Carnegie Mellon University
Software Engineering Institute

SCAIFE API DEFINITION BETA VERSION 0.0.2 FOR DEVELOPERS

Lori Flynn
Ebonie McNeil
June 2019

Introduction

This paper provides the Source Code Analysis Integrated Framework Environment (SCAIFE) API definition for beta version 0.0.2. SCAIFE is an architecture that supports static analysis alert classification and prioritization. It is designed so a wide variety of static analysis tools can integrate with the system using the API definition we are developing. We expect this paper to be of interest to organizations that develop and/or research static analysis alert auditing tools, aggregators, and other frameworks. Developers may refer to this SCAIFE beta API definition to help them to estimate development effort that would be required to modify their organization's tool(s) to make and respond to SCAIFE API calls. Also, this beta API definition is being published to generate feedback from developers and organizations interested in implementing the SCAIFE API, and to help improve SCAIFE API v1.0.0 to become more usable by developers of a wide variety of static analysis tools. Compared to the beta API definitions, the published SCAIFE API v1.0.0 definition will include implementation details, the architecture description, motivations, and a prototype system.

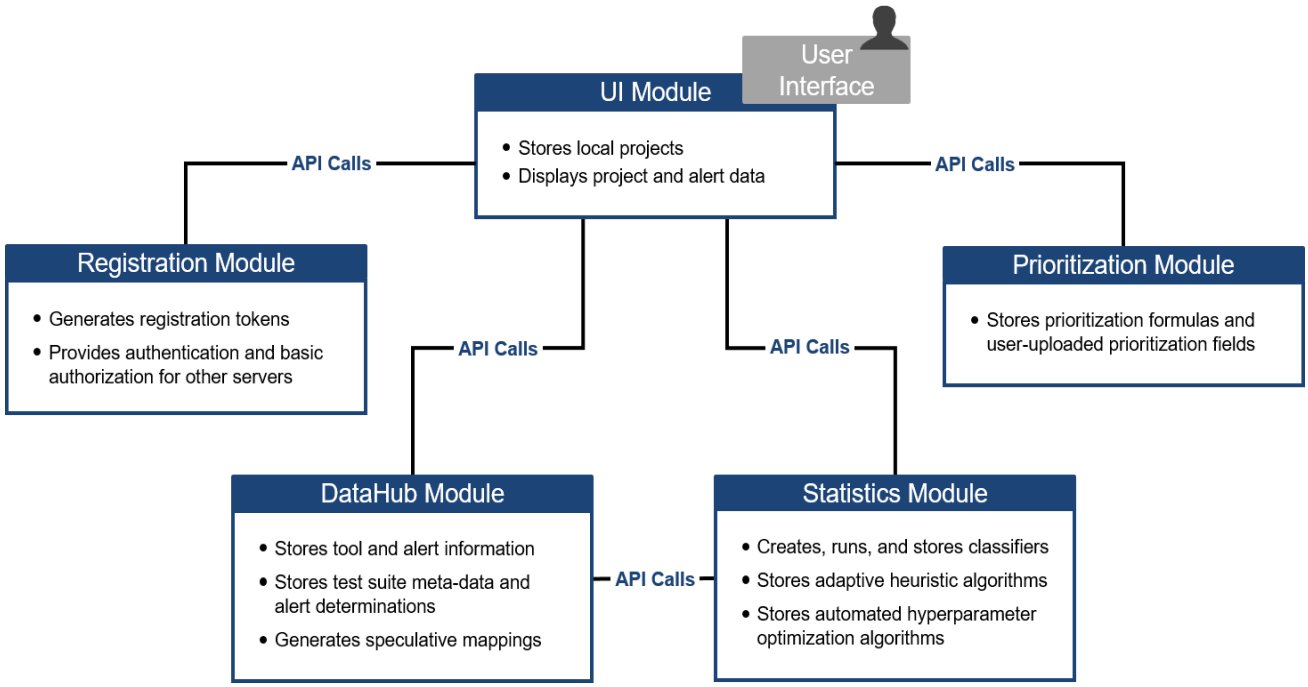

Figure 1: SCAIFE Architecture. The modular structure allows system components to be distributed, or used as combined on a single machine.

A previous version of the SCAIFE API beta definition (version 0.0.1, from September 2018) was published in an appendix of a Software Engineering Institute (SEI) technical report: *Integration of Automated Static Analysis Alert Classification and Prioritization with Auditing Tools: Special Focus on SCALe* [3]. Significant modifications have been made to the API since then based, in part, on feedback received by organizations that are interested in integrating their tools with SCAIFE via API calls. Modifications between SCAIFE API versions 0.0.1 and 0.0.2 include adding a registration server, and adding and modifying many API calls and their associated data models. After examining APIs for SWAMP [11] and SwAT [13], we also added several fields to SCAIFE API v0.0.2 to enable easier future integration of those tools with SCAIFE.

While completing SCAIFE API version 1.0.0, the SCAIFE development team is simultaneously completing a prototype instantiation of the architecture, a multi-server software system whose servers communicate using SCAIFE API calls. The SCAIFE prototype is intended to be used by engineers to audit alerts from multiple static analysis tools via a GUI front end. The back-end system stores audit archive data in the databases, and supports automated alert classification (e.g., true, false, indeterminate, etc.) and advanced alert prioritization based on mathematical user-defined formulas.

The SCAIFE prototype includes the latest version of SCALe [8], the SEI-developed alert auditing framework that provides a GUI front end for examining code and marking determinations (e.g., true or false), and a back end that stores audit data in a database archive. SCALe has been modified to include features for advanced alert prioritization, using mathematical formulas, and for integrating with SCAIFE [5, 6, 7] for automated alert classification and other SCAIFE functionality. The latest version of SCALe includes modifications to enable different modes of operation: SCAIFE-connected, SCALe-only, and Demo modes. The SCAIFE prototype can either be used as-is, or particular servers can be swapped out or modified by developers. The prototype will initially be distributed to research project collaborators who will test it and provide feedback. Readers of this paper who are interested in testing the SCAIFE prototype are invited to contact the authors. (See page 105 for SEI contact information.)

The planned SCAIFE system will provide an architecture with an API and an open-source prototype system that has the following benefits to users:

- They can quickly start to use automated classifiers for static analysis alerts. The system *will not* require
  - a labeled audit archive to be provided in advance, since it uses test suites in a new way [4]
  - a machine learning expert
  - users to create their own frameworks for using classifiers
- They can quickly apply formulas that prioritize static analysis alerts by using factors they care about. These prioritization formulas can combine various fields, including classifier-derived confidence, with mathematical operators.
- They can employ the API to build upon the original prototype system, enabling the use of additional flaw-finding static analysis tools, code metrics tools [1,15], adaptive heuristics [9], classification techniques, and so forth.

The SCAIFE architecture shown in Figure 1 includes five servers; however, the API definition below has only four sections, which describe API method calls for 4 of the servers, but not the UI Module. (This is because the other servers do not make API calls to the UI Module. Calls from the UI Module to the other servers are listed in each of the four sections.) The UI Module represents existing analysis tools that display alert data in a GUI front end—including tool aggregators like SCALe, SWAMP [11], and the Army Combat Capabilities Development Command (CCDC) C5ISR Center's Software Assurance Tool (SwAT) [13]. The UI Module must instantiate API calls to the other four servers. Each API definition section below is furtherU categorized based on the source and destination modules of the API calls. For instance, the Rapid Models Registration and Login Module API Definition section contains only one category of API calls under the label *UIToRegistration*. The source (request) of the API calls comes from the UI Module, and the API calls are forwarded to the destination—the Registration Module. Each server follows this convention with the exception of the DataHub Module. The DataHub Module contains many API calls with multiple source modules (e.g., UI and Stats); to avoid duplication, the label `DataHub-Server` is used for these API calls. All of the resources, or data models, used in the architecture are alphabetized and located at the end of the API definition methods, within the `Models` section, for better readability. The models and methods can be accessed by following the hyperlinks associated with each resource in the SCAIFE API Definition section below.

The following API definition was developed using the Swagger/OpenAPI open-source software development toolset [9, 12]. We chose this toolset because it is in wide use (approximately 10,000 downloads daily) and provides automated code generation from API specifications and automated testing. These features not only support SEI development of the SCAIFE API and the prototype instantiation of the SCAIFE architecture, but also other developers' work to generate implementation code for the SCAIFE API within their own tools.

API Definition YAML File

SEI has published a YAML [14] formatted file specifying the SCAIFE API, available at the CMU-SEI GitHub site "SCAIFE API" [2] for free downloads by the public. The YAML specification provides the SCAIFE API definition beta version 0.0.2, in a format that developers can easily use to view, modify, and automatically generate code from (e.g., with the Swagger Editor and Swagger Codegen tools [12]). The YAML file was almost entirely manually created by SEI developers. The only things that were auto-generated by Swagger tools [12] within the YAML file are some of the examples.

The API Definition Below and How to Use It

The SCAIFE API definition is provided below, in text originally generated by SEI developers in YAML. We used the Swagger Codegen tool [12] to produce an HTML version of the API documentation copied below, and then slightly modified the original output format to improve readability. The version included in this paper is more accessible to readers with diverse job titles and technical capabilities, since it does not require familiarity with YAML format, nor the installation of additional software (e.g., Swagger Editor) to facilitate viewing.

You can access the interface methods in two ways. If you are interested in a particular module, click on the hyperlink for that module's API Definition to be taken to the API calls for that module. You can also find an API call directly by using the links in the `Summary of API Methods` section. For the `PUT /projects/{project_id}/{package_id}/alerts` method in the `DataHubToStats` section, start by clicking on the `Rapid Models Statistics Module API Definition` link, or by clicking on the `PUT /projects/{project_id}/{package_id}/alerts` link under the list of Statistics methods. For this example, both routes take you to the API call definition. The `PUT` request (the `/projects/{project_id}/{package_id}/alerts` API call) in the `DataHubToStats` section is used to forward new alerts from the DataHub Module to the Statistics Module. As you can see, this method expects two parameters in the URL path, denoted by the curly brackets around the `project_id` and `package_id` variables, and specified under the `Path parameters` subheading. All API calls for this architecture accept and return JSON objects, which are defined under the `Consumes` and `Produces` keywords.

