserved. Redistribution
and use in source and binary forms, with or without modification, are permitted
provided that the following conditions are met: 1. Redistributions of source
code must retain the above copyright notice, this list of conditions and the
following disclaimer. 2. Redistributions in binary form must reproduce the
above copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the distribution. 3.
Products derived from this software may not include 'Carnegie Mellon University,'
'SEI' and/or 'Software Engineering Institute' in the name of such derived product,
nor shall 'Carnegie Mellon University,' 'SEI' and/or 'Software Engineering Institute'
be used to endorse or promote products derived from this software without prior
written permission. For written permission, please contact permission@sei.cmu.edu. ACKNOWLEDGMENTS
AND DISCLAIMERS: This material is based upon work funded and supported by
the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie
Mellon University for the operation of the Software Engineering Institute, a
federally funded research and development center. The view, opinions, and/or
findings contained in this material are those of the author(s) and should not
be construed as an official Government position, policy, or decision, unless
designated by other documentation. NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY
AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN 'AS-IS' BASIS.
CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS
FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF
THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND
WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT. [DISTRIBUTION
STATEMENT A] This material has been approved for public release and unlimited
distribution. Please see Copyright notice for non-US Government use and distribution. This
material includes field names used in the Software Assurance Marketplace (SWAMP),
a service that provides continuous software assurance capabilities to developers
and researchers at https://www.mir-swamp.org/#. Copyright © 2012-2020 The Morgridge
Institute for Research, Inc. All rights reserved. This material includes field
names used in the Software Assurance Tool (SwAT), a tool that is used by analysts
to analyze static analysis alerts from multiple static analysis tools. https://www.cerdec.army.mil/
Combat Capabilities Development Command (CCDC) CSISR Center. All rights reserved. DM19-0572
host: '127.0.0.1:8087'
paths:
/status:
get:
tags:
- Registration_Server
summary: Provides Server Status
operationId: get_status
produces:
- application/json
parameters:
- name: x_request_token
in: header
description: Token that associates messages with each other
required: true
type: string
responses:
'200':
description: Server Is Running
schema:
SCAI
API D
[Base URL: 12
This API defini
alert prioritiza
wide range of
of labeled data
analysis alerts
SCAIFE Project
Send email to
SCAIFE API Copyr
with or without m
retain the above
the above copyri
with the distribut
Engineering Insti
Institute' be used
please contact pe
supported by the
the Software Eng
in this material a
designated by ot
MATERIAL IS FUR
EXPRESSED OR I
MERCHANTABILIT
MAKE ANY WARR
[DISTRIBUTION S
notice for non-US
(SWAMP), a servi
swamp.org/#. Co
used in the Softw
analysis tools. ht
DM19-0572
Regist
GET
GET
UI_to_R
POST
POST

Figure 1: Load Registration Module's API into the Swagger-Editor GUI

First, you should register a user in SCAIFE. In the swagger-editor GUI, under the section "UI_to_Registration", select the `POST /register` API call and it will expand. Now your view (on right) looks like the following:

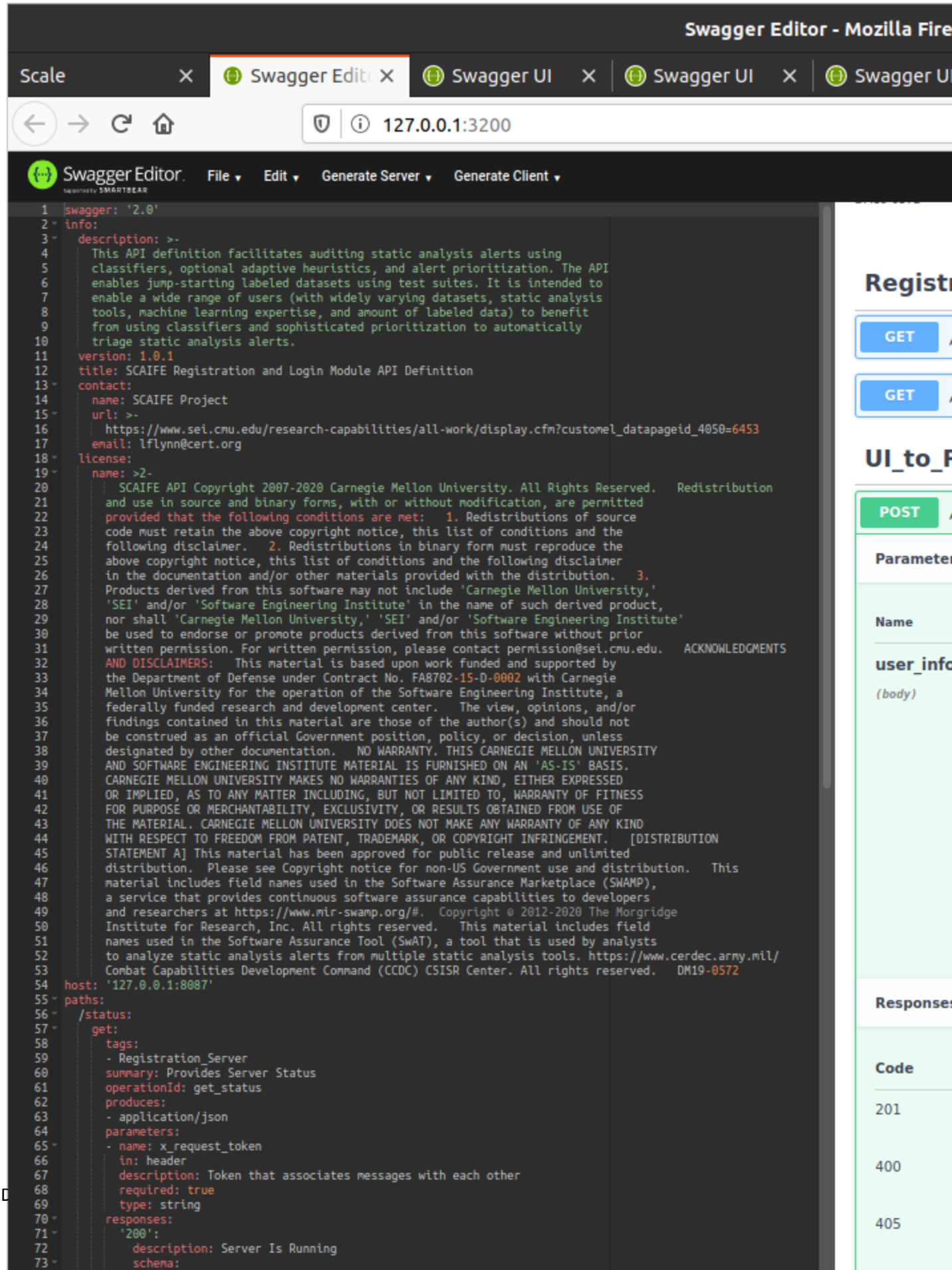

Swagger Editor - Mozilla Fire
Scale
Swagger Edit
Swagger UI
Swagger UI
Swagger UI
127.0.0.1:3200
Swagger Editor
File
Edit
Generate Server
Generate Client
swagger: '2.0'
info:
description: >-
This API definition facilitates auditing static analysis alerts using
classifiers, optional adaptive heuristics, and alert prioritization. The API
enables jump-starting labeled datasets using test suites. It is intended to
enable a wide range of users (with widely varying datasets, static analysis
tools, machine learning expertise, and amount of labeled data) to benefit
from using classifiers and sophisticated prioritization to automatically
triage static analysis alerts.
version: 1.0.1
title: SCAIFE Registration and Login Module API Definition
contact:
name: SCAIFE Project
url: >-
https://www.sei.cmu.edu/research-capabilities/all-work/display.cfm?customel_datapageid_4050=6453
email: lflynn@cert.org
license:
name: >2-
SCAIFE API Copyright 2007-2020 Carnegie Mellon University. All Rights Reserved. Redistribution
and use in source and binary forms, with or without modification, are permitted
provided that the following conditions are met: 1. Redistributions of source
code must retain the above copyright notice, this list of conditions and the
following disclaimer. 2. Redistributions in binary form must reproduce the
above copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the distribution. 3.
Products derived from this software may not include 'Carnegie Mellon University,'
'SEI' and/or 'Software Engineering Institute' in the name of such derived product,
nor shall 'Carnegie Mellon University,' 'SEI' and/or 'Software Engineering Institute'
be used to endorse or promote products derived from this software without prior
written permission. For written permission, please contact permission@sei.cmu.edu. ACKNOWLEDGMENTS
AND DISCLAIMERS: This material is based upon work funded and supported by
the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie
Mellon University for the operation of the Software Engineering Institute, a
federally funded research and development center. The view, opinions, and/or
findings contained in this material are those of the author(s) and should not
be construed as an official Government position, policy, or decision, unless
designated by other documentation. NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY
AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN 'AS-IS' BASIS.
CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS
FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF
THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND
WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT. [DISTRIBUTION
STATEMENT A] This material has been approved for public release and unlimited
distribution. Please see Copyright notice for non-US Government use and distribution. This
material includes field names used in the Software Assurance Marketplace (SWAMP),
a service that provides continuous software assurance capabilities to developers
and researchers at https://www.mir-swamp.org/#. Copyright © 2012-2020 The Morgridge
Institute for Research, Inc. All rights reserved. This material includes field
names used in the Software Assurance Tool (SwAT), a tool that is used by analysts
to analyze static analysis alerts from multiple static analysis tools. https://www.cerdec.army.mil/
Combat Capabilities Development Command (CCDC) CSISR Center. All rights reserved. DM19-0572
host: '127.0.0.1:8087'
paths:
/status:
get:
tags:
- Registration_Server
summary: Provides Server Status
operationId: get_status
produces:
- application/json
parameters:
- name: x_request_token
in: header
description: Token that associates messages with each other
required: true
type: string
responses:
'200':
description: Server Is Running
schema:
Regist
GET
GET
UI_to_
POST
Parameter
Name
user_info
(body)
Response
Code
201
400
405

Figure 2: Register a SCAIFE User, via the Swagger-Editor GUI

Select the button “Try it out” (top-right). That expands the options. Keep the default values. Next, select the blue button “Execute”

The screenshot below shows the successful results on the right, after execution. Note under “Server Response”, it shows “201” and “User Created”, indicating success.

