
Echo VR API Documentation

Release 0.2.0

Ajedi32

Mar 23, 2019

Contents:

1	Echo VR API Reference	3
1.1	Installation	3
1.2	Basic Usage	3
1.3	Reference	4
2	README	5
3	Echo VR API	7
3.1	Installation	7
3.2	Usage	7
3.3	Contributing	8
4	Changelog	11
4.1	Unreleased	11
4.2	0.2.0 - 2018-11-18	11
4.3	0.1.1 - 2018-11-06	12
4.4	0.1.0 - 2018-11-05	12
4.5	0.0.1 - 2018-11-01	12
5	Indices and tables	13

Python bindings for Echo VR's HTTP API.

Echo VR API Reference

The primary entry point of the API is through the `fetch_state()` method. This method returns a `GameState` object which you can then traverse to find any information you could want.

If for some reason you need more direct control over how the API is accessed, you may instead decide to use the `API` class directly.

1.1 Installation

If you haven't already, first [install Python 3 and Pipenv](#).

Now, in your project directory, run:

```
pipenv install echovr-api
```

1.2 Basic Usage

Example:

```
import echovr_api

try:
    game_state = echovr_api.fetch_state()

    print(f"Game status: {game_state.game_status}")
    print(f"Seconds on clock: {game_state.game_clock}")
    print(f"Score: {game_state.blue_team.score} - {game_state.orange_team.score}")

    # See `GameState` reference for available properties/methods
except ConnectionError as e:
    # Echo VR is not running, or you didn't pass the -http parameter when
```

(continues on next page)

(continued from previous page)

```
# starting it.  
except json.decoder.JSONDecodeError as e:  
    # Echo VR is currently not in an Arena match
```

1.3 Reference

For a complete listing of available modules, classes, and methods, see [modindex](#).

You can also view comprehensive documentation of the raw HTTP API itself at the [Unofficial Echo VR API Documentation](#).

CHAPTER 2

README

Python bindings for Echo VR's HTTP API.

3.1 Installation

If you haven't already, [install Python 3](#) and [Pipenv](#).

Now, in your project directory, run:

```
pipenv install echovr-api
```

3.2 Usage

Basic usage example:

```
from requests.exceptions import ConnectionError
import json
import echovr_api

try:
    game_state = echovr_api.fetch_state()

    print(f"Game status: {game_state.game_status}")
    print(f"Seconds on clock: {game_state.game_clock}")

    if (game_state.blue_team.score > game_state.orange_team.score):
        print("Blue team is winning!")
    elif (game_state.orange_team.score > game_state.blue_team.score):
        print("Orange team is winning!")
    else:
        print("It's tied!")
```

(continues on next page)

(continued from previous page)

```
    print(f"Score: {game_state.blue_team.score} - {game_state.orange_team.score}")
except ConnectionError as e:
    print("Connection refused. Make sure you're running Echo VR with the -http option,
→and that you're in a match.")
except json.decoder.JSONDecodeError as e:
    print("Could not decode response. (Not valid JSON.)")
```

For comprehensive documentation of the available methods and classes, please see the [full API Documentation on Read The Docs](#).

3.3 Contributing

To get everything you need to start making changes to this package, first [install Python 3 and Pipenv](#), clone this repository, then run:

```
pipenv install --dev
```

3.3.1 Run the tests

To run all automated tests, run:

```
pipevn run pytest tests/
```

3.3.2 Try it

To play around with the API, open an instance of Echo VR with the `-http` flag, then run:

```
pipenv run python -i ./test.py
```

3.3.3 Generate documentation

To generate documentation for the project, run the following in the project root:

```
pipenv run sphinx-build docs/source/ docs/build/
```

You can then access the built documentation by opening the generated `docs/build/index.html` file in a web browser.

3.3.4 Release process

First, update `CHANGELOG.md` and the version number in `setup.py` and `docs/source/conf.py`. Then commit, tag, and push these changes.

Next, build the package:

```
pipenv install --dev
pipenv run python setup.py sdist bdist_wheel
```

Finally, upload the built packages to PyPi. You can do this using twine (pip install twine):

```
twine upload dist/*
```


All notable changes to this project will be documented in this file.

The format is based on [Keep a Changelog](#), and this project adheres to [Semantic Versioning](#).

4.1 Unreleased

4.1.1 Fixed

- `Team.Color.by_name` throwing an error when called with an invalid name (now returns `None` instead)

4.2 0.2.0 - 2018-11-18

4.2.1 Added

- [API Documentation on Read the Docs](#)
- Support for new API response properties in Echo VR Combat release patch
- `players` property on `GameSate` for listing all players in the game
- `find_team` and `find_player` methods on `GameState`
- Alias `Player#name` as `Player#username`
- `Team.Color#by_name`

4.3 0.1.1 - 2018-11-06

4.3.1 Fixed

- Properly declare dependencies in setup.py

4.4 0.1.0 - 2018-11-05

4.4.1 Added

- A changelog
- Shortcut methods for accessing the default API on localhost
- Concept of team colors, with special methods on the GameState and Team objects
- `Team#score` shortcut method for getting a team's score
- More extensive README documentation on how to install and use the package

4.4.2 Changed

- Fix imports to work when `__init__.py` is not executed directly

4.5 0.0.1 - 2018-11-01

4.5.1 Added

- Simple system for fetching JSON from the API
- Simple object model exposing the basic functionality of the API

CHAPTER 5

Indices and tables

- `genindex`
- `modindex`
- `search`