The request body of this particular API call expects a `multiple_alerts` object. To identify the format for `multiple_alerts`, click on the hyperlink to be redirected to the model definition. Here you will see that the `multiple_alerts` object can contain an array of `meta_alert` objects and/or an array of `alert` objects. Click on the `meta_alert` link to be redirected to the `meta_alert` object's definition, as follows:

meta_alert -

meta_alert_id
String

alert_ids (optional)
array[String]

filepath (optional)
String

line_start (optional)
Integer

condition_id
String

determinations (optional)
determination

verdict (optional)
map[String, array[String]]

A `meta_alert` object also contains additional embedded objects, `determinations`, which can be similarly accessed. To return the top level of a section, use the `Up` hyperlink. From the `meta_alert` object, clicking `Up` will take you to the beginning of the `Summary of API Models` section. From here, to return to the list of API calls, click on the `Jump to Methods` hyperlink. Here, you can explore the path for another API call or take a similar route to find other object formats.

The specification for the formats and ranges of object values is not defined in the beta API definition version 0.0.2. We plan to define this information in the API prior to the release of SCAIFE version 1.0.0.

SCAIFE API (Beta Version 0.0.2)

SCAIFE API Definition

This API facilitates auditing static analysis alerts using classifiers, alert prioritization, and optional adaptive heuristics. It also supports jump-starting labeled datasets using test suites. The API is intended to enable diverse 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.
More information: https://www.sei.cmu.edu/research-capabilities/all-work/display.cfm?customel_datapageid_4050=6453

Contact Info: lflynn@cert.org
Version: 0.0.2
BasePath: 127.0.0.1

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.

API Index of SCAIFE Servers

Summary of API Methods

[Jump to Models]

DataHub Methods

Up

DataHubServer

- `GET /projects/{project_id}/alerts`
- `GET /packages/{package_id}`
- `POST /projects/retrieve`
- `GET /taxonomies`
- `GET /taxonomies/{taxonomy_id}`

- GET /test_suites
- GET /tools/{tool_id}
- GET /tools
- GET /packages
- GET /projects

StatsToDataHub

- POST /projects/adaptive_heuristics/close

UIToDataHub

- POST /packages
- POST /projects
- DELETE /packages/{package_id}
- DELETE /projects/{project_id}
- PUT /packages/{package_id}
- PUT /projects/{project_id}
- POST /packages/{package_id}/alerts
- POST /projects/{project_id}/alerts
- POST /test_suites
- POST /tools

Prioritization Methods

Up

UIToPrioritization

- POST /priorities
- DELETE /priorities/{priority_scheme_id}/projects/{project_id}
- GET /priorities/{priority_scheme_id}/projects/{project_id}
- GET /priorities
- PUT /priorities/{priority_scheme_id}

Registration Methods

Up

UIToRegistration

- GET /server/{server_name}
- POST /login
- POST /register

Statistics Methods

Up

DataHubToStats

- PUT /projects/{project_id}/{package_id}/alerts
- PUT /packages/tools/{tool_id}

UIToStats

- PUT /classifiers/{classifier_instance_id}/adaptive_heuristics/close
- POST /classifiers
- DELETE /classifiers/{classifier_instance_id}
- PUT /classifiers/{classifier_instance_id}
- GET /classifiers/{classifier_instance_id}
- GET /classifiers
- PUT /classifiers/{classifier_instance_id}/retrain
- PUT /classifiers/{classifier_instance_id}/projects/{project_id}

Rapid Models DataHub Module API Definition

DataHubServer

Up

`GET /projects/{project_id}/alerts`

Retrieve all of the alerts and meta-alerts for a specific project. Request only the alerts and meta-alerts for a project in the DataHub. (**getAlertsForProject**)

Path parameters

project_id (required)

Path Parameter — The id of the project

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request headers

Return type

get_alerts_response

Example data

Content-Type: application/json


```
{
  "meta_alerts": [
    {
      "meta_alert_id": "string",
      "alert_ids": [
        "string"
      ],
      "filepath": "string",
      "line_start": 0,
      "condition_id": "string",
      "determination": {
        "flag_list": [
          {
            "flag": true,
            "timestamp": "2019-05-20T14:30:02.658Z"
          }
        ],
        "verdict_list": [
          {
            "verdict": "string",
            "timestamp": "2019-05-20T14:30:02.658Z"
          }
        ],
        "ignored_list": [
          {
            "ignored": true,
            "timestamp": "2019-05-20T14:30:02.658Z"
          }
        ],
```

```
        "dead_list": [
          {
            "dead": true,
            "timestamp": "2019-05-20T14:30:02.658Z"
          }
        ],
        "inapplicable_environment_list": [
          {
            "inapplicable_environment": true,
            "timestamp": "2019-05-20T14:30:02.658Z"
          }
        ],
        "dangerous_construct_list": [
          {
            "dangerous_construct": "string",
            "timestamp": "2019-05-20T14:30:02.658Z"
          }
        ],
        "notes_list": [
          {
            "notes": "string",
            "timestamp": "2019-05-20T14:30:02.658Z"
          }
        ]
      },
      "verdict": {
        "additionalProp1": [
          "string"
        ],
        "additionalProp2": [
          "string"
        ],
        "additionalProp3": [
          "string"
        ]
      }
    }
  ],
  "alerts": [
    {
      "alert_id": "string",
      "tool_id": "string",
      "checker_id": "string",
      "primary_message": {
        "line_start": 0,
        "line_end": 0,
        "filepath": "string"
```

```
      },
      "more_messages": [
        {
          "line_start": 0,
          "line_end": 0,
          "filepath": "string",
          "message_text": "string"
        }
      ]
    }
  ],
  "request_id": "string"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

Return the Alerts Associated with the Project get_alerts_response

400

Invalid Request

404

Alerts Not Found

default

Unexpected Error error

Up

`GET /packages/{package_id}`

Get an existing package from the DataHub Module. Send the package_id to the Data-Hub to retrieve the package information stored. This function is used to send information to the DataHub from the UI and Stats modules. (**getPackage**)

Path parameters

package_id (required)

Path Parameter — The id of the package to retrieve

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request headers

Return type

get_package_response

Example data

Content-Type: application/json


```
{
  "package" : {
    "alerts" : [ {
      "alert_id" : "alert_id",
      "primary_message" : {
        "filepath" : "filepath",
        "line_start" : 6,
        "line_end" : 1
      },
      "tool_id" : "tool_id",
      "checker_id" : "checker_id",
      "more_messages" : [ "", "" ]
    }, {
      "alert_id" : "alert_id",
      "primary_message" : {
        "filepath" : "filepath",
        "line_start" : 6,
        "line_end" : 1
      },
      "tool_id" : "tool_id",
      "checker_id" : "checker_id",
      "more_messages" : [ "", "" ]
    } ],
    "test_suite_id" : "test_suite_id",
    "updated_at" : "2000-01-23T04:56:07.000+00:00",
```

```
"package_name" : "package_name",
"created_at" : "2000-01-23T04:56:07.000+00:00",
"package_id" : "package_id",
"tools" : [ {
  "tool_name" : "tool_name",
  "tool_version" : "tool_version",
  "checker_data" : [ {
    "checker_name" : "checker_name",
    "checker_id" : "checker_id",
    "conditions" : [ {
      "condition_name" : "condition_name",
      "title" : "title",
      "condition_id" : "condition_id",
      "platform" : "platform"
    }, {
      "condition_name" : "condition_name",
      "title" : "title",
      "condition_id" : "condition_id",
      "platform" : "platform"
    } ]
  }, {
    "checker_name" : "checker_name",
    "checker_id" : "checker_id",
    "conditions" : [ {
      "condition_name" : "condition_name",
      "title" : "title",
      "condition_id" : "condition_id",
      "platform" : "platform"
    }, {
      "condition_name" : "condition_name",
      "title" : "title",
      "condition_id" : "condition_id",
      "platform" : "platform"
    } ]
  } ],
  "code_metrics_data" : "{}",
  "tool_id" : "tool_id",
  "category" : "category"
}, {
  "tool_name" : "tool_name",
  "tool_version" : "tool_version",
  "checker_data" : [ {
    "checker_name" : "checker_name",
    "checker_id" : "checker_id",
    "conditions" : [ {
      "condition_name" : "condition_name",
      "title" : "title",
```

```
          "condition_id" : "condition_id",
          "platform" : "platform"
        }, {
          "condition_name" : "condition_name",
          "title" : "title",
          "condition_id" : "condition_id",
          "platform" : "platform"
        } ]
      }, {
        "checker_name" : "checker_name",
        "checker_id" : "checker_id",
        "conditions" : [ {
          "condition_name" : "condition_name",
          "title" : "title",
          "condition_id" : "condition_id",
          "platform" : "platform"
        }, {
          "condition_name" : "condition_name",
          "title" : "title",
          "condition_id" : "condition_id",
          "platform" : "platform"
        } ]
      } ],
      "code_metrics_data" : "{}",
      "tool_id" : "tool_id",
      "category" : "category"
    } ],
    "package_description" : "package_description"
  },
  "request_id" : "request_id"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

Return an Existing Package get_package_response

400

Invalid Request

404

Package Unavailable

default

Unexpected Error error

Up

`POST /projects/retrieve`

Get projects from the DataHub. Send tool and taxonomy information the stats module already has, along with projects it is requesting. This way, a response from the Data-Hub can efficiently leave out taxonomies and tools that the Stats Module already has info for. (**getProjects**)

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request body

projects_requested projects_requested (required)

Body Parameter — Tool and taxonomy information the requesting module already has, along with projects it is requesting.

