opendevelop Documentation

Release 0.2 Alpha

opendevelop_team

February 08, 2014

CHAPTER 1

Overview

OpenDevelop is an open source platform dedicated on facilitating the procedure of running code. It provides a state-of-the-art asynchronous REST API to allow users run their code with a single HTTP request. OpenDevelop has been designed and designated to be used by sourceLair as its sole backend for any code-executing task.

However it is also opensource so that anyone can set up a private opendevelop installation.

Opendevelop is based on the following technologies

- Django
- Docker
- Celery

Caution: Right now OpenDevelop is in Alpha stage and under heavy development. The documentation might not be complete and we do not suggest using it for a production system at the moment.

CHAPTER 2

Contents

2.1 Opendevelop Basic Entities

2.1.1 Sandbox

The core of the opendevelop project is the sandbox. A sandbox is a linux container created with docker. In this container the end user can execute any kind of code in a secure and isolated way.

2.1.2 App

Every Sandbox that is created has to be associated with a specific App. An opendevelop App is a logical entity that many sandboxes can be associated with. Every App has an id and a secret key used for Oauth authentication in every API call.

2.1.3 Opendevelop User

An opendevelop App belongs to a user. Every user can have multiple Apps associated with him.

2.1.4 Image

Image is the logical entity that contains information about the image that runs in the linux containers where used by the Sandboxes. The user can choose and executed their code in images provided by Opendevelop or create their own custom image so that running code with opendevelop feels really like home.

2.1.5 DockerServer

The Opendevelop's architecture is designed in such a way so that it can easily scale up in a destributed way. So a DockerServer is a server where the Sandboxes are created and the code is executed. This way the webserver that serves Opendevelop chooses where to dispatch the incoming requests for sandbox creations.

2.2 Opendevelop API

2.2.1 RESTful API

Opendevelop's end user has at his disposal an asynchronous HTTP REST API to use and easily run code.

Authentication

In order to make API calls the user will need to be authenticated according to OAuth protocol. So every request must include a Basic OAuth HTTP header that includes the id and the secret key of the App.

```
{'Authorization' : "Basic " + base64("<id>:<secret>")}
```

2.2.2 API calls

Sandboxes

Verb	URI	Description
GET	`/api/sandboxes`	List all sandboxes associated with the given App
GET	`/api/sandboxes/sandbox_id`	Show information about a specific App
POST	`/api/sandboxes`	Create a new sandbox and run the given code inside

List Sandboxes

Example request

```
Request Url: http://opendevelop/api/sandboxes
Request Method: GET
Params: {}
```

Example response

```
{
"sandboxes": [
    {
        "status": "terminated",
        "image": "my_image",
        "cmd": "[\"python test.py\"]",
        "return_code": 127,
        "logs": "sh: 0: Can't open start\n"
    },
    {
        "status": "running",
        "image": "my_image",
        "cmd": "[\"ls -a\"]",
        "return_code": null,
        "logs": null
    }
  ]
}
```

Show Sandbox

Example request

```
GET /api/sandboxes/1
```

Example response

```
{
"status": "terminated",
"image": "my_image",
"cmd": "[\"python test.py\"]",
"return_code": 0,
"logs": "hello opendevelop!\n"
}
```

Create Sandbox

Example request

```
Request Url: http://opendevelop/api/sandboxes
Request Method: POST
Files: {
    "0": {
        "webkitRelativePath": "",
        "lastModifiedDate": "2013-12-22T22:27:47.000Z",
        "name": "test.py",
        "type": "text/x-python-script",
        "size": 46
       },
    "length": 1
}
Params: {
    "image": "my_image",
    "cmd": "[\"python test.py\"]"
    "timeout": "10"
}
```

Timeout is an optional parameter that allows the user to specify the maximum time in seconds that the execution of the sandbox will last before it automatically gets killed.

Example response

```
a326efb1fe1f980a
```

Images

Verb	URI	Description
GET	`/api/images`	List all available images to be used for sandbox creation

Example request

```
Request Url: http://opendevelop/api/images
Request Method: GET
Status Code: 200
Params: {}
```

Example response

```
["base"]
```

2.3 Opendevelop Command Line Interface

OpenDevelop comes bundled with a command line interface that helps you administer your OpenDevelop instance. To get more information about every command, you can run

```
./manage.py <command> -h
```

2.3.1 Add a Docker server

In order to add a new Docker server to OpenDevelop, you have to run the following command:

```
./manage.py server-create --name=name_of_your_server --url=docker_api_url --buckets=buckets_directory
```

2.3.2 Add a new user

In order to add a new OpenDevelop user, you have to run the following command (the *–organization* flag is optional and declares the new account as an organization account):

```
./manage.py user-create --username=sourcelair --email=opendevelop@soureclair.com --passwd=123 --orga
```

2.3.3 Add a new app

To create a new OpenDevelop app, you have to run the following command, by supplying the name of your app and the username of the owner

```
./manage.py app-create --app_name=myapp --username=user_that_owns_the_app
```

2.3.4 Add a new image

To add a new OpenDevelop image you have to run the following command and supply a name for your image, as well as a small description for it, its Docker image counterpart and a docker index url

```
./manage.py image-create --name=my_image --description=what_my_image_does --url=docker_index_url --d
```

2.4 Clients

OpenDevelop clients wrap its HTTP REST API and expose it to a native-like API for each language. Available OpenDevelop clients are listed below.

2.4.1 Python client

A Python client is available for OpenDevelop at http://www.github.com/sourcelair/opendevelop-py

Install client The Python client can be installed using pip.

```
pip install opendevelop
```

2.5 Opendevelop Private Installation

You can easily build a private installation of Opendevelop and give your users the chance to run code in the cloud.

Opendevelop is a Django project which dispatches the requests to the docker servers. So an opendevelop installation should have one web server and multiple docker servers.

2.5.1 Install

OpenDevelop is being developed using Ubuntu and until it reaches a more stable state, the documentation will assume you are installing it on an Ubuntu machine, preferably Ubuntu 13.10 or greater.

At first you should clone the Opendevelop public repository.

```
git clone git@github.com:sourceLair/opendevelop.git
```

Before running the installer make sure you have *rabbitmq-server* install. If you do not have it installed you can run

```
sudo apt-get install rabbitmq-server
```

on your terminal to install it.

Next thing to do is run the install script as a root user from the root directory of opendevelop.

```
sudo python install/installer.py
```

Last thing is to create the OpenDevelop models into the database. To do that you have to run the following two commands, from within the manage.py directory.

```
./manage.py syncdb
./manage.py migrate
```

Starting the service

In order to get OpenDevelop up and running you need to start the *Celery* and the *Django server* from the command line, from within the manage.py directory, in two different Bash sessions.

```
./manage.py runserver

./manage.py celeryd
```