Pylonic Documentation

Release

Patrick Pierson

May 10, 2018

Contents

PyIonic is a Python library to interact with Ion Channel's API. With PyIonic it should be very easy for a user to interact with Ion Channel and get their data.

Quick Start

Install PyIonic:

```
pip install pyionic
```

Set the IONCHANNEL_SECRET_KEY:

```
export IONCHANNEL_SECRET_KEY=####IONCHANNEL_SECRET_KEY####
```

Examples

Sample code to report the name, id, and source of all of the projects in a team.

```python
users = core.Users()
team_id = list(users.get_self()['data']['teams'].keys())[0]
projects = core.Projects()
print('--------------------------------')
for project in projects.get_projects(team_id)['data']:
    print('Name: %s' % project['name'])
    print('ID: %s' % project['id'])
    print('Source: %s' % project['source'])
    print('--------------------------------')
```

Sample code to get an analysis for all projects in a team. Reports the name, id, source and if the project passed the last analysis.

```python
users = core.Users()
team_id = list(users.get_self()['data']['teams'].keys())[0]
projects = core.Projects()
analysis = core.Analysis()
print('--------------------------------')
for project in projects.get_projects(team_id)['data']:
    print('Name: %s' % project['name'])
    print('ID: %s' % project['id'])
    print('Source: %s' % project['source'])
    analysis_id = analysis.get_analysis_summery(
        team_id=team_id,
        project_id=project['id']
    )['data']['id']
    if analysis.get_analysis(
        team_id=team_id,
        project_id=project['id'],
        analysis_id=analysis_id
    )['data']['status'] == 'finished':
        print('Scan is good!')
    print('--------------------------------')
```

Sample code to count all of the vulnerabilities for Python 3.4

```python
vuln = core.Vulnerability()
vulnerabilities = vuln.get_vulnerabilities('python', '3.4')
print('%s total vulnerabilities found.' % vulnerabilities['meta']['total_count'])
```

Tests

To setup tests you must first export a valid token for the pyionic test team:

```
export IONCHANNEL_SECRET_KEY=####IONCHANNEL_SECRET_KEY####
```

Then run:

```
pipenv run python setup.py test
```

Documentation

4.1 Core Services

The Core Services are a group of services that Ion Channel provides to give users the ability to conduct risk analysis against their software supply chain.

4.2 Analysis Service

Analysis data is important to users because it contains information about scans ran against their projects. This information includes vulnerability and dependency information as well as virus checks and community data.

class pyionic.core.**Analysis**

The Analysis class interacts with endpoints to return Analysis data. Analysis data includes a scan against the project specified and the data returned is json formatted.

get_analysis(*team_id*, *project_id*, *analysis_id*)

Gets an analysis against a team and project.

Parameters

- **team_id** (*str*) – The team id for the team a user wants to lookup

- **project_id** (*str*) – The project id for the project in a team a user wants to lookup

- **analysis_id** (*str*) – The analysis id for the analysis a user wants to lookup

Returns A requests.get(url).json() instance

get_analysis_summery(*team_id*, *project_id*)

Gets a summary of analysis against a team and project.

Parameters

- **team_id** (*str*) – The team id for the team a user wants to lookup

- **project_id** (*str*) – The project id for the project in a team a user wants to lookup

Returns A `requests.get(url).json()` instance

4.3 Projects Service

Project data is important to users because it contains information about projects they have previously specified.

class `pyionic.core.`**`Projects`**

The Projects class interacts with endpoints to return Project data. Project data includes the project information specified and the data returned is json formated.

`add_alias`(*team_id*, *project_id*, *name*, *org=''*, *version=''*)

Adds alias to project.

Parameters

- **`team_id`** (*str*) – The team id for the team a user wants to add an alias to
- **`project_id`** (*str*) – The project id for the project in a team a user wants to add an alias to
- **`name`** (*str*) – The name of the alias to be added
- **`org`** (*str*) – The org of the alias to be added
- **`version`** (*str*) – The version of the alias to be added

Returns A `requests.get(url).json()` instance

`get_project`(*team_id*, *project_id*)

Gets project for a team.

Parameters

- **`team_id`** (*str*) – The team id for the team a user wants to lookup
- **`project_id`** (*str*) – The project id for the project in a team a user wants to lookup

Returns A `requests.get(url).json()` instance

`get_projects`(*team_id*, *limit=10*)

Gets projects for a team.

Parameters

- **`team_id`** (*str*) – The team id for the team a user wants to lookup
- **`limit`** (*str*) – Limits the number of results returned to the requested size

Returns A `requests.get(url).json()` instance

4.4 Reports Service

Reports data is important to users because it contains information about projects and analysis they have previously specified.

class `pyionic.core.`**`Reports`**

The Reports class interacts with endpoints to return Report data. Reports data includes analysis, team, project and other data to generate Ion Channel reports.

get_analysis_report(*team_id*, *project_id*, *analysis_id*)
    Get Analysis Report takes an analysis_id, team_id, and project_id. It returns the corresponding analysis report

> **Parameters**
>
> > - **team_id** (*str*) – The team id for the team a user wants to lookup
> >
> > - **project_id** (*str*) – The project id for the project in a team a user wants to lookup
> >
> > - **analysis_id** (*str*) – The analysis id for the analysis in a project a user wants to lookup
>
> **Returns** A `requests.get(url).json()` instance

get_raw_analysis_report(*team_id*, *project_id*, *limit=5*)
    Get Raw Analysis Report takes an analysis_id, team_id, and project_id. It returns the corresponding analysis report json.