Request headers

Return type

get_projects_response

Example data

Content-Type: application/json

```
{
  "projects" : [ {
    "project_description" : "project_description",
    "code_language" : "code_language",
    "package" : {
      "alerts" : [ {
        "alert_id" : "alert_id",
        "primary_message" : {
```

```
        "filepath" : "filepath",
        "line_start" : 6,
        "line_end" : 1
      },
      "tool_id" : "tool_id",
      "checker_id" : "checker_id",
      "more_messages" : [ "", "" ]
    }, {
      "alert_id" : "alert_id",
      "primary_message" : {
        "filepath" : "filepath",
        "line_start" : 6,
        "line_end" : 1
      },
      "tool_id" : "tool_id",
      "checker_id" : "checker_id",
      "more_messages" : [ "", "" ]
    } ],
    "test_suite_id" : "test_suite_id",
    "updated_at" : "2000-01-23T04:56:07.000+00:00",
    "package_name" : "package_name",
    "created_at" : "2000-01-23T04:56:07.000+00:00",
    "package_id" : "package_id",
    "tools" : [ {
      "tool_name" : "tool_name",
      "tool_version" : "tool_version",
      "checker_data" : [ {
        "checker_name" : "checker_name",
        "checker_id" : "checker_id",
        "conditions" : [ {
          "condition_name" : "condition_name",
          "title" : "title",
          "condition_id" : "condition_id",
          "platform" : "platform"
        }, {
          "condition_name" : "condition_name",
          "title" : "title",
          "condition_id" : "condition_id",
          "platform" : "platform"
        } ]
      }, {
        "checker_name" : "checker_name",
        "checker_id" : "checker_id",
        "conditions" : [ {
          "condition_name" : "condition_name",
          "title" : "title",
          "condition_id" : "condition_id",
```

```
            "platform" : "platform"
          }, {
            "condition_name" : "condition_name",
            "title" : "title",
            "condition_id" : "condition_id",
            "platform" : "platform"
          } ]
        } ],
        "code_metrics_data" : "{}",
        "tool_id" : "tool_id",
        "category" : "category"
      }, {
        "tool_name" : "tool_name",
        "tool_version" : "tool_version",
        "checker_data" : [ {
          "checker_name" : "checker_name",
          "checker_id" : "checker_id",
          "conditions" : [ {
            "condition_name" : "condition_name",
            "title" : "title",
            "condition_id" : "condition_id",
            "platform" : "platform"
          }, {
            "condition_name" : "condition_name",
            "title" : "title",
            "condition_id" : "condition_id",
            "platform" : "platform"
          } ]
        }, {
          "checker_name" : "checker_name",
          "checker_id" : "checker_id",
          "conditions" : [ {
            "condition_name" : "condition_name",
            "title" : "title",
            "condition_id" : "condition_id",
            "platform" : "platform"
          }, {
            "condition_name" : "condition_name",
            "title" : "title",
            "condition_id" : "condition_id",
            "platform" : "platform"
          } ]
        } ],
        "code_metrics_data" : "{}",
        "tool_id" : "tool_id",
        "category" : "category"
      } ],
```

```
      "package_description" : "package_description"
    },
    "existing_taxonomies" : [ "existing_taxonomies", "exist-
ing_taxonomies" ],
    "meta_alerts" : [ {
      "filepath" : "filepath",
      "verdict" : {
        "key" : [ "verdict", "verdict" ]
      },
      "determination" : {
        "flag_list" : [ {
          "flag" : true,
          "timestamp" : "2000-01-23T04:56:07.000+00:00"
        }, {
          "flag" : true,
          "timestamp" : "2000-01-23T04:56:07.000+00:00"
        } ],
        "inapplicable_environment_list" : [ {
          "inapplicable_environment" : true,
          "timestamp" : "2000-01-23T04:56:07.000+00:00"
        }, {
          "inapplicable_environment" : true,
          "timestamp" : "2000-01-23T04:56:07.000+00:00"
        } ],
        "ignored_list" : [ {
          "ignored" : true,
          "timestamp" : "2000-01-23T04:56:07.000+00:00"
        }, {
          "ignored" : true,
          "timestamp" : "2000-01-23T04:56:07.000+00:00"
        } ],
        "verdict_list" : [ {
          "verdict" : "verdict",
          "timestamp" : "2000-01-23T04:56:07.000+00:00"
        }, {
          "verdict" : "verdict",
          "timestamp" : "2000-01-23T04:56:07.000+00:00"
        } ],
        "dead_list" : [ {
          "dead" : true,
          "timestamp" : "2000-01-23T04:56:07.000+00:00"
        }, {
          "dead" : true,
          "timestamp" : "2000-01-23T04:56:07.000+00:00"
        } ],
        "dangerous_construct_list" : [ {
          "dangerous_construct" : "dangerous_construct",
```

```
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      }, {
        "dangerous_construct" : "dangerous_construct",
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      } ],
      "notes_list" : [ {
        "notes" : "notes",
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      }, {
        "notes" : "notes",
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      } ]
    },
    "meta_alert_id" : "meta_alert_id",
    "line_start" : 0,
    "condition_id" : "condition_id",
    "alert_ids" : [ "alert_ids", "alert_ids" ]
  }, {
    "filepath" : "filepath",
    "verdict" : {
      "key" : [ "verdict", "verdict" ]
    },
    "determination" : {
      "flag_list" : [ {
        "flag" : true,
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      }, {
        "flag" : true,
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      } ],
      "inapplicable_environment_list" : [ {
        "inapplicable_environment" : true,
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      }, {
        "inapplicable_environment" : true,
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      } ],
      "ignored_list" : [ {
        "ignored" : true,
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      }, {
        "ignored" : true,
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      } ],
      "verdict_list" : [ {
        "verdict" : "verdict",
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
```

```
      }, {
        "verdict" : "verdict",
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      } ],
      "dead_list" : [ {
        "dead" : true,
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      }, {
        "dead" : true,
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      } ],
      "dangerous_construct_list" : [ {
        "dangerous_construct" : "dangerous_construct",
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      }, {
        "dangerous_construct" : "dangerous_construct",
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      } ],
      "notes_list" : [ {
        "notes" : "notes",
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      }, {
        "notes" : "notes",
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      } ]
    },
    "meta_alert_id" : "meta_alert_id",
    "line_start" : 0,
    "condition_id" : "condition_id",
    "alert_ids" : [ "alert_ids", "alert_ids" ]
  } ],
  "new_taxonomies" : [ {
    "taxonomy_id" : "taxonomy_id",
    "taxonomy_version" : "taxonomy_version",
    "conditions" : [ {
      "condition_name" : "condition_name",
      "title" : "title",
      "condition_id" : "condition_id",
      "platform" : "platform"
    }, {
      "condition_name" : "condition_name",
      "title" : "title",
      "condition_id" : "condition_id",
      "platform" : "platform"
    } ],
    "taxonomy_fields" : "{}",
    "taxonomy_name" : "taxonomy_name"
```

```
    }, {
      "taxonomy_id" : "taxonomy_id",
      "taxonomy_version" : "taxonomy_version",
      "conditions" : [ {
        "condition_name" : "condition_name",
        "title" : "title",
        "condition_id" : "condition_id",
        "platform" : "platform"
      }, {
        "condition_name" : "condition_name",
        "title" : "title",
        "condition_id" : "condition_id",
        "platform" : "platform"
      } ],
      "taxonomy_fields" : "{}",
      "taxonomy_name" : "taxonomy_name"
    } ],
    "project_id" : "project_id",
    "project_name" : "project_name"
  }, {
    "project_description" : "project_description",
    "code_language" : "code_language",
    "package" : {
      "alerts" : [ {
        "alert_id" : "alert_id",
        "primary_message" : {
          "filepath" : "filepath",
          "line_start" : 6,
          "line_end" : 1
        },
        "tool_id" : "tool_id",
        "checker_id" : "checker_id",
        "more_messages" : [ "", "" ]
      }, {
        "alert_id" : "alert_id",
        "primary_message" : {
          "filepath" : "filepath",
          "line_start" : 6,
          "line_end" : 1
        },
        "tool_id" : "tool_id",
        "checker_id" : "checker_id",
        "more_messages" : [ "", "" ]
      } ],
      "test_suite_id" : "test_suite_id",
      "updated_at" : "2000-01-23T04:56:07.000+00:00",
      "package_name" : "package_name",
```

```
    "created_at" : "2000-01-23T04:56:07.000+00:00",
    "package_id" : "package_id",
    "tools" : [ {
      "tool_name" : "tool_name",
      "tool_version" : "tool_version",
      "checker_data" : [ {
        "checker_name" : "checker_name",
        "checker_id" : "checker_id",
        "conditions" : [ {
          "condition_name" : "condition_name",
          "title" : "title",
          "condition_id" : "condition_id",
          "platform" : "platform"
        }, {
          "condition_name" : "condition_name",
          "title" : "title",
          "condition_id" : "condition_id",
          "platform" : "platform"
        } ]
      }, {
        "checker_name" : "checker_name",
        "checker_id" : "checker_id",
        "conditions" : [ {
          "condition_name" : "condition_name",
          "title" : "title",
          "condition_id" : "condition_id",
          "platform" : "platform"
        }, {
          "condition_name" : "condition_name",
          "title" : "title",
          "condition_id" : "condition_id",
          "platform" : "platform"
        } ]
      } ],
      "code_metrics_data" : "{}",
      "tool_id" : "tool_id",
      "category" : "category"
    }, {
      "tool_name" : "tool_name",
      "tool_version" : "tool_version",
      "checker_data" : [ {
        "checker_name" : "checker_name",
        "checker_id" : "checker_id",
        "conditions" : [ {
          "condition_name" : "condition_name",
          "title" : "title",
          "condition_id" : "condition_id",
```

```
            "platform" : "platform"
          }, {
            "condition_name" : "condition_name",
            "title" : "title",
            "condition_id" : "condition_id",
            "platform" : "platform"
          } ]
        }, {
          "checker_name" : "checker_name",
          "checker_id" : "checker_id",
          "conditions" : [ {
            "condition_name" : "condition_name",
            "title" : "title",
            "condition_id" : "condition_id",
            "platform" : "platform"
          }, {
            "condition_name" : "condition_name",
            "title" : "title",
            "condition_id" : "condition_id",
            "platform" : "platform"
          } ]
        } ],
        "code_metrics_data" : "{}",
        "tool_id" : "tool_id",
        "category" : "category"
      } ],
      "package_description" : "package_description"
    },
    "existing_taxonomies" : [ "existing_taxonomies", "exist-
ing_taxonomies" ],
    "meta_alerts" : [ {
      "filepath" : "filepath",
      "verdict" : {
        "key" : [ "verdict", "verdict" ]
      },
      "determination" : {
        "flag_list" : [ {
          "flag" : true,
          "timestamp" : "2000-01-23T04:56:07.000+00:00"
        }, {
          "flag" : true,
          "timestamp" : "2000-01-23T04:56:07.000+00:00"
        } ],
        "inapplicable_environment_list" : [ {
          "inapplicable_environment" : true,
          "timestamp" : "2000-01-23T04:56:07.000+00:00"
        }, {
```

```
        "inapplicable_environment" : true,
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      } ],
      "ignored_list" : [ {
        "ignored" : true,
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      }, {
        "ignored" : true,
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      } ],
      "verdict_list" : [ {
        "verdict" : "verdict",
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      }, {
        "verdict" : "verdict",
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      } ],
      "dead_list" : [ {
        "dead" : true,
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      }, {
        "dead" : true,
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      } ],
      "dangerous_construct_list" : [ {
        "dangerous_construct" : "dangerous_construct",
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      }, {
        "dangerous_construct" : "dangerous_construct",
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      } ],
      "notes_list" : [ {
        "notes" : "notes",
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      }, {
        "notes" : "notes",
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      } ]
    },
    "meta_alert_id" : "meta_alert_id",
    "line_start" : 0,
    "condition_id" : "condition_id",
    "alert_ids" : [ "alert_ids", "alert_ids" ]
  }, {
    "filepath" : "filepath",
    "verdict" : {
      "key" : [ "verdict", "verdict" ]
```

```
    },
    "determination" : {
      "flag_list" : [ {
        "flag" : true,
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      }, {
        "flag" : true,
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      } ],
      "inapplicable_environment_list" : [ {
        "inapplicable_environment" : true,
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      }, {
        "inapplicable_environment" : true,
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      } ],
      "ignored_list" : [ {
        "ignored" : true,
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      }, {
        "ignored" : true,
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      } ],
      "verdict_list" : [ {
        "verdict" : "verdict",
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      }, {
        "verdict" : "verdict",
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      } ],
      "dead_list" : [ {
        "dead" : true,
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      }, {
        "dead" : true,
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      } ],
      "dangerous_construct_list" : [ {
        "dangerous_construct" : "dangerous_construct",
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      }, {
        "dangerous_construct" : "dangerous_construct",
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
      } ],
      "notes_list" : [ {
        "notes" : "notes",
        "timestamp" : "2000-01-23T04:56:07.000+00:00"
```

```
        }, {
          "notes" : "notes",
          "timestamp" : "2000-01-23T04:56:07.000+00:00"
        } ]
      },
      "meta_alert_id" : "meta_alert_id",
      "line_start" : 0,
      "condition_id" : "condition_id",
      "alert_ids" : [ "alert_ids", "alert_ids" ]
    } ],
    "new_taxonomies" : [ {
      "taxonomy_id" : "taxonomy_id",
      "taxonomy_version" : "taxonomy_version",
      "conditions" : [ {
        "condition_name" : "condition_name",
        "title" : "title",
        "condition_id" : "condition_id",
        "platform" : "platform"
      }, {
        "condition_name" : "condition_name",
        "title" : "title",
        "condition_id" : "condition_id",
        "platform" : "platform"
      } ],
      "taxonomy_fields" : "{}",
      "taxonomy_name" : "taxonomy_name"
    }, {
      "taxonomy_id" : "taxonomy_id",
      "taxonomy_version" : "taxonomy_version",
      "conditions" : [ {
        "condition_name" : "condition_name",
        "title" : "title",
        "condition_id" : "condition_id",
        "platform" : "platform"
      }, {
        "condition_name" : "condition_name",
        "title" : "title",
        "condition_id" : "condition_id",
        "platform" : "platform"
      } ],
      "taxonomy_fields" : "{}",
      "taxonomy_name" : "taxonomy_name"
    } ],
    "project_id" : "project_id",
    "project_name" : "project_name"
  } ],
  "request_id" : "request_id"
```

```
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

Return Existing Projects get_projects_response

400

Invalid Request

404

Projects Unavailable

default

Unexpected Error error

Up

`GET /taxonomies`

Get a list of Taxonomies available in the Modules. This function is used to send information to the DataHub from the UI and Stats modules. (**getTaxonomies**)

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request headers

Return type

taxonomy_response

Example data

Content-Type: application/json