Also, note that the screenshot below also shows the exact Curl command that was used (which could be used by your code in SwAT, but likely you will want to use the Java client code that has been auto-generated for you instead). That Curl command (which uses the default values that you kept) is:

```
curl -X POST "http://127.0.0.1:8087/register" -H  "accept:
application/json" -H  "Content-Type: application/json" -d "{
\"first_name\": \"Ebonie\",  \"last_name\": \"McNeil\",
\"organization_name\": \"Software Engineering Institute\",
\"username\": \"username\",  \"password\": \"password\"}"
```

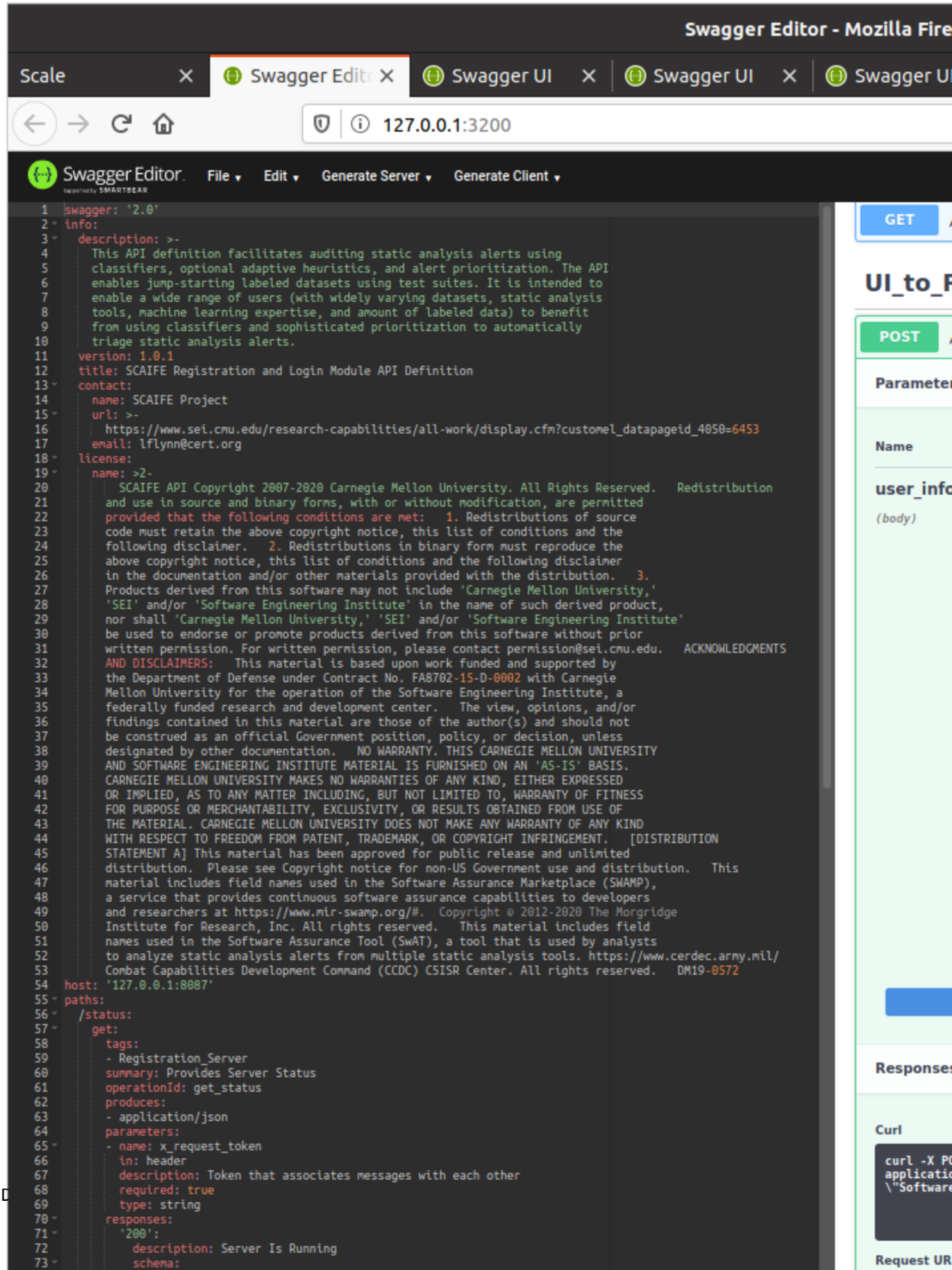

Swagger Editor - Mozilla Fire
Scale
Swagger Edit
Swagger UI
Swagger UI
Swagger UI
127.0.0.1:3200
Swagger Editor
File
Edit
Generate Server
Generate Client
swagger: '2.0'
info:
description: >-
This API definition facilitates auditing static analysis alerts using
classifiers, optional adaptive heuristics, and alert prioritization. The API
enables jump-starting labeled datasets using test suites. It is intended to
enable a wide range of users (with widely varying datasets, static analysis
tools, machine learning expertise, and amount of labeled data) to benefit
from using classifiers and sophisticated prioritization to automatically
triage static analysis alerts.
version: 1.0.1
title: SCAIFE Registration and Login Module API Definition
contact:
name: SCAIFE Project
url: >-
https://www.sei.cmu.edu/research-capabilities/all-work/display.cfm?customel_datapageid_4050=6453
email: lflynn@cert.org
license:
name: >2-
SCAIFE API Copyright 2007-2020 Carnegie Mellon University. All Rights Reserved. Redistribution
and use in source and binary forms, with or without modification, are permitted
provided that the following conditions are met: 1. Redistributions of source
code must retain the above copyright notice, this list of conditions and the
following disclaimer. 2. Redistributions in binary form must reproduce the
above copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the distribution. 3.
Products derived from this software may not include 'Carnegie Mellon University,'
'SEI' and/or 'Software Engineering Institute' in the name of such derived product,
nor shall 'Carnegie Mellon University,' 'SEI' and/or 'Software Engineering Institute'
be used to endorse or promote products derived from this software without prior
written permission. For written permission, please contact permission@sei.cmu.edu. ACKNOWLEDGMENTS
AND DISCLAIMERS: This material is based upon work funded and supported by
the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie
Mellon University for the operation of the Software Engineering Institute, a
federally funded research and development center. The view, opinions, and/or
findings contained in this material are those of the author(s) and should not
be construed as an official Government position, policy, or decision, unless
designated by other documentation. NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY
AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN 'AS-IS' BASIS.
CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS
FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF
THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND
WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT. [DISTRIBUTION
STATEMENT A] This material has been approved for public release and unlimited
distribution. Please see Copyright notice for non-US Government use and distribution. This
material includes field names used in the Software Assurance Marketplace (SWAMP),
a service that provides continuous software assurance capabilities to developers
and researchers at https://www.mir-swamp.org/#. Copyright © 2012-2020 The Morgridge
Institute for Research, Inc. All rights reserved. This material includes field
names used in the Software Assurance Tool (SwAT), a tool that is used by analysts
to analyze static analysis alerts from multiple static analysis tools. https://www.cerdec.army.mil/
Combat Capabilities Development Command (CCDC) CSISR Center. All rights reserved. DM19-0572
host: '127.0.0.1:8087'
paths:
/status:
get:
tags:
- Registration_Server
summary: Provides Server Status
operationId: get_status
produces:
- application/json
parameters:
- name: x_request_token
in: header
description: Token that associates messages with each other
required: true
type: string
responses:
'200':
description: Server Is Running
schema:
GET
POST
Parameter
Name
user_info
(body)
Response
Curl
curl -X P
applicatio
\"Software
Request UR

Figure 3: Results of Successful Execution of SCAIFE User Registration, via Swagger-Editor

Select the `POST /registration` API call in the GUI again, so it compresses the view to just that API name again.

To send tool output (or full SCAIFE projects, etc.) to SCAIFE, first SwAT (or the user of swagger-editor) must log into the Registration module (`POST /login`) to retrieve an access_token for the DataHub. In the swagger-editor GUI, under the section "UI_to_Registration", select the `POST /login` API call and it will expand. Now your view (on right) looks like the following:

