

openIDL - Contributing

Table of Contents

- [Overview](#)
- [Development Flow](#)
- [Pull Request Process](#)
 - [Pull Request Requirements](#)
 - [Definition of Done](#)
 - [Active Maintainers](#)
- [Developing](#)
 - [Developing the UI](#)
 - [Developing the APIs](#)
 - [Developing Common Code](#)
 - [Developing Utility Code](#)
- [Testing](#)
- [Contributing](#)
- [Deploying](#)

Overview

This page provides guidance to those wanting to contribute to the project.

Here is a video on contributing to open source - <https://egghead.io/courses/how-to-contribute-to-an-open-source-project-on-github>

There is also a [readme.md](#) file at the top level that guides common activities.

Development Flow

- Report the issue you are facing or the new feature you would like to introduce and get approval from the maintainers of the project. Alternatively, choose an already reported issue.
- Pull or clone the repository
- Create a new branch
- Make the necessary changes
- Test your changes
 - Write and run tests
 - Look for instructions in the README.MD file in the project where your changes are for additional instructions on testing the code
- Create a Pull Request and assign reviewers
- Once approved, merge the pull request

Pull Request Process

Every pull request must follow the guidelines set here to be approved and considered for merging. Every pull request should be linked to a reported GitHub Issue in the openIDL project. Pull requests have been set up with checks that all have to pass. At least two maintainers should approve the pull request.

Pull Request Requirements

Rule	Required	Comment
At least 2 approvals from maintainers from two different organizations	✓	
The pull request is linked to a GitHub Issue (new feature or a bug)	✓	
Developer Certificate of Origin (DCO)	✓	Always commit your changes with <code>git commit -s</code> (--signoff)
Automated builds pass	✓	
Automated tests pass	✓	
Security scans pass (Snyk)	✓	

Definition of Done

TBD

Active Maintainers

Name	Organization	GitHub ID
Jeff Braswell	openIDL	tahoeblue
Sean Bohan	openIDL	SeanBohan
Ken Sayers	AAIS	kens-aais
Peter Antley	AAIS	antleypk2
Aashish	Chainyard	aashishchainyard
Surya Lanka	Chainyard	suryalanka-cy
Tsvetan Georgiev	Senofi	TsvetanG
Yanko Zhelyazkov	Senofi	yzhivkov



New maintainers are welcome.

Developing

Each subdirectory has a readme.md file that helps with the details of local development.

Developing the UI

Developing the APIs

Developing Common Code

Standard object notation is JSON. Keys/Attributes will be named using PascalCase example: `{'Foo': 'val'},{'FooBar': 'val2'}`

Objects will have singular names. Lists of objects will be names as plural form of singular object. ex: `{"Foos" : [{"Foo":"val3"}, {"Foo":"val4"}]}`

Developing Utility Code

Testing

Contributing

Deploying