> **Parameters**
>
> > - **team_id** (*str*) – The team id for the team a user wants to lookup
> >
> > - **project_id** (*str*) – The project id for the project in a team a user wants to lookup
> >
> > - **limit** (*str*) – Limits the number of results returned to the requested size
>
> **Returns** A `requests.get(url).json()` instance

4.5 Rulesets Service

Rulesets are important to users because it sets up an analysis to pass/fail based on conditions.

class pyionic.core.**Rulesets**
    The Rulesets class interacts with endpoints to return Ruleset data. Ruleset data includes the ruleset information specified and the data returned is json formated.

get_applied_ruleset_for_project(*team_id*, *project_id*)
    Gets applied ruleset for a project in a team.

> **Parameters**
>
> > - **team_id** (*str*) – The team id for the team a user wants to lookup
> >
> > - **project_id** (*str*) – The project id for the project in a team a user wants to lookup
>
> **Returns** A `requests.get(url).json()` instance

get_ruleset(*team_id*, *ruleset_id*)
    Gets ruleset for a team.

> **Parameters**
>
> > - **team_id** (*str*) – The team id for the team a user wants to lookup
> >
> > - **ruleset_id** (*str*) – The ruleset id for the ruleset in a team a user wants to lookup
>
> **Returns** A `requests.get(url).json()` instance

get_rulesets(*team_id*, *limit=10*)
    Gets rulesets for a team.

> **Parameters**
>
> > - **team_id** (*str*) – The team id for the team a user wants to lookup

- **limit** (*str*) – Limits the number of results returned to the requested size

    Returns A `requests.get(url).json()` instance

4.6 Scanner Service

Scans are important to users because it sets up an analysis to run and reports the status of the analysis.

class pyionic.core.**Scanner**
  The Scanner class interacts with endpoints to return Scanner data as well as conduct scans. The data returned is json formated.

  analyze_project (*team_id*, *project_id*)
    Starts an analysis against the given project and team.

      Parameters

      - **team_id** (*str*) – The team id for the team a user wants to analyze

      - **project_id** (*str*) – The project id for the project in a team a user wants to analyze

      Returns A `requests.post(url).json()` instance

  get_analysis_status (*team_id*, *project_id*, *analysis_id*)
    Gets an analysis status for the specified analysis

      Parameters

      - **team_id** (*str*) – The team id for the team a user wants to lookup

      - **project_id** (*str*) – The project id for the project in a team a user wants to lookup

      - **analysis_id** (*str*) – The analysis id for the analysis in a project a user wants to lookup

      Returns A `requests.get(url).json()` instance

4.7 Teams Service

Team information is important to users because it allows them to get information about their team.

class pyionic.core.**Teams**
  The Teams class interacts with endpoints to return Teams data. The data returned is json formated.

  get_team (*team_id*)
    Gets a teams info.

      Parameters team_id (*str*) – The team id for the team a user wants to lookup

      Returns A `requests.get(url).json()` instance

4.8 Users Service

User information is important to users because it allows them to get information about themselves.

class pyionic.core.**Users**
  The Users class interacts with endpoints to return Users data. The data returned is json formated.

get_self()
>   Gets self info.

>   >   **Returns** A `requests.get(url).json()` instance

4.9 Vulnerability Service

Vulnerability data is important to users because it gives them information about CVEs and CPEs that may effect their projects.

class `pyionic.core.`**Vulnerability**
>   The Vulnerability class interacts with endpoints to return Vulnerability data. The data returned is json formated.

>   **get_products**(*external_id*)
>   >   Gets data on specified product.

>   >   >   **Parameters** **external_id** (`str`) – The external id for the cpe a user wants to lookup

>   >   >   **Returns** A `requests.get(url).json()` instance

>   **get_vulnerabilities**(*product*, *version=''*, *limit=10*)
>   >   Gets data on specified product and version if specified.

>   >   >   **Parameters**

>   >   >   >   • **product** (`str`) – The product name for the user wants to lookup
>   >   >   >   • **version** (`str`) – The version number the user wants to lookup
>   >   >   >   • **limit** (`str`) – Limits the number of results returned to the requested size

>   >   >   **Returns** A `requests.get(url).json()` instance

>   **get_vulnerability**(*external_id*)
>   >   Gets data on specified CVE.

>   >   >   **Parameters** **external_id** (`str`) – The external id for the cve a user wants to lookup

>   >   >   **Returns** A `requests.get(url).json()` instance

4.10 Indices and tables

• genindex

• search

Indices and tables

- genindex
- search

Index