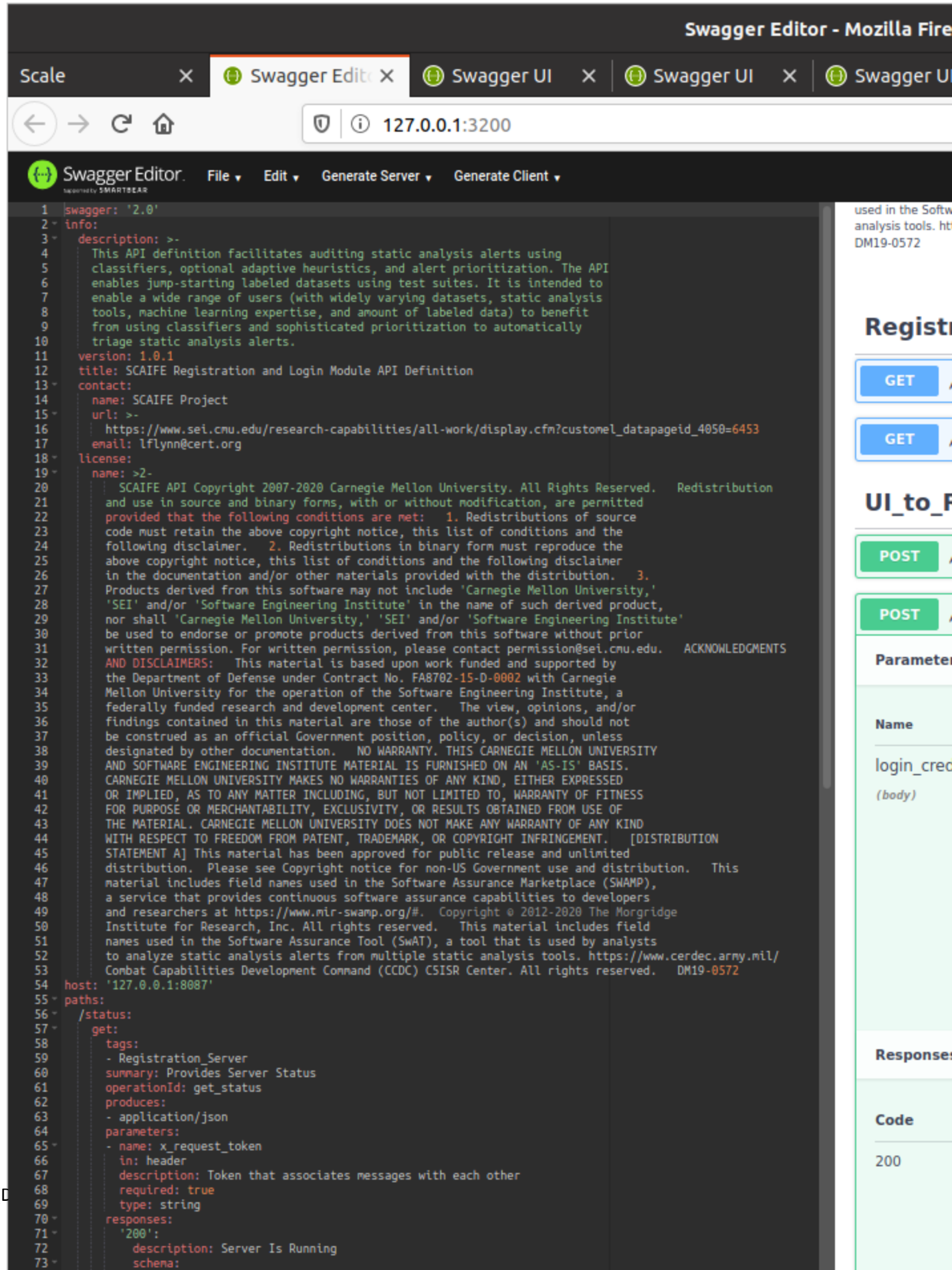

Swagger Editor - Mozilla Fire
Scale
Swagger Edit
Swagger UI
Swagger UI
Swagger UI
127.0.0.1:3200
Swagger Editor
File
Edit
Generate Server
Generate Client
swagger: '2.0'
info:
description: >-
This API definition facilitates auditing static analysis alerts using
classifiers, optional adaptive heuristics, and alert prioritization. The API
enables jump-starting labeled datasets using test suites. It is intended to
enable a wide range of users (with widely varying datasets, static analysis
tools, machine learning expertise, and amount of labeled data) to benefit
from using classifiers and sophisticated prioritization to automatically
triage static analysis alerts.
version: 1.0.1
title: SCAIFE Registration and Login Module API Definition
contact:
name: SCAIFE Project
url: >-
https://www.sei.cmu.edu/research-capabilities/all-work/display.cfm?customel_datapageid_4050=6453
email: lflynn@cert.org
license:
name: >2-
SCAIFE API Copyright 2007-2020 Carnegie Mellon University. All Rights Reserved. Redistribution
and use in source and binary forms, with or without modification, are permitted
provided that the following conditions are met: 1. Redistributions of source
code must retain the above copyright notice, this list of conditions and the
following disclaimer. 2. Redistributions in binary form must reproduce the
above copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the distribution. 3.
Products derived from this software may not include 'Carnegie Mellon University,'
'SEI' and/or 'Software Engineering Institute' in the name of such derived product,
nor shall 'Carnegie Mellon University,' 'SEI' and/or 'Software Engineering Institute'
be used to endorse or promote products derived from this software without prior
written permission. For written permission, please contact permission@sei.cmu.edu. ACKNOWLEDGMENTS
AND DISCLAIMERS: This material is based upon work funded and supported by
the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie
Mellon University for the operation of the Software Engineering Institute, a
federally funded research and development center. The view, opinions, and/or
findings contained in this material are those of the author(s) and should not
be construed as an official Government position, policy, or decision, unless
designated by other documentation. NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY
AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN 'AS-IS' BASIS.
CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS
FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF
THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND
WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT. [DISTRIBUTION
STATEMENT A] This material has been approved for public release and unlimited
distribution. Please see Copyright notice for non-US Government use and distribution. This
material includes field names used in the Software Assurance Marketplace (SWAMP),
a service that provides continuous software assurance capabilities to developers
and researchers at https://www.mir-swamp.org/#. Copyright © 2012-2020 The Morgridge
Institute for Research, Inc. All rights reserved. This material includes field
names used in the Software Assurance Tool (SwAT), a tool that is used by analysts
to analyze static analysis alerts from multiple static analysis tools. https://www.cerdec.army.mil/
Combat Capabilities Development Command (CCDC) CSISR Center. All rights reserved. DM19-0572
host: '127.0.0.1:8087'
paths:
/status:
get:
tags:
- Registration_Server
summary: Provides Server Status
operationId: get_status
produces:
- application/json
parameters:
- name: x_request_token
in: header
description: Token that associates messages with each other
required: true
type: string
responses:
'200':
description: Server Is Running
schema:
used in the Softw
analysis tools. ht
DM19-0572
Regist
GET
GET
UI_to_R
POST
POST
Parameter
Name
login_cred
(body)
Response
Code
200

Figure 4: SCAIFE Login API Call Options, via Swagger-Editor GUI

Select the button “Try it out” (top-right). That expands the options.
This time, you may need to edit from the default username and password (since the default here may be different than the default used in `POST /registration`). Values you should use:

```
"username": "username",
"password": "password"
```

Next, select the blue button “Execute”.

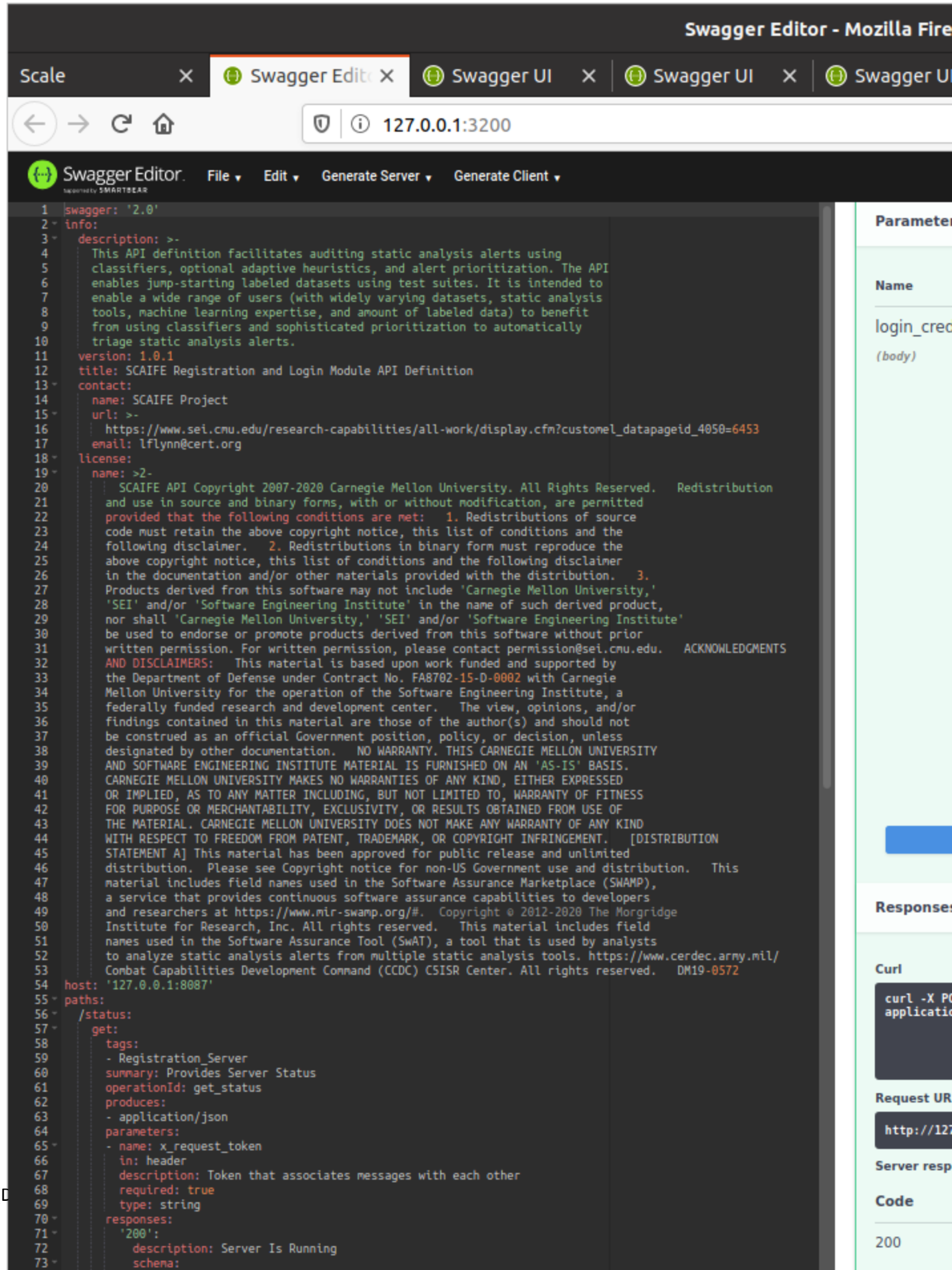

Swagger Editor - Mozilla Fire
Scale
Swagger Edit
Swagger UI
Swagger UI
Swagger UI
127.0.0.1:3200
Swagger Editor
File
Edit
Generate Server
Generate Client
swagger: '2.0'
info:
description: >-
This API definition facilitates auditing static analysis alerts using
classifiers, optional adaptive heuristics, and alert prioritization. The API
enables jump-starting labeled datasets using test suites. It is intended to
enable a wide range of users (with widely varying datasets, static analysis
tools, machine learning expertise, and amount of labeled data) to benefit
from using classifiers and sophisticated prioritization to automatically
triage static analysis alerts.
version: 1.0.1
title: SCAIFE Registration and Login Module API Definition
contact:
name: SCAIFE Project
url: >-
https://www.sei.cmu.edu/research-capabilities/all-work/display.cfm?customel_datapageid_4050=6453
email: lflynn@cert.org
license:
name: >2-
SCAIFE API Copyright 2007-2020 Carnegie Mellon University. All Rights Reserved. Redistribution
and use in source and binary forms, with or without modification, are permitted
provided that the following conditions are met: 1. Redistributions of source
code must retain the above copyright notice, this list of conditions and the
following disclaimer. 2. Redistributions in binary form must reproduce the
above copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the distribution. 3.
Products derived from this software may not include 'Carnegie Mellon University,'
'SEI' and/or 'Software Engineering Institute' in the name of such derived product,
nor shall 'Carnegie Mellon University,' 'SEI' and/or 'Software Engineering Institute'
be used to endorse or promote products derived from this software without prior
written permission. For written permission, please contact permission@sei.cmu.edu. ACKNOWLEDGMENTS
AND DISCLAIMERS: This material is based upon work funded and supported by
the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie
Mellon University for the operation of the Software Engineering Institute, a
federally funded research and development center. The view, opinions, and/or
findings contained in this material are those of the author(s) and should not
be construed as an official Government position, policy, or decision, unless
designated by other documentation. NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY
AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN 'AS-IS' BASIS.
CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS
FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF
THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND
WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT. [DISTRIBUTION
STATEMENT A] This material has been approved for public release and unlimited
distribution. Please see Copyright notice for non-US Government use and distribution. This
material includes field names used in the Software Assurance Marketplace (SWAMP),
a service that provides continuous software assurance capabilities to developers
and researchers at https://www.mir-swamp.org/#. Copyright © 2012-2020 The Morgridge
Institute for Research, Inc. All rights reserved. This material includes field
names used in the Software Assurance Tool (SwAT), a tool that is used by analysts
to analyze static analysis alerts from multiple static analysis tools. https://www.cerdec.army.mil/
Combat Capabilities Development Command (CCDC) CSISR Center. All rights reserved. DM19-0572
host: '127.0.0.1:8087'
paths:
/status:
get:
tags:
- Registration_Server
summary: Provides Server Status
operationId: get_status
produces:
- application/json
parameters:
- name: x_request_token
in: header
description: Token that associates messages with each other
required: true
type: string
responses:
'200':
description: Server Is Running
schema:
Parameter
Name
login_cred
(body)
Response
Curl
curl -X P
applicatio
Request UR
http://127
Server resp
Code
200

Figure 5: Results of Successful Execution of SCAIFE Login, via Swagger-Editor GUI

Note in the screenshot above that the return value is successful "200" and the Curl command used is shown:

```
curl -X POST "http://127.0.0.1:8087/login" -H "accept:
application/json" -H "Content-Type: application/json" -d
"{\"username\": \"username\", \"password\": \"password\"}"
```

The access token returned (user access token) will be used in the subsequent call to send tool output to the DataHub.