```
{
  "request_id" : "request_id",
  "taxonomy_list" : [ {
    "taxonomy_id" : "taxonomy_id",
    "taxonomy_version" : "taxonomy_version",
    "taxonomy_name" : "taxonomy_name"
  }, {
    "taxonomy_id" : "taxonomy_id",
    "taxonomy_version" : "taxonomy_version",
    "taxonomy_name" : "taxonomy_name"
  } ]
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

Returns a list of taxonomy ids, names and versions. taxonomy_response

400

Invalid Request

404

Taxonomies Unavailable

default

Unexpected Error error

Up

`GET /taxonomies/{taxonomy_id}`

Get data of a specific taxonomy based on the name and version. This function is used to send information to the DataHub from the UI and Stats modules. (**getTaxonomy**)

Path parameters

taxonomy_id (required)

Path Parameter — Taxonomy id to retrieve data

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request headers

Return type

get_taxonomy_response

Example data

Content-Type: application/json


```
{
  "taxonomy" : {
    "taxonomy_id" : "taxonomy_id",
    "taxonomy_version" : "taxonomy_version",
    "conditions" : [ {
      "condition_name" : "condition_name",
      "title" : "title",
      "condition_id" : "condition_id",
      "platform" : "platform"
    }, {
      "condition_name" : "condition_name",
      "title" : "title",
      "condition_id" : "condition_id",
      "platform" : "platform"
    } ],
    "taxonomy_fields" : "{}",
    "taxonomy_name" : "taxonomy_name"
  },
  "request_id" : "request_id"
}
```


Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

Returns data for a specific taxonomy get_taxonomy_response

400

Invalid Request

404

Taxonomy Unavailable

default

Unexpected Error error

Up

`GET /test_suites`

Get a list of Test Suite ids, names and versions that are available. This function is used to send information to the DataHub from the UI and Stats modules. (**getTest-SuiteList**)

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request headers

Return type

test_suite_response

Example data

Content-Type: application/json

```
{
  "request_id" : "request_id",
  "test_suite_list" : [ {
    "test_suite_version" : "test_suite_version",
    "test_suite_id" : "test_suite_id",
    "test_suite_name" : "test_suite_name"
  }, {
```

```
    "test_suite_version" : "test_suite_version",
    "test_suite_id" : "test_suite_id",
    "test_suite_name" : "test_suite_name"
  } ]
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

Returns a list of test suites test_suite_response

400

Invalid Request

404

Test Suites Unavailable

default

Unexpected Error error

Up

`GET /tools/{tool_id}`

Get specific tool information from the DataHub Module. This function is used to send information to the DataHub from the UI and Stats modules. (**getToolData**)

Path parameters

tool_id (required)

Path Parameter — Tool id to retrieve data

Request headers

Return type

get_tool_response

Example data

Content-Type: application/json

```
{
  "request_id" : "request_id",
  "tool" : {
    "tool_name" : "tool_name",
    "tool_version" : "tool_version",
    "checker_data" : [ {
      "checker_name" : "checker_name",
      "checker_id" : "checker_id",
      "conditions" : [ {
        "condition_name" : "condition_name",
        "title" : "title",
        "condition_id" : "condition_id",
        "platform" : "platform"
      }, {
        "condition_name" : "condition_name",
        "title" : "title",
        "condition_id" : "condition_id",
        "platform" : "platform"
      } ]
    }, {
      "checker_name" : "checker_name",
      "checker_id" : "checker_id",
      "conditions" : [ {
        "condition_name" : "condition_name",
        "title" : "title",
        "condition_id" : "condition_id",
        "platform" : "platform"
      }, {
        "condition_name" : "condition_name",
        "title" : "title",
        "condition_id" : "condition_id",
        "platform" : "platform"
      } ]
    } ],
    "code_metrics_data" : "{}",
    "tool_id" : "tool_id",
    "category" : "category"
  }
}
```

Produces

This API call produces the following media types according to the Accept request

header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

Returns data for a particular tool get_tool_response

400

Invalid Request

404

Tool Information Unavailable

default

Unexpected Error error

Up

`GET /tools`

Get a list of tool ids, versions and names available in the Module. This function is used to send information to the DataHub from the UI and Stats modules. (**getToolList**)

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request headers

Return type

tool_response

Example data

Content-Type: application/json

```
{
  "tool_list" : [ {
```

```
    "tool_name" : "tool_name",
    "tool_version" : "tool_version",
    "tool_id" : "tool_id"
  }, {
    "tool_name" : "tool_name",
    "tool_version" : "tool_version",
    "tool_id" : "tool_id"
  } ],
  "request_id" : "request_id"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

Returns a list of tool id, version and name tool_response

400

Invalid Request

404

Tool Unavailable

default

Unexpected Error error

Up

`GET /packages`

Retrieve a list of all available packages (**listPackages**)

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request headers

Return type

list_packages_response

Example data

Content-Type: application/json

```
{
  "packages" : [ {
    "package_language" : "package_language",
    "language_versions" : [ "language_versions", "language_ver-
sions" ],
    "package_name" : "package_name",
    "package_id" : "package_id",
    "package_description" : "package_description"
  }, {
    "package_language" : "package_language",
    "language_versions" : [ "language_versions", "language_ver-
sions" ],
    "package_name" : "package_name",
    "package_id" : "package_id",
    "package_description" : "package_description"
  } ],
  "request_id" : "request_id"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

Return a List of Available Packages list_packages_response

400

Invalid Request

404

No Packages Available

default

Unexpected Error error

Up

`GET /projects`

Retrieve a list of all available projects (**listProjects**)

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request headers

Return type

list_projects

Example data

Content-Type: application/json