Copy the access token from the swagger-editor interface, to a file.
NOTE: Later, you will do this within SwAT. After making this client API call, SwAT should store the access token.

In this example, the access token value is:

```
"x_access_token":
"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6InVzZXJuYW1
lIiwib3JnYW5pemF0aW9uX2lkIjoiNWYwNDE0YTc4OWU3NGUxYWFlYWQ1YzI2Iiwi
ZXhwIjoxNTk0MTkwMzM5fQ.AFytH_VQ_eSk6G37nCcGjDHw0rxagqKXaw1n_GB6Ta
0"
```

Next:

- Click on `GET /servers/{server_name}`
- Click `Try it out` button
- Enter the user access_token copied in previous step
- Enter `datahub` for `server_name`
- Click `Execute` button
- Copy the `x_access_token` that is returned, to use it for the rest of the tests

Swagger-editor GUI shows that the curl command used was:

```
curl -X GET "http://127.0.0.1:8087/servers/datahub" -H  "accept:
application/json" -H  "x_access_token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6InVzZXJuYW1lIiwib
3JnYW5pemF0aW9uX2lkIjoiNWYwNDE0YTc4OWU3NGUxYWFlYWQ1YzI2IiwiZXhwIjoxNTk
0MTkwMzM5fQ.AFytH_VQ_eSk6G37nCcGjDHw0rxagqKXaw1n_GB6Ta0"
```

A screenshot of the successful call is shown below:

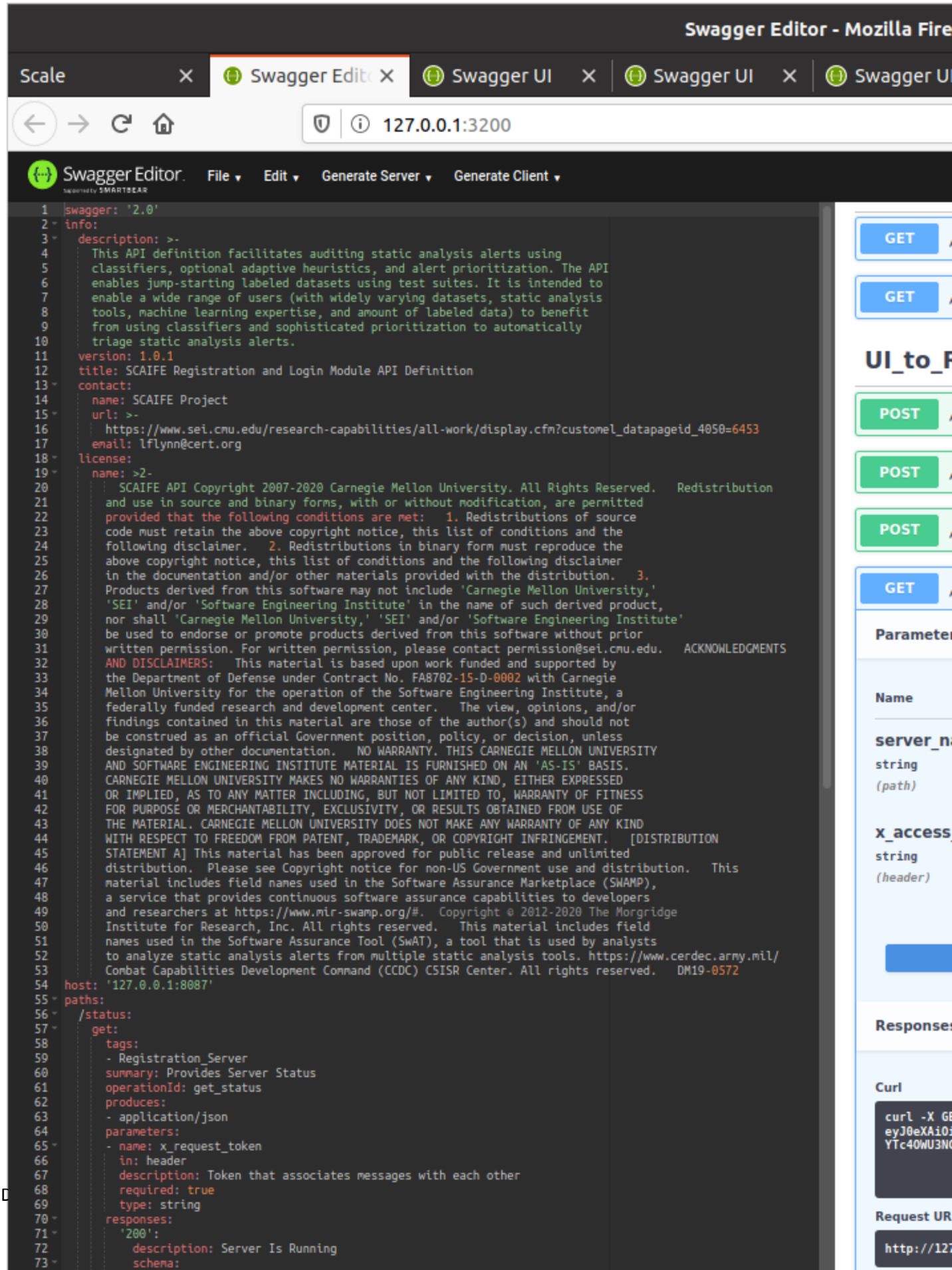

Swagger Editor - Mozilla Fire
Scale
Swagger Edit
Swagger UI
Swagger UI
Swagger UI
127.0.0.1:3200
Swagger Editor
File
Edit
Generate Server
Generate Client
swagger: '2.0'
info:
description: >-
This API definition facilitates auditing static analysis alerts using
classifiers, optional adaptive heuristics, and alert prioritization. The API
enables jump-starting labeled datasets using test suites. It is intended to
enable a wide range of users (with widely varying datasets, static analysis
tools, machine learning expertise, and amount of labeled data) to benefit
from using classifiers and sophisticated prioritization to automatically
triage static analysis alerts.
version: 1.0.1
title: SCAIFE Registration and Login Module API Definition
contact:
name: SCAIFE Project
url: >-
https://www.sei.cmu.edu/research-capabilities/all-work/display.cfm?customel_datapageid_4050=6453
email: lflynn@cert.org
license:
name: >2-
SCAIFE API Copyright 2007-2020 Carnegie Mellon University. All Rights Reserved. Redistribution
and use in source and binary forms, with or without modification, are permitted
provided that the following conditions are met: 1. Redistributions of source
code must retain the above copyright notice, this list of conditions and the
following disclaimer. 2. Redistributions in binary form must reproduce the
above copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the distribution. 3.
Products derived from this software may not include 'Carnegie Mellon University,'
'SEI' and/or 'Software Engineering Institute' in the name of such derived product,
nor shall 'Carnegie Mellon University,' 'SEI' and/or 'Software Engineering Institute'
be used to endorse or promote products derived from this software without prior
written permission. For written permission, please contact permission@sei.cmu.edu. ACKNOWLEDGMENTS
AND DISCLAIMERS: This material is based upon work funded and supported by
the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie
Mellon University for the operation of the Software Engineering Institute, a
federally funded research and development center. The view, opinions, and/or
findings contained in this material are those of the author(s) and should not
be construed as an official Government position, policy, or decision, unless
designated by other documentation. NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY
AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN 'AS-IS' BASIS.
CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED
OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS
FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF
THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND
WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT. [DISTRIBUTION
STATEMENT A] This material has been approved for public release and unlimited
distribution. Please see Copyright notice for non-US Government use and distribution. This
material includes field names used in the Software Assurance Marketplace (SWAMP),
a service that provides continuous software assurance capabilities to developers
and researchers at https://www.mir-swamp.org/#. Copyright © 2012-2020 The Morgridge
Institute for Research, Inc. All rights reserved. This material includes field
names used in the Software Assurance Tool (SwAT), a tool that is used by analysts
to analyze static analysis alerts from multiple static analysis tools. https://www.cerdec.army.mil/
Combat Capabilities Development Command (CCDC) CSISR Center. All rights reserved. DM19-0572
host: '127.0.0.1:8087'
paths:
/status:
get:
tags:
- Registration_Server
summary: Provides Server Status
operationId: get_status
produces:
- application/json
parameters:
- name: x_request_token
in: header
description: Token that associates messages with each other
required: true
type: string
responses:
'200':
description: Server Is Running
schema:
GET
GET
POST
POST
POST
GET
Parameter
Name
server_n
string
(path)
x_access
string
(header)
Response
Curl
curl -X G
eyJ0eXAiO
YTc4OWU3N
Request UR
http://12

Figure 6: Successful Execution of Getting SCAIFE Server Access Token, via Swagger-Editor GUI