```
{
  "projects" : [ {
    "project_description" : "project_description",
    "code_language" : "code_language",
    "project_id" : "project_id",
    "package_id" : "package_id",
    "project_name" : "project_name"
  }, {
    "project_description" : "project_description",
    "code_language" : "code_language",
    "project_id" : "project_id",
    "package_id" : "package_id",
    "project_name" : "project_name"
  } ],
  "request_id" : "request_id"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

Return a List of Available Projects list_projects

400

Invalid Request

404

No Projects Available

default

Unexpected Error error

StatsToDataHub

Up

`POST /projects/adaptive_heuristics/close`

Send a list of project_ids to implement the adaptive heuristic alert forwarding close request on the DataHub. The request stops the forwarding of project alerts to the Stats Module. (**closeAdaptiveHeuristics**)

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request body

project_ids project_ids (required)

Body Parameter — Multiple project_ids to close adaptive heuristic alert forwarding request

Request headers

Return type

adaptive_heuristic_close_response

Example data

Content-Type: application/json

```
{
  "project_status" : [ {
    "status_message" : "status_message",
    "project_id" : "project_id",
    "project_updated" : true
  }, {
    "status_message" : "status_message",
    "project_id" : "project_id",
    "project_updated" : true
  } ],
  "request_id" : "request_id"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

Adaptive Heuristic Forwarding Successfully Closed adaptive_heuristic_close_response

206

Some Error Occurred adaptive_heuristic_close_response

400

Invalid Request

default

Unexpected Error error

UIToDataHub

Up

```
POST /packages
```

Create new instances of packages. This request will return the package_ids that should be used by the UI Module for referencing these packages within the DataHub. (**createPackages**)

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request body

packages_metadata_list packages_metadata_list (required)

Body Parameter — List of name, test_suite, description, and source file for packages

Request headers

Return type

create_packages_response

Example data

Content-Type: application/json

```
{
  "packages" : [ {
    "status_message" : "status_message",
    "package_name" : "package_name",
    "alert_mappings" : {
      "key" : "alert_mappings"
    },
    "package_id" : "package_id"
  }, {
    "status_message" : "status_message",
    "package_name" : "package_name",
    "alert_mappings" : {
      "key" : "alert_mappings"
    },
    "package_id" : "package_id"
  } ],
  "request_id" : "request_id"
}
```

Produces

This API call produces the following media types according to the Accept request

header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

OK, Created create_packages_response

206

Unable to Create All Packages create_packages_response

400

Unable to Create Packages

default

Unexpected Error error

Up

`POST /projects`

Create new instances of a projects. Packages and taxonomies should be uploaded to the DataHub prior to creating a project that uses them. This request will return the project_ids that should be used by the UI Module for referencing these projects within the DataHub. (**createProjects**)

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request body

multiple_project_data multiple_project_data (required)

Body Parameter — Project Data to Create

Request headers

Return type

create_projects_response

Example data

Content-Type: application/json

```
{
  "projects" : [ {
    "status_message" : "status_message",
    "project_id" : "project_id",
    "meta_alert_mappings" : {
      "key" : "meta_alert_mappings"
    },
    "project_name" : "project_name"
  }, {
    "status_message" : "status_message",
    "project_id" : "project_id",
    "meta_alert_mappings" : {
      "key" : "meta_alert_mappings"
    },
    "project_name" : "project_name"
  } ],
  "request_id" : "request_id"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

OK, All Projects Were Created Successfully create_projects_response

206

Partial Creation Completed create_projects_response

400

Unable to Create Projects

default

Unexpected Error error

Up

- `DELETE /packages/{package_id}`

Delete a specific package from the DataHub by supplying the package_id for the package to delete. (**deletePackage**)

Path parameters

package_id (required)

Path Parameter — The id of the package to retrieve

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request headers

Return type

request_token

Example data

Content-Type: application/json

```
{
  "message" : "message",
  "request_id" : "request_id"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

Package was Successfully Deleted request_token

400

Invalid Request

404

Unable to Delete Package

default

Unexpected Error error

Up

- `DELETE /projects/{project_id}`

Delete a project from the DataHub Module. Send the project_id to the DataHub to delete the project information stored. (**deleteProject**)

Path parameters

project_id (required)

Path Parameter — The id of the project to retrieve

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request headers

Return type

request_token

Example data

Content-Type: application/json

```
{
  "message" : "message",
  "request_id" : "request_id"
}
```

Produces

This API call produces the following media types according to the Accept request

header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

Project was Successfully Deleted request_token

400

Invalid Request

404

Project Unavailable

default

Unexpected Error error

Up

`PUT /packages/{package_id}`

Modify a specific package from the DataHub by supplying the package_id for the package to update. (**editPackage**)

Path parameters

package_id (required)

Path Parameter — The id of the package to retrieve

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request body

edit_package **edit_package** (required)

Body Parameter — Package data to modify the existing package with

Request headers

Return type

edit_package_response

Example data

Content-Type: application/json

```
{
  "alert_mappings" : {
    "key" : "alert_mappings"
  },
  "message" : "message",
  "request_id" : "request_id"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

Package was Successfully Updated edit_package_response

400

Invalid Request

404

Unable to Update Package

default

Unexpected Error error

Up

`PUT /projects/{project_id}`

Modify a specific project from the DataHub by supplying the project_id for the project to update. (**editProject**)

Path parameters

project_id (required)

Path Parameter — The id of the project to retrieve

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request body

edit_project edit_project (required)

Body Parameter — Project data to modify the existing project with

Request headers

Return type

edit_project_response

Example data

Content-Type: application/json

```
{
  "meta_alert_mappings" : {
    "key" : "meta_alert_mappings"
  },
  "message" : "message",
  "request_id" : "request_id"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

Package was Successfully Updated edit_project_response

400

Invalid Request

404

Unable to Update Project

default

Unexpected Error error

Up

`POST /packages/{package_id}/alerts`

Upload alerts for a package. Upload new alerts to the DataHub to add to a specific package. (**sendAlertsForPackage**)

Path parameters

package_id (required)

Path Parameter — The id of the package to retrieve

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request body

alerts upload_alerts (required)

Body Parameter —

Request headers

Return type

upload_alerts_response

Example data

Content-Type: application/json

```
{
  "alerts" : [ {
```

```
    "alert_id" : "alert_id",
    "primary_message" : {
      "filepath" : "filepath",
      "line_start" : 6,
      "line_end" : 1
    },
    "checker_id" : "checker_id",
    "message" : "message"
  }, {
    "alert_id" : "alert_id",
    "primary_message" : {
      "filepath" : "filepath",
      "line_start" : 6,
      "line_end" : 1
    },
    "checker_id" : "checker_id",
    "message" : "message"
  } ],
  "request_id" : "request_id"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

Return the Alert Ids upload_alerts_response

400

Unable to Upload Alerts

default

Unexpected Error error

Up

`POST /projects/{project_id}/alerts`

Upload meta-alerts for a specific project. This method can also be used to send only

meta-alert determinations of an existing meta-alert to the DataHub, which is most useful when a project has an open adaptive heuristic set. (**sendMetaAlertDeterminations**)

Path parameters

project_id (required)

Path Parameter — The id of the project

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request body

meta_alerts upload_meta_alerts (required)

Body Parameter —

Request headers

Return type

upload_meta_alerts_response

Example data

Content-Type: application/json

```
{
  "meta_alerts" : [ {
    "meta_alert_id" : "meta_alert_id",
    "message" : "message",
    "condition_id" : "condition_id",
    "alert_ids" : [ "alert_ids", "alert_ids" ]
  }, {
    "meta_alert_id" : "meta_alert_id",
    "message" : "message",
    "condition_id" : "condition_id",
    "alert_ids" : [ "alert_ids", "alert_ids" ]
  } ],
  "request_id" : "request_id"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

Return the Meta-Alert IDs Associated with the Project upload_meta_alerts_response

400

Unable to Upload Meta-Alerts

default

Unexpected Error error

Up

`POST /test_suites`

Upload a Test Suite to the DataHub Module. (**uploadTestSuite**)

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request body

test_suite_data test_suite_data (required)

Body Parameter — Test Suite information to upload

Request headers

Return type

test_suite_upload_response

Example data

Content-Type: application/json

```
{
    "test_suite_id": "string",
    "test_suite_name": "string",
    "test_suite_version": "string",
    "request_id": "string",
    "status_message": "string"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

OK test_suite_upload_response

400

Unable to Upload Test Suite

default

Unexpected Error error

Up

`POST /tools`

Upload new tool data to the Module. Returns a tool_id for future referencing the tool in the DataHub Module. (**uploadTool**)

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request body

tool_data tool_data (required)

Body Parameter — Tool information to upload

Request headers

Return type

tool_upload_response

Example data

Content-Type: application/json

```
{
  "tool_name" : "tool_name",
  "tool_version" : "tool_version",
  "tool_id" : "tool_id",
  "request_id" : "request_id"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

OK, Tool Successfully Uploaded tool_upload_response

400

Unable to Upload Tool Information

default

Unexpected Error error

Rapid Models Prioritization Module API Definition

UIToPrioritization

Up

`POST /priorities`

Create a new prioritization scheme (**createPrioritization**)

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request body

create_prioritization_data create_prioritization_data (required)

Body Parameter — Prioritization scheme to create

Request headers

Return type

create_prioritization_response

Example data

Content-Type: application/json

```
{
  "priority_scheme_id" : "priority_scheme_id",
  "priority_scheme_name" : "priority_scheme_name",
  "request_id" : "request_id"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

OK, Created create_prioritization_response

400

Invalid Request

401

Global and Remote Flags Cannot Both be True

405

Cannot Create Prioritization Scheme

default

Unexpected Error error

Up

- `DELETE /priorities/{priority_scheme_id}/projects/{project_id}`

Delete a specific prioritization scheme (**deletePrioritization**)