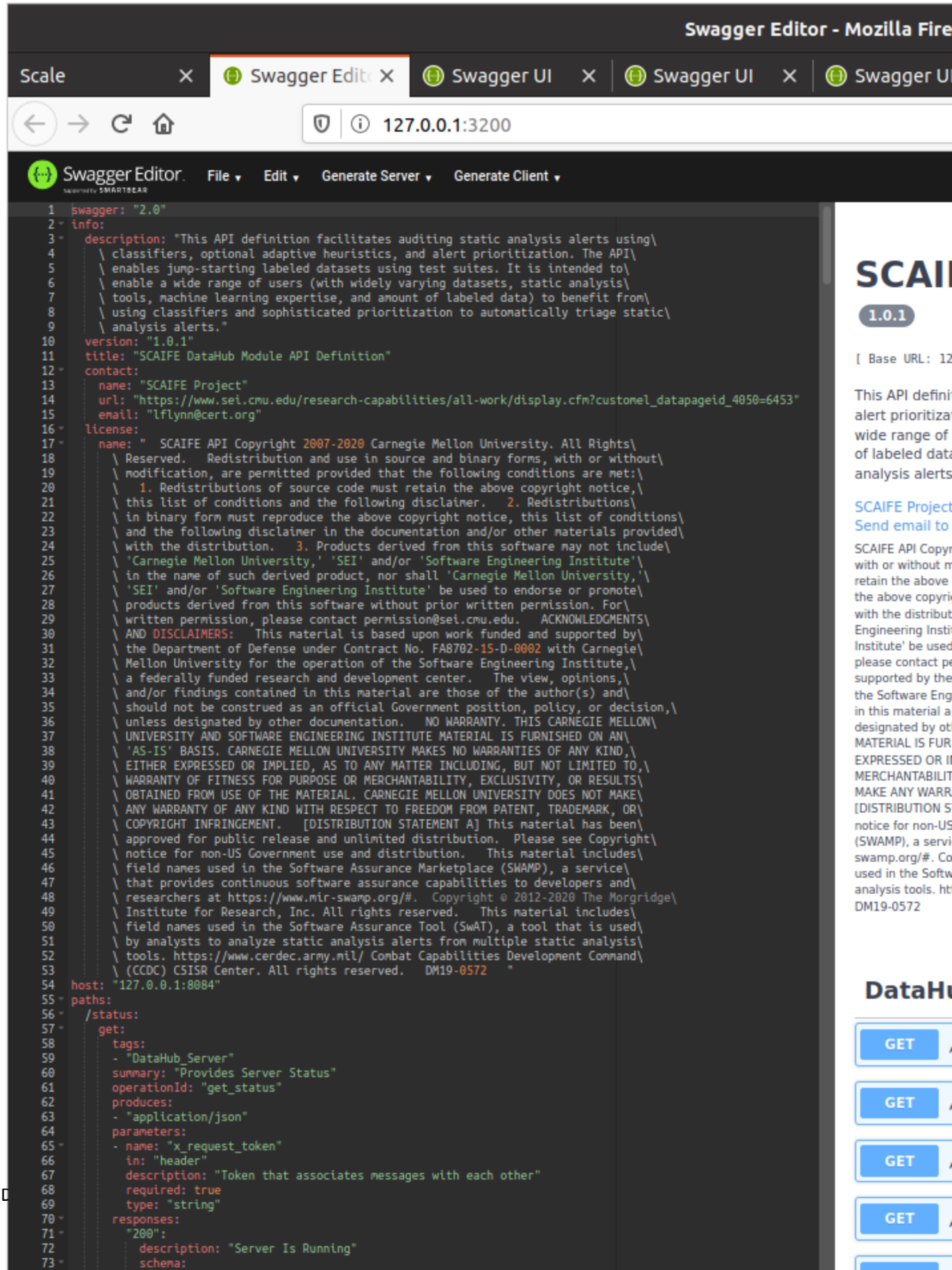
Note in the screenshot above, the token received in the response body is:

```
  "x_access_token":
"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6InVzZXJuYW1lIiwi
b3JnYW5pemF0aW9uX2lkIjoiNWYwNDE0YTc4OWU3NGUxYWFlYWQ1YzI2Iiwic2VydmVyX2
tleSI6ImRhdGFodWJrZXkiLCJleHAiOjE1OTQxOTE3NDh9.jo2YfR0wzyznjeqW-
p4QykxMHCg6-Rt6njhZD5ip8bM”
```

Use the DataHub Module’s API with the Swagger-Editor GUI

Next, load the DataHub’s `swagger.yaml` file into the swagger-editor GUI. From the menubar (top-left) select `File->Import file` and then navigate to `model_api/datahub_server_stub/swagger_server/swagger/swagger.yaml`

See initial view screenshotted below:


Swagger Editor - Mozilla Fire
Scale
Swagger Edit
Swagger UI
Swagger UI
Swagger UI
127.0.0.1:3200
Swagger Editor
File
Edit
Generate Server
Generate Client
swagger: "2.0"
info:
description: "This API definition facilitates auditing static analysis alerts using\
\ classifiers, optional adaptive heuristics, and alert prioritization. The API\
\ enables jump-starting labeled datasets using test suites. It is intended to\
\ enable a wide range of users (with widely varying datasets, static analysis\
\ tools, machine learning expertise, and amount of labeled data) to benefit from\
\ using classifiers and sophisticated prioritization to automatically triage static\
\ analysis alerts."
version: "1.0.1"
title: "SCAIFE DataHub Module API Definition"
contact:
name: "SCAIFE Project"
url: "https://www.sei.cmu.edu/research-capabilities/all-work/display.cfm?customel_datapageid_4050=6453"
email: "lflynn@cert.org"
license:
name: " SCAIFE API Copyright 2007-2020 Carnegie Mellon University. All Rights\
\ Reserved. Redistribution and use in source and binary forms, with or without\
\ modification, are permitted provided that the following conditions are met:\
\ 1. Redistributions of source code must retain the above copyright notice,\
\ this list of conditions and the following disclaimer. 2. Redistributions\
\ in binary form must reproduce the above copyright notice, this list of conditions\
\ and the following disclaimer in the documentation and/or other materials provided\
\ with the distribution. 3. Products derived from this software may not include\
\ 'Carnegie Mellon University,' 'SEI' and/or 'Software Engineering Institute'\
\ in the name of such derived product, nor shall 'Carnegie Mellon University,'\
\ 'SEI' and/or 'Software Engineering Institute' be used to endorse or promote\
\ products derived from this software without prior written permission. For\
\ written permission, please contact permission@sei.cmu.edu. ACKNOWLEDGMENTS\
\ AND DISCLAIMERS: This material is based upon work funded and supported by\
\ the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie\
\ Mellon University for the operation of the Software Engineering Institute,\
\ a federally funded research and development center. The view, opinions,\
\ and/or findings contained in this material are those of the author(s) and\
\ should not be construed as an official Government position, policy, or decision,\
\ unless designated by other documentation. NO WARRANTY. THIS CARNEGIE MELLON\
\ UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN\
\ 'AS-IS' BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND,\
\ EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO,\
\ WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS\
\ OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE\
\ ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR\
\ COPYRIGHT INFRINGEMENT. [DISTRIBUTION STATEMENT A] This material has been\
\ approved for public release and unlimited distribution. Please see Copyright\
\ notice for non-US Government use and distribution. This material includes\
\ field names used in the Software Assurance Marketplace (SWAMP), a service\
\ that provides continuous software assurance capabilities to developers and\
\ researchers at https://www.mir-swamp.org/#. Copyright © 2012-2020 The Morgridge\
\ Institute for Research, Inc. All rights reserved. This material includes\
\ field names used in the Software Assurance Tool (SwAT), a tool that is used\
\ by analysts to analyze static analysis alerts from multiple static analysis\
\ tools. https://www.cerdec.army.mil/ Combat Capabilities Development Command\
\ (CCDC) CSISR Center. All rights reserved. DM19-0572 "
host: "127.0.0.1:8084"
paths:
/status:
get:
tags:
- "DataHub_Server"
summary: "Provides Server Status"
operationId: "get_status"
produces:
- "application/json"
parameters:
- name: "x_request_token"
in: "header"
description: "Token that associates messages with each other"
required: true
type: "string"
responses:
"200":
description: "Server Is Running"
schema:
SCAI
1.0.1
[Base URL: 12
This API defini
alert prioritiza
wide range of
of labeled data
analysis alerts
SCAIFE Project
Send email to
SCAIFE API Copyr
with or without m
retain the above
the above copyri
with the distribut
Engineering Insti
Institute' be used
please contact pe
supported by the
the Software Eng
in this material a
designated by ot
MATERIAL IS FUR
EXPRESSED OR I
MERCHANTABILIT
MAKE ANY WARR
[DISTRIBUTION S
notice for non-US
(SWAMP), a servi
swamp.org/#. Co
used in the Softw
analysis tools. ht
DM19-0572
DataHu
GET
GET
GET
GET

Next :

- Get the package id
    - Navigate to Datahub_Server `GET /packages`
    - Click `Try it out`
    - Enter `x_access_token`, and any value for `x_request_token` (e.g., for the latter, "`abc`")
    - Click `Execute` and retrieve the `package_id`

See screenshot below after successful execution (Code "200" and the Response body includes `package_id` of `5f04041a823a3994fe4a08f7` Since we've only uploaded one SCAIFE package (using that SCALe script at the beginning), we take the only package_id that is in the response.