Path parameters

priority_scheme_id (required)

Path Parameter — The id of the prioritization scheme

project_id (required)

Path Parameter — The id of the project associated with this project

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request headers

Return type

request_token

Example data

Content-Type: application/json

```
{
  "message" : "message",
  "request_id" : "request_id"
}
```

Responses

200

OK request_token

400

Invalid Request

405

Cannot Delete Prioritization Scheme

default

Unexpected Error error

Up

`GET /priorities/{priority_scheme_id}/projects/{project_id}`

Retrieve a specific prioritization scheme (**getPrioritization**)

Path parameters

priority_scheme_id (required)

Path Parameter — The id of the prioritization scheme

project_id (required)

Path Parameter — The id of the project associated with this project

Request headers

Return type

priority_scheme_data

Example data

Content-Type: application/json

```
{
  "priority_scheme_name" : "priority_scheme_name",
  "is_remote" : true,
  "formula" : "formula",
  "is_global" : true,
  "request_id" : "request_id",
```

```
  "weighted_columns" : "{}"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

OK priority_scheme_data

400

Invalid Request

404

Prioritization Scheme Unavailable

default

Unexpected Error error

Up

`GET /priorities`

List all prioritization schemes (**listPrioritizations**)

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request headers

Return type

prioritization_list

Example data

Content-Type: application/json

```
{
  "priority_list" : [ {
    "priority_scheme_id" : "priority_scheme_id",
    "priority_scheme_name" : "priority_scheme_name"
  }, {
    "priority_scheme_id" : "priority_scheme_id",
    "priority_scheme_name" : "priority_scheme_name"
  } ],
  "request_id" : "request_id"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

OK prioritization_list

404

Prioritization Schemes Not Found

default

Unexpected Error error

Up

`PUT /priorities/{priority_scheme_id}`

Update an existing prioritization scheme (**updatePrioritization**)

Path parameters

priority_scheme_id (required)

Path Parameter — The id of the prioritization scheme

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request body

update_priority_data update_priority_data (required)

Body Parameter — Prioritization Scheme to update

Request headers

Return type

request_token

Example data

Content-Type: application/json

```
{
  "message" : "message",
  "request_id" : "request_id"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

OK request_token

400

Invalid Request

404

Prioritization Scheme Unavailable

405

Cannot Update Prioritization Scheme

default

Unexpected Error error

Rapid Models Registration and Login Module API Definition

UIToRegistration

Up

`GET /server/{server_name}`

Get access token to use other servers (**getServerAccess**)

Path parameters

server_name (required)

Path Parameter — Name of the server to grant access to, expected values [statistics, datahub, prioritization]

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request headers

Return type

access_token

Example data

Content-Type: application/json

```
{
  "x_access_token" : "x_access_token"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

OK access_token

400

Invalid Request

405

Server Access Unavailable

default

Unexpected Error error

Up

`POST /login`

Login page; Authenticate to the SCAIFE system (**loginUser**)

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request body

login_credentials login_credentials (optional)

Body Parameter — Login credentials for the user.

Return type

access_token

Example data

Content-Type: application/json

```
{
  "x_access_token" : "x_access_token"
```

```
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

OK access_token

400

Invalid Request

405

Login Unavailable

default

Unexpected Error error

Up

`POST /register`

Registration page; Create new users in the SCAIFE system (**registerUsers**)

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request body

user_information user_information (required)

Body Parameter — User information

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

201

User Created

400

Invalid Request

405

Registration Unavailable

default

Unexpected Error error

Rapid Models Statistics Module API Definition

DataHubToStats

Up

`PUT /projects/{project_id}/{package_id}/alerts`

Forward new Alerts that have been uploaded to the DataHub and have a current open adaptive heuristic request for its respective package. Returns status message for the DataHub to track if the request was completed. (**sendAlertUpdatesForClassifier**)

Path parameters

project_id (required)

Path Parameter — The id of the project associated with these alerts

package_id (required)

Path Parameter — The id of the package associated with these alerts

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request body

multiple_alerts multiple_alerts (required)

Body Parameter — Updated alert data

Request headers

Return type

alert_updates_response

Example data

Content-Type: application/json

```
{
  "project_id" : "project_id",
  "package_id" : "package_id"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

OK alert_updates_response

400

Unable to Upload Alerts

404

Invalid Package

default

Unexpected Error error

Up

```
PUT /packages/tools/{tool_id}
```

Send FFSA or code metrics tool info to the Stats Module. When a new tool is uploaded, the DataHub can send new tool info for packages with open adaptive heuristic requests automatically to keep the Stats Module in sync. (**sendNewTool**)

Path parameters

tool_id (required)

Path Parameter — The id of the tool uploaded to the DataHub

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request body

packages_with_tool_data packages_with_tool_data (required)

Body Parameter — Tool info, including name, version, plus FFSA checker info OR code metrics field info and package ids associated with this new tool.

Request headers

Return type

Integer

Example data

Content-Type: application/json

```
0
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

OK Integer

400

Unable to Upload Tool Information

default

Unexpected Error error

UIToStats

Up

```
PUT /classifiers/{classifier_instance_id}/adaptive_heuris-
tics/close
```

Stop adaptive heuristic forward request. Send a request to close (set to false) the adaptive heuristic for the packages listed in the classifier instance. (**closeAdaptiveHeuristicDataForwarding**)

Path parameters

classifier_instance_id (required)

Path Parameter — The id of the classifier instance to run on the target domain

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request headers

Return type

close_adaptive_heuristics_response

Example data

Content-Type: application/json

```
{
  "message" : "message",
  "classifier_instance_id" : "classifier_instance_id"
}
```

Responses

200

OK close_adaptive_heuristics_response

400

Invalid Close Request

404

Classifier Instance Unavailable

default

Unexpected Error error

Up

`POST /classifiers`

Create a new classifier instance. Send Classifier information including Automated Hyper-Parameter Optimization (AHPO) and Adaptive Heuristics to the Stats Module along with package_ids for packages to use in creating/training a classifier. Returns an id that is used to then run the classifier and any additional information for the classifier. (**createClassifierInstance**)

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request body

classifier_instance classifier_instance (required)

Body Parameter — Classifier information to create

Request headers

Return type

create_classifier_response

Example data

Content-Type: application/json

```
{
  "analysis_messages" : "{}",
  "project_id" : "project_id",
  "classifier_instance_id" : "classifier_instance_id"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

OK create_classifier_response

400

Unable to Create Classifier

default

Unexpected Error error

Up

- `DELETE /classifiers/{classifier_instance_id}`

Delete a specific classifier from the Stats module by classifier_instance_id. (**delete-ClassifierInstance**)

Path parameters

classifier_instance_id (required)

Path Parameter — The id of the classifier to delete

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request headers

Return type

request_token

Example data

Content-Type: application/json

```
{
  "message" : "message",
  "request_id" : "request_id"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

Classifier Successfully Deleted request_token

400

Invalid Request

404

Unable to Delete Classifier

default

Unexpected Error error

Up

`PUT /classifiers/{classifier_instance_id}`

Edit a specific classifier from the Stats module by classifier_instance_id. (**editClassifierInstance**)

Path parameters

classifier_instance_id (required)

Path Parameter — The id of the classifier instance to edit

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request body

classifier_instance classifier_instance (required)

Body Parameter — Classifier information to edit

Request headers

Return type

create_classifier_response

Example data

Content-Type: application/json

```
{
  "analysis_messages" : "{}",
  "project_id" : "project_id",
  "classifier_instance_id" : "classifier_instance_id"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

OK create_classifier_response

400

Unable to Edit Classifier

404

Invalid Request

default

Unexpected Error error

Up

`GET /classifiers/{classifier_instance_id}`

Get analysis for a specific Classifier including performance metrics. (**getClassifierInstanceAnalysis**)

Path parameters

classifier_instance_id (required)

Path Parameter — The id of the classifier to get analysis info

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request headers

Return type

analysis_results

Example data

Content-Type: application/json

```
{
  "classifier_analysis" : "{}",
  "classifier_instance_id" : "classifier_instance_id"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

- `text/html`

Responses

200

OK analysis_results

404

Classifier Information Unavailable

default

Unexpected Error error

Up

`GET /classifiers`

List all classifiers and their associated data. Use the ids returned from this request to work with classifiers. (**listClassifiers**)

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request headers

Return type

array[list_classifiers_response]

Example data

Content-Type: application/json

```
[ {
  "classifier_id" : "classifier_id",
  "adaptive_heuristics" : [ {
    "adaptive_heuristic_name" : "adaptive_heuristic_name",
    "adaptive_heuristic_id" : "adaptive_heuristic_id",
    "adaptive_heuristic_parameters" : "{}"
  }, {
    "adaptive_heuristic_name" : "adaptive_heuristic_name",
    "adaptive_heuristic_id" : "adaptive_heuristic_id",
```

```
    "adaptive_heuristic_parameters" : "{}"
  } ],
  "ahpos" : [ {
    "ahpo_id" : "ahpo_id",
    "ahpo_name" : "ahpo_name"
  }, {
    "ahpo_id" : "ahpo_id",
    "ahpo_name" : "ahpo_name"
  } ],
  "classifier_name" : "classifier_name"
}, {
  "classifier_id" : "classifier_id",
  "adaptive_heuristics" : [ {
    "adaptive_heuristic_name" : "adaptive_heuristic_name",
    "adaptive_heuristic_id" : "adaptive_heuristic_id",
    "adaptive_heuristic_parameters" : "{}"
  }, {
    "adaptive_heuristic_name" : "adaptive_heuristic_name",
    "adaptive_heuristic_id" : "adaptive_heuristic_id",
    "adaptive_heuristic_parameters" : "{}"
  } ],
  "ahpos" : [ {
    "ahpo_id" : "ahpo_id",
    "ahpo_name" : "ahpo_name"
  }, {
    "ahpo_id" : "ahpo_id",
    "ahpo_name" : "ahpo_name"
  } ],
  "classifier_name" : "classifier_name"
} ]
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

OK

404

Classifiers Unavailable

default

Unexpected Error error

Up

`PUT /classifiers/{classifier_instance_id}/retrain`

Returns new probability values if there are updates from the selected classifier_instance. (**retrain**)

Path parameters

classifier_instance_id (required)

Path Parameter — The id of the classifier instance to run on the target domain

Consumes

This API call consumes the following media types via the Content-Type request header:

- `application/json`

Request body

classifier_instance_data classifier_instance_data (required)

Body Parameter — Information to send close adaptive heuristic request

Request headers

Return type

classifier_results

Example data

Content-Type: application/json

```
{
  "probability_data" : [ {
    "probability" : 0.8008281904610115,
    "meta_alert_id" : "meta_alert_id"
  }, {
    "probability" : 0.8008281904610115,
    "meta_alert_id" : "meta_alert_id"
  } ],
  "project_id" : "project_id",
  "classifier_analysis" : "{}",
```

```
  "classifier_instance_id" : "classifier_instance_id"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- `application/json`

Responses

200

OK classifier_results

400

Unable to Run Classifier

404

Invalid Request

default

Unexpected Error error

Up

`PUT /classifiers/{classifier_instance_id}/projects/{project_id}`

Run a specific classifier instance on the project identified by id in the path. The response contains estimated class probablities and analysis of classifier performance. (**runClassifierInstance**)

Path parameters

classifier_instance_id (required)

Path Parameter — The id of the classifier instance to run on the target domain

project_id (required)

Path Parameter — The id of the project containing the packages

Consumes

This API call consumes the following media types via the Content-Type request header:

- application/json

Request headers

Return type

classifier_results

Example data

Content-Type: application/json