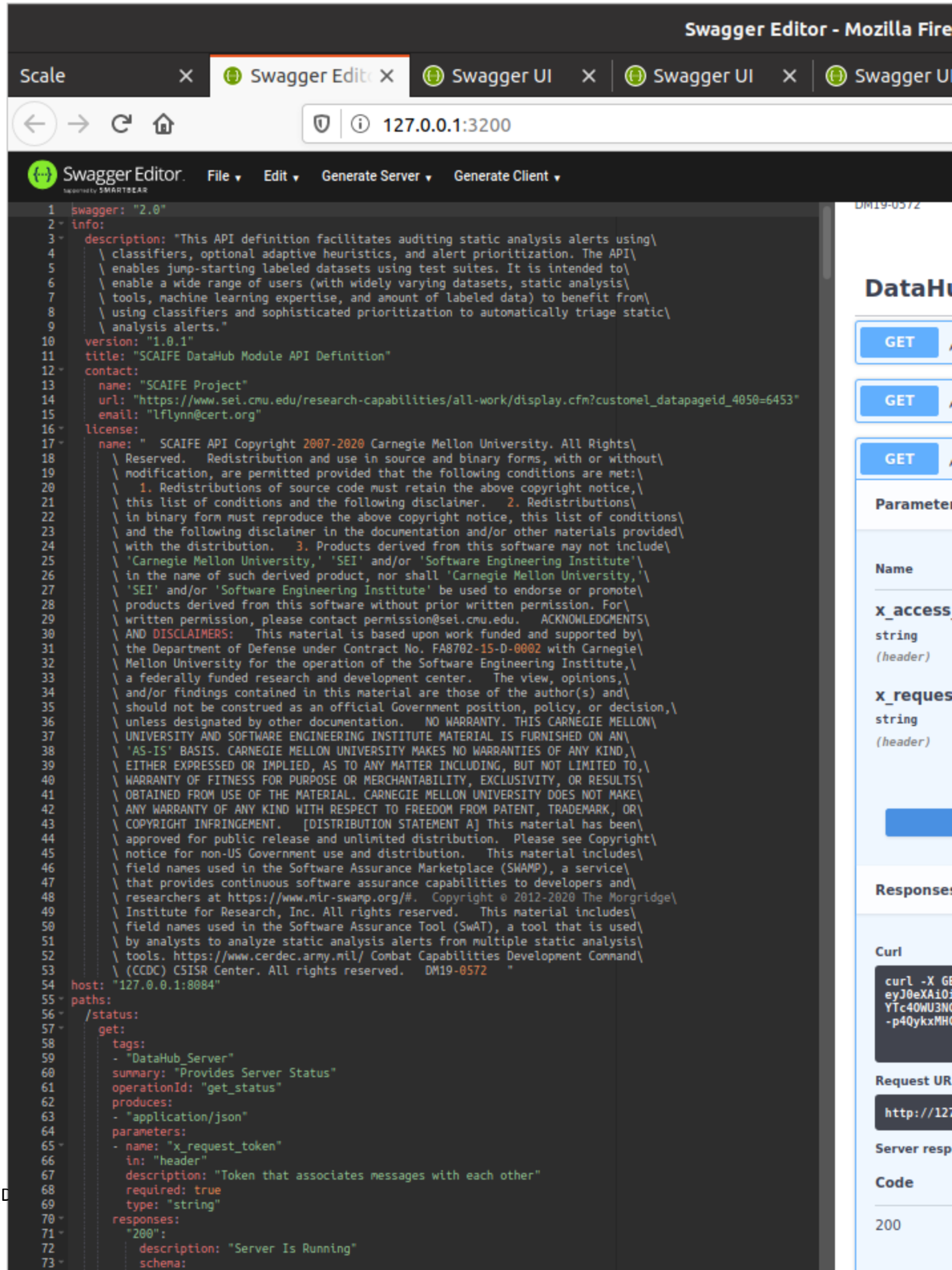

Swagger Editor - Mozilla Fire
Scale
Swagger Edit
Swagger UI
Swagger UI
Swagger UI
127.0.0.1:3200
Swagger Editor
File
Edit
Generate Server
Generate Client
swagger: "2.0"
info:
description: "This API definition facilitates auditing static analysis alerts using\
\ classifiers, optional adaptive heuristics, and alert prioritization. The API\
\ enables jump-starting labeled datasets using test suites. It is intended to\
\ enable a wide range of users (with widely varying datasets, static analysis\
\ tools, machine learning expertise, and amount of labeled data) to benefit from\
\ using classifiers and sophisticated prioritization to automatically triage static\
\ analysis alerts."
version: "1.0.1"
title: "SCAIFE DataHub Module API Definition"
contact:
name: "SCAIFE Project"
url: "https://www.sei.cmu.edu/research-capabilities/all-work/display.cfm?customel_datapageid_4050=6453"
email: "lflynn@cert.org"
license:
name: " SCAIFE API Copyright 2007-2020 Carnegie Mellon University. All Rights\
\ Reserved. Redistribution and use in source and binary forms, with or without\
\ modification, are permitted provided that the following conditions are met:\
\ 1. Redistributions of source code must retain the above copyright notice,\
\ this list of conditions and the following disclaimer. 2. Redistributions\
\ in binary form must reproduce the above copyright notice, this list of conditions\
\ and the following disclaimer in the documentation and/or other materials provided\
\ with the distribution. 3. Products derived from this software may not include\
\ 'Carnegie Mellon University,' 'SEI' and/or 'Software Engineering Institute'\
\ in the name of such derived product, nor shall 'Carnegie Mellon University,'\
\ 'SEI' and/or 'Software Engineering Institute' be used to endorse or promote\
\ products derived from this software without prior written permission. For\
\ written permission, please contact permission@sei.cmu.edu. ACKNOWLEDGMENTS\
\ AND DISCLAIMERS: This material is based upon work funded and supported by\
\ the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie\
\ Mellon University for the operation of the Software Engineering Institute,\
\ a federally funded research and development center. The view, opinions,\
\ and/or findings contained in this material are those of the author(s) and\
\ should not be construed as an official Government position, policy, or decision,\
\ unless designated by other documentation. NO WARRANTY. THIS CARNEGIE MELLON\
\ UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN\
\ 'AS-IS' BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND,\
\ EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO,\
\ WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS\
\ OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE\
\ ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR\
\ COPYRIGHT INFRINGEMENT. [DISTRIBUTION STATEMENT A] This material has been\
\ approved for public release and unlimited distribution. Please see Copyright\
\ notice for non-US Government use and distribution. This material includes\
\ field names used in the Software Assurance Marketplace (SWAMP), a service\
\ that provides continuous software assurance capabilities to developers and\
\ researchers at https://www.mir-swamp.org/#. Copyright © 2012-2020 The Morgridge\
\ Institute for Research, Inc. All rights reserved. This material includes\
\ field names used in the Software Assurance Tool (SwAT), a tool that is used\
\ by analysts to analyze static analysis alerts from multiple static analysis\
\ tools. https://www.cerdec.army.mil/ Combat Capabilities Development Command\
\ (CCDC) CSISR Center. All rights reserved. DM19-0572 "
host: "127.0.0.1:8084"
paths:
/status:
get:
tags:
- "DataHub_Server"
summary: "Provides Server Status"
operationId: "get_status"
produces:
- "application/json"
parameters:
- name: "x_request_token"
in: "header"
description: "Token that associates messages with each other"
required: true
type: "string"
responses:
"200":
description: "Server Is Running"
schema:
DataHu
GET
GET
GET
Parameter
Name
x_access
string
(header)
x_reques
string
(header)
Response
Curl
curl -X G
eyJ0eXAiO
YTc4OWU3N
-p4QykxMH
Request UR
http://127
Server resp
Code
200

Figure 7: Successful Execution of Getting Packages from DataHub, via Swagger-Editor GUI

Next:

- Get the tool id
    - Navigate to Datahub_Server `GET /tools`
    - Click `Try it out`
    - Enter `x_access_token`, and any value for `x_request_token` (e.g., for the latter, "`abc`")
    - Click `Execute` and retrieve the `tool_id`

Results of successful execution are shown in the screenshot below, which in this case shows a `tool_id` of `5f040412823a3994fe4a066d`