```
{
  "probability_data" : [ {
    "probability" : 0.8008281904610115,
    "meta_alert_id" : "meta_alert_id"
  }, {
    "probability" : 0.8008281904610115,
    "meta_alert_id" : "meta_alert_id"
  } ],
  "project_id" : "project_id",
  "classifier_analysis" : "{}",
  "classifier_instance_id" : "classifier_instance_id"
}
```

Produces

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

Responses

200

OK classifier_results

400

Unable to Run Classifier

404

Invalid Request

default

Unexpected Error error

Models

[Jump to Methods]

Summary of API Models

39. list_classifiers_response_ahpos-
40. list_packages_response-
41. list_projects-
42. login_credentials-
43. message-
44. meta_alert-
45. meta_alert_no_id-
46. multiple_alerts-
47. multiple_project_data-
48. package-
49. package_info-
50. package_metadata-
51. package_response-
52. packages_metadata_list-
53. packages_with_tool_data-
54. packages_with_tool_data_tool_data-
55. prioritization_list-
56. prioritization_list_priority_list-
57. priority_scheme_data-
58. project-
59. project_heuristic_message-
60. project_ids-
61. project_info-
62. project_metadata-
63. project_response-
64. projects_requested-
65. projects_requested_project_status-
66. request_token-
67. secondary_message-
68. stats_checker-
69. taxonomy-
70. taxonomy_list-
71. taxonomy_response-
72. test_suite_data-
73. test_suite_list-
74. test_suite_response-
75. test_suite_upload_response-
76. tool-
77. tool_data-
78. tool_list-
79. tool_response-
80. tool_upload_response-
81. tools_taxonomies_present-

82. update_priority_data -
83. upload_alerts -
84. upload_alerts_inner -
85. upload_alerts_response -
86. upload_alerts_response_alerts -
87. upload_meta_alerts -
88. upload_meta_alerts_inner -
89. upload_meta_alerts_response -
90. upload_meta_alerts_response_meta_alerts -
91. user_information -

access_token - Up

x_access_token (optional)
String

adaptive_heuristic_close_response - Up

request_id
String

project_status (optional)
array[project_heuristic_message]

alert - Up

alert_id
String

tool_id
String

checker_id
String

primary_message
message

more_messages (optional)
array[secondary_message]

alert_no_id - Up

tool_id
String

checker_id
String

primary_message

message

more_messages (optional)
array[secondary_message]

alert_updates_response - Up

project_id (optional)
String

package_id (optional)
String

analysis_results - Up

classifier_instance_id (optional)
String

classifier_analysis (optional)
Object

checker - Up

checker_id
String

checker_name (optional)
String

conditions (optional)
array[condition]

classifier_instance - Up

classifier_id
String

classifier_type (optional)
String

classifier_instance_name (optional)
String

project_ids
array[String]

ahpo_id (optional)
String

ahpo_data (optional)
map[String, Object]

adaptive_heuristic_id (optional)
String

adaptive_heuristic_parameters (optional)
map[String, Object]

classifier_instance_data - Up

project_id
String

The ID of the target project to run the classifier on

timestamp
Date

The current time format: date-time

classifier_results - Up

classifier_instance_id (optional)
String

project_id
String

ID of project in the target domain

probability_data
array[classifier_results_probability_data]

classifier_analysis (optional)
Object

classifier_results_probability_data - Up

meta_alert_id (optional)
String

probability (optional)
Double

format: double

close_adaptive_heuristics_response - Up

classifier_instance_id (optional)
String

message (optional)
String

condition - Up

condition_id
String

condition_name (optional)
String

title
String

platform (optional)
String

create_classifier_response - Up

classifier_instance_id (optional)
String

project_id (optional)
String

analysis_messages (optional)
Object

Additional information that will help to understand this classifier instance's performance

create_packages_response - Up

request_id
String

packages (optional)
array[package_response]

create_prioritization_data - Up

priority_scheme_name
String

project_ids (optional)
array[String]

formula
String

weighted_columns (optional)
Object

is_global

Boolean

is_remote
Boolean

create_prioritization_response - Up

priority_scheme_id (optional)
String

priority_scheme_name (optional)
String

request_id (optional)
String

create_projects_response - Up

request_id
String

projects (optional)
array[project_response]

determination - Up

flag_list (optional)
array[determination_flag_list]

verdict_list (optional)
array[determination_verdict_list]

ignored_list (optional)
array[determination_ignored_list]

dead_list (optional)
array[determination_dead_list]

inapplicable_environment_list (optional)
array[determination_inapplicable_environment_list]

dangerous_construct_list (optional)
array[determination_dangerous_construct_list]

notes_list (optional)
array[determination_notes_list]

determination_dangerous_construct_list - Up

dangerous_construct (optional)
String

timestamp (optional)
Date

format: date-time

determination_dead_list - Up

dead (optional)
Boolean

timestamp (optional)
Date

format: date-time

determination_flag_list - Up

flag (optional)
Boolean

timestamp (optional)
Date

format: date-time

determination_ignored_list - Up

ignored (optional)
Boolean

timestamp (optional)
Date

format: date-time

determination_inapplicable_environment_list - Up

inapplicable_environment (optional)
Boolean

timestamp (optional)
Date

format: date-time

determination_notes_list - Up

notes (optional)
String

timestamp (optional)
Date

format: date-time

determination_verdict_list - Up

verdict (optional)
String

timestamp (optional)
Date

format: date-time

edit_package - Up

package_name (optional)
String

package_description (optional)
String

tool_ids (optional)
array[String]

alerts (optional)
array[alert]

edit_package_response - Up

request_id (optional)
String

alert_mappings (optional)
map[String, String]

message (optional)
String

edit_project - Up

project_name (optional)
String

project_description (optional)
String

taxonomy_ids (optional)
array[String]

meta_alerts (optional)
array[meta_alert]

edit_project_response - Up

request_id (optional)
String

meta_alert_mappings (optional)
map[String, String]

message (optional)
String

error - Up

code
Integer

message
String

get_alerts_response - Up

meta_alerts (optional)
array[meta_alert]

alerts (optional)
array[alert]

request_id (optional)
String

get_package_response - Up

request_id
String

package
package

get_projects_response - Up

request_id (optional)
String

projects (optional)
array[project]

get_taxonomy_response - Up

request_id (optional)
String

taxonomy (optional)
taxonomy

get_tool_response - Up

request_id (optional)
String

tool (optional)
tool

list_classifiers_response - Up

classifier_id (optional)
String

classifier_name (optional)
String

ahpos (optional)
array[list_classifiers_response_ahpos]

adaptive_heuristics (optional)
array[list_classifiers_response_adaptive_heuristics]

list_classifiers_response_adaptive_heuristics - Up

adaptive_heuristic_id (optional)
String

adaptive_heuristic_name (optional)
String

adaptive_heuristic_parameters (optional)
Object

list_classifiers_response_ahpos - Up

ahpo_id (optional)
String

ahpo_name (optional)
String

list_packages_response - Up

request_id
String

packages
array[package_info]

list_projects - Up

request_id (optional)
String

projects (optional)
array[project_info]

login_credentials - Up

username (optional)
String

password (optional)
String

message - Up

line_start
Integer

line_end (optional)
Integer

filepath
String

meta_alert - Up

meta_alert_id
String

alert_ids (optional)
array[String]

filepath (optional)
String

line_start (optional)
Integer

condition_id
String

determination (optional)
determination

verdict (optional)
map[String, array[String]]

meta_alert_no_id - Up

alert_ids (optional)
array[String]

condition_id

String

filepath (optional)
String

line_start (optional)
Integer

determination (optional)
determination

verdict (optional)
map[String, array[String]]

multiple_alerts - Up

meta_alerts (optional)
array[meta_alert]

alerts (optional)
array[alert]

multiple_project_data - Up

array[project_metadata]

package - Up

package_id (optional)
String

package_name (optional)
String

package_description (optional)
String

test_suite_id (optional)
String

tools (optional)
array[tool]

alerts (optional)
array[alert]

created_at (optional)
Date

format: date-time

updated_at (optional)
Date

format: date-time

package_info - Up

package_id (optional)
String

package_name
String

package_language (optional)
String

language_versions (optional)
array[String]

package_description
String

package_metadata - Up

package_name
String

package_description
String

Description of the package

package_language
String

language_versions (optional)
array[String]

codesource_file (optional)
byte[]

Source file to upload format: binary

codesource_url (optional)
String

Link to the source code format: uri

test_suite_id (optional)
String

ID of the associated test suite

alerts (optional)
array[alert_no_id]

tool_ids (optional)
array[String]

package_response - Up

package_id (optional)
String

package_name (optional)
String

alert_mappings (optional)
map[String, String]

status_message (optional)
String

Message indicating whether the package was created. Expected Messages include Created, Unable to Create Package, and Invalid Request.

packages_metadata_list - Up

array[package_metadata]

packages_with_tool_data - Up

package_ids (optional)
array[String]

tool_data (optional)
packages_with_tool_data_tool_data

packages_with_tool_data_tool_data - Up

tool_name (optional)
String

tool_version (optional)
String

tool_type (optional)
String

checker_data (optional)
array[stats_checker]

code_metrics_data (optional)

Object

prioritization_list - Up

request_id (optional)
String

priority_list (optional)
array[prioritization_list_priority_list]

prioritization_list_priority_list - Up

priority_scheme_id (optional)
String

priority_scheme_name (optional)
String

priority_scheme_data - Up

priority_scheme_name
String

formula
String

weighted_columns (optional)
Object

is_global (optional)
Boolean

is_remote (optional)
Boolean

request_id (optional)
String

project - Up

project_id (optional)
String

project_name (optional)
String

project_description (optional)
String

code_language (optional)
String

package (optional)
package

meta_alerts (optional)
array[meta_alert]

existing_taxonomies (optional)
array[String]

new_taxonomies (optional)
array[taxonomy]

project_heuristic_message - Up

project_id (optional)
String

project_updated (optional)
Boolean

status_message (optional)
String

project_ids - Up

array[String]

project_info - Up

project_id (optional)
String

project_name (optional)
String

project_description (optional)
String

code_language (optional)
String

package_id (optional)
String

project_metadata - Up

project_name
String

project_description
String

code_language (optional)
String

package_id (optional)
String

meta_alerts (optional)
array[meta_alert_no_id]

taxonomy_ids (optional)
array[String]

project_response - Up

project_id (optional)
String

project_name (optional)
String

meta_alert_mappings (optional)
map[String, String]

status_message (optional)
String

Message indicating whether the project was created. Expected Messages include Created, Unable to Create Project, and Invalid Request.

projects_requested - Up

project_status (optional)
array[projects_requested_project_status]

tool_taxonomies_present (optional)
tools_taxonomies_present

projects_requested_project_status - Up

project_id (optional)
String

set_adaptive_heuristic_open (optional)
Boolean

request_token - Up

request_id
String

ID used to correlate messages with each other

message (optional)
String

secondary_message - Up

line_start
Integer

line_end (optional)
Integer

filepath
String

message_text
String

stats_checker - Up

checker_id (optional)
String

checker_name (optional)
String

condition_ids (optional)
array[String]

taxonomy - Up

taxonomy_id (optional)
String

taxonomy_name (optional)
String

Name of the taxonomy, i.e., CERT or CWE

conditions (optional)
array[condition]

taxonomy_version (optional)
String

taxonomy_fields (optional)
Object

Fields associated with this taxonomy, i.e., likelihood

taxonomy_list - Up

taxonomy_id

String

taxonomy_name
String

taxonomy_version
String

taxonomy_response - Up

request_id (optional)
String

taxonomy_list (optional)
array[taxonomy_list]

test_suite_data - Up

test_suite_name
String

test_suite_version
String

metadata_files (optional)
array[byte[]]

format: binary

sourcefile (optional)
byte[]

format: binary

sourcefile_url (optional)
String

format: uri

sourcefunction (optional)
byte[]

format: binary

sourcefunction_url (optional)
String

format: uri

manifest_file (optional)

byte[]

format: binary

manifest_url (optional)
String

Link to download the associated manifest; format: uri

use_license_file (optional)
byte[]

format: binary

author (optional)
String

test_suite_list - Up

test_suite_id
String

test_suite_name
String

test_suite_version
String

test_suite_response - Up

request_id (optional)
String

test_suite_list (optional)
array[test_suite_list]

test_suite_upload_response - Up

test_suite_id
String

test_suite_name
String

test_suite_version
String

request_id
String

status_message (optional)

String

tool - Up

tool_id
String

tool_name
String

tool_version
String

category (optional)
String

checker_data (optional)
array[checker]

code_metrics_data (optional)
Object

tool_data - Up

tool_name
String

tool_version
String

category
String

checker_data (optional)
array[checker]

code_metrics_data (optional)
Object

tool_list - Up

tool_id
String

tool_name
String

tool_version
String

tool_response - Up

request_id
String

tool_list
array[tool_list]

tool_upload_response - Up

request_id
String

tool_id
String

tool_name (optional)
String

tool_version (optional)
String

tools_taxonomies_present - Up

tool_ids (optional)
array[String]

List of tools already present at the source module (stats). The destination module (DataHub) will use this list to avoid sending duplicate tool information.

taxonomy_ids (optional)
array[String]

List of taxonomies already present at the source module (stats). The destination module (DataHub) will use this list to avoid sending duplicate taxonomy information.

update_priority_data - Up

update_project
Boolean

project_ids (optional)
array[String]

priority_scheme_name
String

formula (optional)
String

weighted_columns (optional)
Object

upload_alerts - Up

array[upload_alerts_inner]

upload_alerts_inner - Up

alert_id (optional)
String

tool_id (optional)
String

checker_id (optional)
String

primary_message (optional)
message

more_messages (optional)
array[secondary_message]

upload_alerts_response - Up

request_id (optional)
String

ID used to correlate messages with each other

alerts (optional)
array[upload_alerts_response_alerts]

upload_alerts_response_alerts - Up

alert_id (optional)
String

primary_message (optional)
message

checker_id (optional)
String

message (optional)
String

upload_meta_alerts - Up

array[upload_meta_alerts_inner]

upload_meta_alerts_inner - Up

meta_alert_id (optional)
String

alert_ids (optional)
array[String]

condition_id (optional)
String

determination (optional)
determination

upload_meta_alerts_response - Up

request_id (optional)
String

ID used to correlate messages with each other

meta_alerts (optional)
array[upload_meta_alerts_response_meta_alerts]

upload_meta_alerts_response_meta_alerts - Up

alert_ids (optional)
array[String]

condition_id (optional)
String

meta_alert_id (optional)
String

message (optional)
String

user_information - Up

first_name (optional)
String

last_name (optional)
String

organization_name
String

username
String

password
String

References

1. Bright Silence. [n. d.]. ccsm repository. *GitHub Website*. May 23, 2019 [accessed]. github.com/bright-tools/ccsm

2. Flynn, Lori; and McNeil, Ebonie. "SCAIFE API Beta Version 0.0.2 YAML Specification", June 12, 2019. https://github.com/cmu-sei/SCAIFE-API

3. Flynn, Lori; McNeil, Ebonie; Svoboda, David; Leung, Derek; Kurtz, Zachary; & Lee, Jiyeon. *Integration of Automated Static Analysis Alert Classification and Prioritization with Auditing Tools: Special Focus on SCALe*. CMU/SEI-2019-TR-007. Software Engineering Institute, Carnegie Mellon University. May 2019. resources.sei.cmu.edu/library/asset-view.cfm?assetid=546157

4. Flynn, Lori; Kurtz, Zachary; & Snavely, William. Static Analysis Alert Test Suites as a Source of Training Data for Alert Classifiers [blog post]. *SEI Insights.* April 2018. insights.sei.cmu.edu/sei_blog/2018/04/static-analysis-alert-test-suites-as-a-source-of-training-data-for-alert-classifiers.html

5. Flynn, Lori. Improve Your Static Analysis Audits Using CERT SCALe's New Features. *SEI Webinar Series*. December 2018. resources.sei.cmu.edu/library/asset-view.cfm?assetID=538843

6. Flynn, Lori. Improve Your Static Analysis Audits Using CERT SCALe's New Features [SEI presentation]. Software Engineering Institute, Carnegie Mellon University. November 2018. resources.sei.cmu.edu/library/asset-view.cfm?assetID=532194

7. Flynn, Lori and McNeil, Ebonie. SCALe v. 3: Automated Classification and Advanced Prioritization of Static Analysis Alerts [blog post]. *SEI Insights*. December 2018. insights.sei.cmu.edu/sei_blog/2018/12/scale-v-3-automated-classification-and-advanced-prioritization-of-static-analysis-alerts.html

8. Flynn, Lori. SCALe: A Tool for Managing Output from Static Analysis Tools [blog post]. *SEI Insights*. September 2018. insights.sei.cmu.edu/sei_blog/2018/09/scale-a-tool-for-managing-output-from-static-code-analyzers.html

9. Heckman, Sarah, and Laurie Williams. A systematic literature review of actionable alert identification techniques for automated static code analysis. *Information and Software Technology*. April 2011.

10. OpenAPI Initiative. Swagger OpenAPI Specification. *GitHub*. May 23, 2019 [accessed]. github.com/OAI/OpenAPI-Specification/blob/master/versions/2.0.md

11. Continuous Assurance. Software Assurance Marketplace (SWAMP). *Continuous Assurance Website*. May 23, 2019 [accessed]. continuousassurance.org/about-us/faqs/

12. Swagger pinned repositories. *GitHub Website.* May 23, 2019 [accessed]. github.com/swagger-api

13. Combat Capabilities Development Command (CCDC) C5ISR Center. United States Army Communications-Electronics Research, Development and Engineering Center. *C5ISR Website*. May 23, 2019 [accessed]. www.cerdec.army.mil/
14. Wikipedia contributors, "YAML," *Wikipedia, The Free Encyclopedia,* https://en.wikipedia.org/w/index.php?title=YAML&oldid=899632804 (accessed June 12, 2019).
15. Yin, Terry. [n. d.]. lizard Repository. *GitHub Website*. May 23, 2019 [accessed]. github.com/terryyin/lizard

Contact Us

Software Engineering Institute
4500 Fifth Avenue, Pittsburgh, PA 15213-2612

Phone: 412/268.5800 | 888.201.4479
Web: www.sei.cmu.edu
Email: info@sei.cmu.edu

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.

DM19-0530