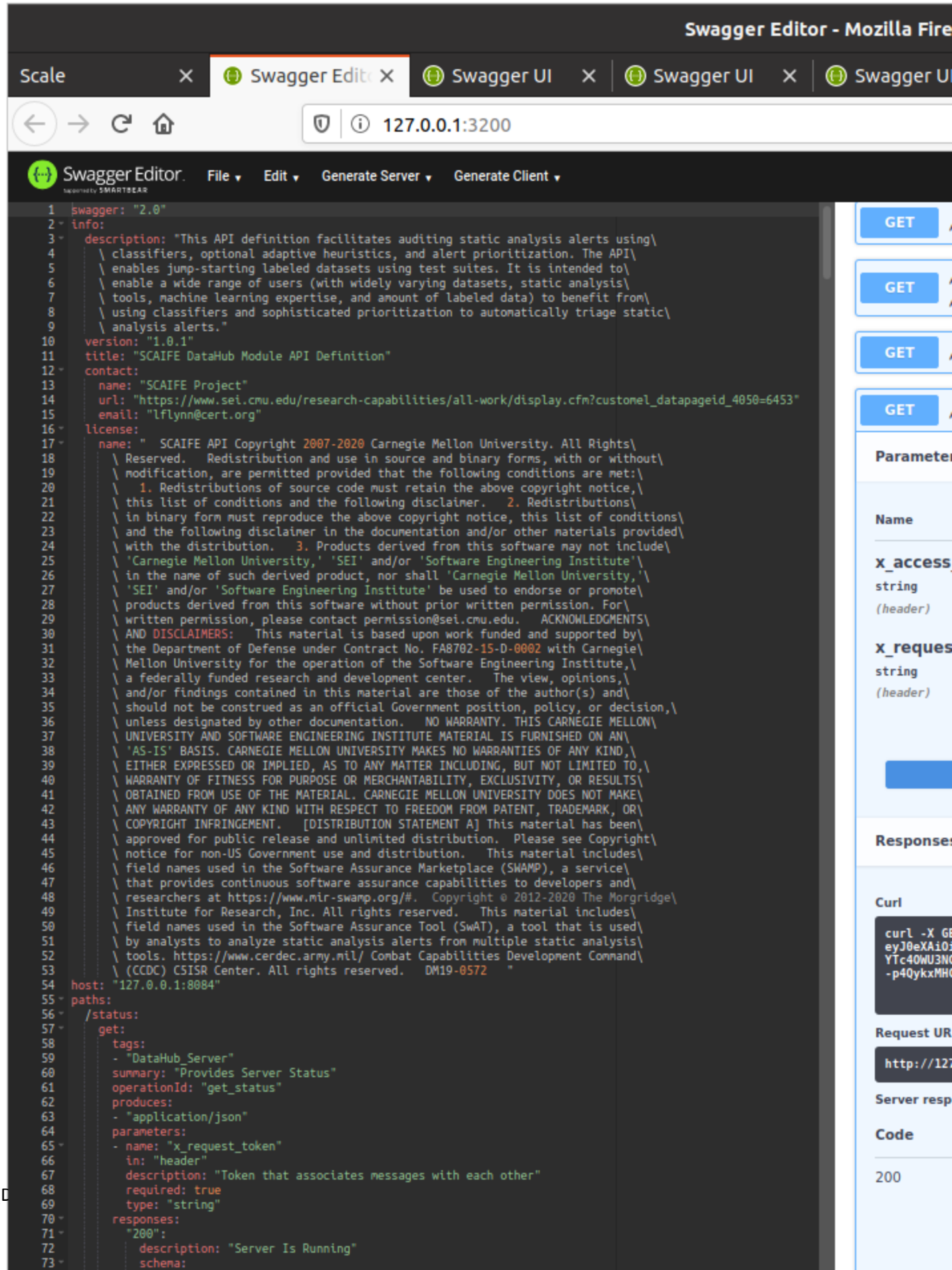

Swagger Editor - Mozilla Fire
Scale
Swagger Edit
Swagger UI
Swagger UI
Swagger UI
127.0.0.1:3200
Swagger Editor
File
Edit
Generate Server
Generate Client
swagger: "2.0"
info:
description: "This API definition facilitates auditing static analysis alerts using\
\ classifiers, optional adaptive heuristics, and alert prioritization. The API\
\ enables jump-starting labeled datasets using test suites. It is intended to\
\ enable a wide range of users (with widely varying datasets, static analysis\
\ tools, machine learning expertise, and amount of labeled data) to benefit from\
\ using classifiers and sophisticated prioritization to automatically triage static\
\ analysis alerts."
version: "1.0.1"
title: "SCAIFE DataHub Module API Definition"
contact:
name: "SCAIFE Project"
url: "https://www.sei.carnegie.edu/research-capabilities/all-work/display.cfm?customel_datapageid_4050=6453"
email: "lflynn@cert.org"
license:
name: " SCAIFE API Copyright 2007-2020 Carnegie Mellon University. All Rights\
\ Reserved. Redistribution and use in source and binary forms, with or without\
\ modification, are permitted provided that the following conditions are met:\
\ 1. Redistributions of source code must retain the above copyright notice,\
\ this list of conditions and the following disclaimer. 2. Redistributions\
\ in binary form must reproduce the above copyright notice, this list of conditions\
\ and the following disclaimer in the documentation and/or other materials provided\
\ with the distribution. 3. Products derived from this software may not include\
\ 'Carnegie Mellon University,' 'SEI' and/or 'Software Engineering Institute'\
\ in the name of such derived product, nor shall 'Carnegie Mellon University,'\
\ 'SEI' and/or 'Software Engineering Institute' be used to endorse or promote\
\ products derived from this software without prior written permission. For\
\ written permission, please contact permission@sei.cmu.edu. ACKNOWLEDGMENTS\
\ AND DISCLAIMERS: This material is based upon work funded and supported by\
\ the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie\
\ Mellon University for the operation of the Software Engineering Institute,\
\ a federally funded research and development center. The view, opinions,\
\ and/or findings contained in this material are those of the author(s) and\
\ should not be construed as an official Government position, policy, or decision,\
\ unless designated by other documentation. NO WARRANTY. THIS CARNEGIE MELLON\
\ UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN\
\ 'AS-IS' BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND,\
\ EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO,\
\ WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS\
\ OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE\
\ ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR\
\ COPYRIGHT INFRINGEMENT. [DISTRIBUTION STATEMENT A] This material has been\
\ approved for public release and unlimited distribution. Please see Copyright\
\ notice for non-US Government use and distribution. This material includes\
\ field names used in the Software Assurance Marketplace (SWAMP), a service\
\ that provides continuous software assurance capabilities to developers and\
\ researchers at https://www.mir-swamp.org/#. Copyright © 2012-2020 The Morgridge\
\ Institute for Research, Inc. All rights reserved. This material includes\
\ field names used in the Software Assurance Tool (SwAT), a tool that is used\
\ by analysts to analyze static analysis alerts from multiple static analysis\
\ tools. https://www.cerdec.army.mil/ Combat Capabilities Development Command\
\ (CCDC) CSISR Center. All rights reserved. DM19-0572 "
host: "127.0.0.1:8084"
paths:
/status:
get:
tags:
- "DataHub_Server"
summary: "Provides Server Status"
operationId: "get_status"
produces:
- "application/json"
parameters:
- name: "x_request_token"
in: "header"
description: "Token that associates messages with each other"
required: true
type: "string"
responses:
"200":
description: "Server Is Running"
schema:
GET
GET
GET
GET
Parameter
Name
x_access
string
(header)
x_reques
string
(header)
Response
Curl
curl -X G
eyJ0eXAiO
YTc4OWU3N
-p4QykxMH
Request UR
http://12
Server resp
Code
200

Figure 8: : Successful Execution of Getting Tools, from DataHub, via Swagger-Editor GUI

Next, you will upload a cppcheck tool output file to the DataHub.
Please use the file `model_api/ui_server_stub/scale.app/demo/micro_juliet_v1.2_cppcheck_b/micro_juliet_cppcheck_tool_output.xml` that is included in the code provided to you.

- Navigate to Datahub_Server `POST /tools/{tool_id}/packages/{package_id}`
- Click `Try it out`
- Enter `x_access_token`, and any value for `x_request_token` (e.g., for the latter, "`abc`")
- Enter `tool_id` and `package_id`
- Select the "`Browse`" button for `raw_tool_output file` , then browse to select the file `model_api/ui_server_stub/scale.app/demo/micro_juliet_v1.2_cppcheck_b/micro_juliet_cppcheck_tool_output.xml`
- Click `Execute` and retrieve the `tool_id`

The request to send tool output is (generically, with purple font to indicate where substitution should be made):

```
curl -X POST http://127.0.0.1:8084/tools/$tool_id/packages/$package_id
-H "accept: application/json" -H "x_access_token: $access-token" -H
"x_request_token: $random_string" -H "Content-Type: multipart/form-
data" -F raw_tool_output=@$CPPCHECK_XML_FILE.xml;type=text/xml
```

The actual Curl command used for a particular test is shown in the screenshot below. It is:

```
curl -X POST
"http://127.0.0.1:8084/tools/5f040412823a3994fe4a066d/packages/5f04041
a823a3994fe4a08f7" -H  "accept: application/json" -H  "x_access_token:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6InVzZXJuYW1lIiwib
3JnYW5pemF0aW9uX2lkIjoiNWYwNDE0YTc4OWU3NGUxYWFlYWQ1YzI2Iiwic2VydmVyX2t
leSI6ImRhdGFodWJrZXkiLCJleHAiOjE1OTQxOTE3NDh9.jo2YfR0wzyznjeqW-
p4QykxMHCg6-Rt6njhZD5ip8bM" -H  "x_request_token: abc" -H  "Content-
Type: multipart/form-data" -F
"raw_tool_output=@micro_juliet_cppcheck_tool_output.xml;type=text/xml"
```

The DataHub returns the following, as can also be seen in the screenshot below:

```
{
  "task_id": "e12d2d3d-8dc8-4d03-8962-07e3f3c4b68b",
  "task_url": "/status/tool-output/e12d2d3d-8dc8-4d03-8962-
07e3f3c4b68b"
}
```

See screenshot after successful tool output upload, below:

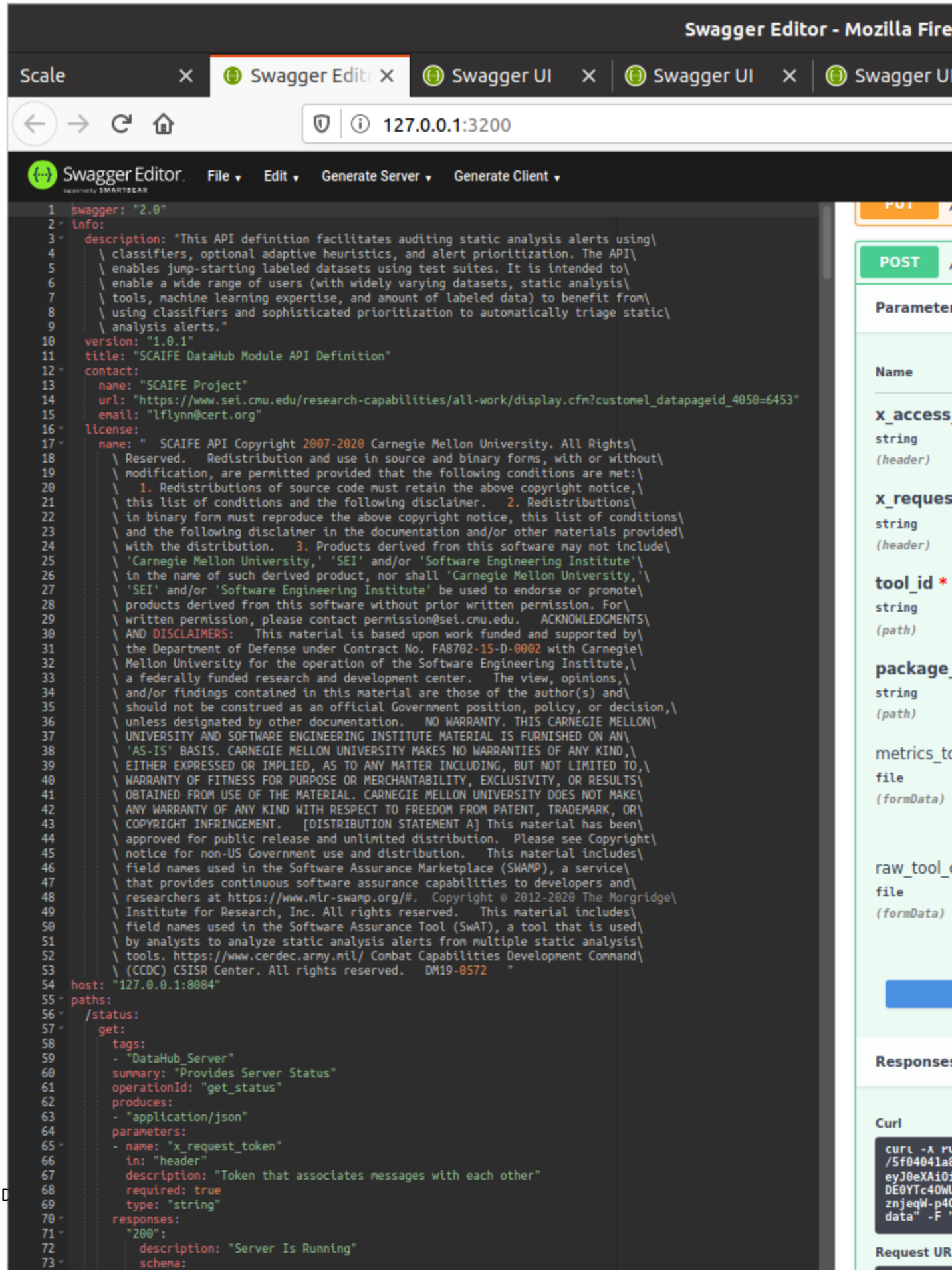

Swagger Editor - Mozilla Fire
Scale
Swagger Edit
Swagger UI
Swagger UI
Swagger UI
127.0.0.1:3200
Swagger Editor
File
Edit
Generate Server
Generate Client
swagger: "2.0"
info:
description: "This API definition facilitates auditing static analysis alerts using\
\ classifiers, optional adaptive heuristics, and alert prioritization. The API\
\ enables jump-starting labeled datasets using test suites. It is intended to\
\ enable a wide range of users (with widely varying datasets, static analysis\
\ tools, machine learning expertise, and amount of labeled data) to benefit from\
\ using classifiers and sophisticated prioritization to automatically triage static\
\ analysis alerts."
version: "1.0.1"
title: "SCAIFE DataHub Module API Definition"
contact:
name: "SCAIFE Project"
url: "https://www.sei.cmu.edu/research-capabilities/all-work/display.cfm?customel_datapageid_4050=6453"
email: "lflynn@cert.org"
license:
name: " SCAIFE API Copyright 2007-2020 Carnegie Mellon University. All Rights\
\ Reserved.   Redistribution and use in source and binary forms, with or without\
\ modification, are permitted provided that the following conditions are met:\
\   1. Redistributions of source code must retain the above copyright notice,\
\ this list of conditions and the following disclaimer.   2. Redistributions\
\ in binary form must reproduce the above copyright notice, this list of conditions\
\ and the following disclaimer in the documentation and/or other materials provided\
\ with the distribution.   3. Products derived from this software may not include\
\ 'Carnegie Mellon University,' 'SEI' and/or 'Software Engineering Institute'\
\ in the name of such derived product, nor shall 'Carnegie Mellon University,'\
\ 'SEI' and/or 'Software Engineering Institute' be used to endorse or promote\
\ products derived from this software without prior written permission. For\
\ written permission, please contact permission@sei.cmu.edu.   ACKNOWLEDGMENTS\
\ AND DISCLAIMERS: This material is based upon work funded and supported by\
\ the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie\
\ Mellon University for the operation of the Software Engineering Institute,\
\ a federally funded research and development center.   The view, opinions,\
\ and/or findings contained in this material are those of the author(s) and\
\ should not be construed as an official Government position, policy, or decision,\
\ unless designated by other documentation.   NO WARRANTY. THIS CARNEGIE MELLON\
\ UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN\
\ 'AS-IS' BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND,\
\ EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO,\
\ WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS\
\ OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE\
\ ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR\
\ COPYRIGHT INFRINGEMENT.   [DISTRIBUTION STATEMENT A] This material has been\
\ approved for public release and unlimited distribution. Please see Copyright\
\ notice for non-US Government use and distribution.   This material includes\
\ field names used in the Software Assurance Marketplace (SWAMP), a service\
\ that provides continuous software assurance capabilities to developers and\
\ researchers at https://www.mir-swamp.org/#. Copyright © 2012-2020 The Morgridge\
\ Institute for Research, Inc. All rights reserved.   This material includes\
\ field names used in the Software Assurance Tool (SwAT), a tool that is used\
\ by analysts to analyze static analysis alerts from multiple static analysis\
\ tools. https://www.cerdec.army.mil/ Combat Capabilities Development Command\
\ (CCDC) CSISR Center. All rights reserved.   DM19-0572   "
host: "127.0.0.1:8084"
paths:
/status:
get:
tags:
- "DataHub_Server"
summary: "Provides Server Status"
operationId: "get_status"
produces:
- "application/json"
parameters:
- name: "x_request_token"
in: "header"
description: "Token that associates messages with each other"
required: true
type: "string"
responses:
"200":
description: "Server Is Running"
schema:
POST
Parameter
Name
x_access
string
(header)
x_reques
string
(header)
tool_id *
string
(path)
package_
string
(path)
metrics_to
file
(formData)
raw_tool_
file
(formData)
Response
Curl
curl -X
/5f0404la
eyJ0eXAiO
DE0YTc4OW
znjeqW-p4
data" -F
Request UR

Figure 9: Successful Tool Output Upload to DataHub, via Swagger-Editor GUI

Using Auto-Generated Client Code to Implement SCAIFE API Calls from Your Tool

The Swagger-Codegen tool automatically generates code from an Open API formatted YAML or JSON file. Swagger-Codegen can be used on each of the five SCAIFE swagger.yaml files to generate client code in a wide variety of languages (including C, Python, Java, and many more). We have auto-generated Java client code for the latest version of each of the SCAIFE API files.

This client code makes the API call using the language generated, with the correct parameters and return values set. However, a developer must do additional work to make the code functional. The developer must set the API call parameters using data values that might initially be hardcoded with test values, and later would be substituted for values from dataflows within their own tool. Automatically inserting reasonable values for the parameters may require coding new logic within your tool. Similarly, the return value(s) from the API call will need to be dealt with by your tool, which will require some development of your tool's code.

We suggest that the first API call you automate be either 1) registering with the SCAIFE registration server; or 2) making the API call to upload raw tool output. Both of these (and other options you may wish to consider) are described above in the Swagger-Editor example. For implementing the API call from your tool, select where in your tool's code you will place the generated client code for that API call. Next, modify as needed to hardcode test parameters or to use your tool's dataflow to select parameter values for your API call. Then, start up the SCAIFE containers as before (same as described at the top of this document, possibly using detail in the last section). Next, test, debug, and complete implementation of that SCAIFE API call from your tool.

We welcome feedback as you work through this process. We hope to be able to cite your tool integration (although you can be anonymous, if you prefer) as a successful example of SCAIFE integration with a variety of tools.

Detail for auto-generating code from the SCAIFE APIs yourself

Information about Swagger-codegen (free, downloadable software from GitHub) is here:
https://github.com/swagger-api/swagger-codegen
NOTE: The SCAIFE swagger APIs use version 2, so the ***master*** branch of swagger-codegen should be used (not the 3.0.0 branch).

The command below was used for Java code generation for the SCAIFE registration module (Note that others should modify the command below, since the `swagger.yaml` file location is specific to the directory structure of your own machine.):

```
java -jar modules/swagger-codegen-cli/target/swagger-codegen-
cli.jar generate \
   -i
/home/lflynn/scale/epp/model_api/registration_server_stub/swagger
_server/swagger/swagger.yaml \
   -l java \
   -o /home/lflynn/temp/swagger-java-client
```

A similar command was used to generate Java code for the other 4 SCAIFE modules, using their own `swagger.yaml` filepaths.

Similar commands can be used to auto-generate code in many different languages. Note that "-l java" is the key part of the command that determines the generated code language.

Useful information and commands for SCAIFE containers

How to stop, restart, and check if containers are running during testing

NOTE: Commands provided here are slightly different than in the HTML manual, since those commands are for a SCAIFE VM distribution with dependent containers (containers that share files with the host machine), but this document describes using independent containers (that do not share files with the host machine) with a code-only distribution. (E.g., this document's instructions are different in that way from instructions at model_api/ui_server_stub/scale.app/public/doc/scale2/SCAIFE-Server-Management.html)

SERVER	PORT	Mongo SERVER associated with column 1's swagger server	Mongo Server's PORT	Redis Server's PORT
scale	8083			
datahub	8084	mongodb_datahub	28084	28184
priority	8085	mongodb_priority	28085	
stats	8086	mongodb_stats	28086	
registration	8087	mongodb_registration	28087	

Figure 10: Reference Table with Port IDs per Server (Container)

Command to test that swagger server is running (substitute $PORT for number in table above)

Note that the following command requires that the `wget` package be installed. (On a machine that uses `apt`, the following command would install it: `sudo apt-get install wget`)

```
wget -q -O - localhost:$PORT/status --header="x_request_token: deadbeef"
```

Example command and response:

```
model_api$ wget -q -O - localhost:8084/status --
header="x_request_token: deadbeef"
{
  "message": "DataHub Server is Running Properly",
  "request_id": "deadbeef"
}
```

Commands to refresh all SCAIFE containers

```
docker-compose down
docker-compose -f docker-compose.yml up --build
```

Commands to refresh a single SCAIFE independent container (SCALe or swagger)
CAUTION: in many cases, this will make the other SCAIFE containers have inconsistent state. However, this is sometimes useful when solely testing one container (e.g., when only testing the SCALe container, without uploading any projects to the datahub container)

```
docker-compose stop ${SERVER}
docker-compose rm ${SERVER}
docker-compose -f docker-compose.yml up ${SERVER} --build
```

Example commands:

```
docker-compose stop datahub
docker-compose rm datahub
docker-compose -f docker-compose.yml up datahub
```

Commands to pause and resume all SCAIFE independent containers

To pause SCAIFE, without resetting any data

```
docker-compose stop
```

To resume SCAIFE from a pause

```
docker-compose start
```

How to Run Scripts, Tests, and Other Commands in SCAIFE Containers

Using docker-compose to test script-based creation of SCALe projects and use of them in an independent container

NOTE: the first two commands below require all the SCAIFE containers to be active (up)

```
docker exec scale python scripts/automation/create_manual_test_project.1.microju
docker exec scale python scripts/automation/create_manual_test_project.1.roseche
docker exec scale python scripts/automation/create_basic_project.py
```

Command to get a bash command line for a container (e.g., and then inspect the container's internal filesystem)

Access a bash command line on the container `$SERVER` (e.g., datahub) you specify

```
docker exec -it $SERVER /bin/bash
```

Example commands:

```
docker exec -it datahub /bin/bash
root@622cbc872c78:/usr/src/app# ls
     ABOUT       api_html_formatter.py           init.sh
     requirements.txt  swagger_server.egg-info
     Dockerfile  api_json_and_html_generator.py  insert_project.py
     setup.py          test-requirements.txt
     README.md   git_push.sh                     nosetests.xml
     swagger_server    tox.ini
```

How to read log files in SCAIFE containers

You can access the log of a swagger server's associated Mongo container.

Command to obtain a SCAIFE Mongodb log

```
docker logs mongodb_${SERVER}
```

Here are some useful log files:

Container	Path	Purpose
SCALe	`/scale/log/development.log`	Rails log

If `$CONTAINER` is your container's name (specified above via the `--name` flag, and a value you see listed if you run the command `docker container ls`), and `$LOG` is the path of a log file in a container, you can copy the log file from container to host with this command:

Command to access SCAIFE log files

```
docker cp $CONTAINER:$LOG .
```

You could also watch new output to the log file, without copying it, with this command:

Command to monitor log files in containers

```
docker exec -it $CONTAINER tail -f $LOG
```

Commands to connect to SCALe in an independent container, access one of its sqlite3 databases, then execute sqlite3 commands

```
# if SCALe is not running, start it
cd model_api
docker-compose -f docker-compose.yml up scale
# Access SCALe
docker exec -it scale /bin/bash
# Inside container
sqlite3 db/development.sqlite
# Next, commands within the sqlite3 command environment (commandline starts with
# List all tables
.tables
# Show everything from the 'projects' table
SELECT * FROM projects;
.quit
# This line outputs a dump of the SCALe internal database
echo .dump | sqlite3 db/development.sqlite
```