

**HealthAPIx**

**Developer Portal Deployment**

**Document**

**Version Control**

Version	Date	Summary of Changes
1.0	18-Dec-2015	Details of HealthAPIx Devportal deployment

## Table of Contents

<b>1.</b>	<b>Introduction.....</b>	<b>4</b>
<b>2.</b>	<b>Assumptions.....</b>	<b>4</b>
<b>3.</b>	<b>Getting Started.....</b>	<b>4</b>
<b>3.1</b>	<b>Flame – Open source project for Health APIx solution. ....</b>	<b>4</b>
<b>3.1.1</b>	<b>Accessing Flame. ....</b>	<b>5</b>
<b>3.2</b>	<b>Apigee developer portal request. ....</b>	<b>5</b>
<b>3.3</b>	<b>Pantheon – create your free account. ....</b>	<b>6</b>
<b>3.4</b>	<b>Flame customized source code migration to out-of-box portal ....</b>	<b>8</b>
<b>3.5</b>	<b>Theming of SmartDocs Pages: .....</b>	<b>13</b>
<b>3.5.1</b>	<b>Applying HBR Template.....</b>	<b>13</b>
<b>3.5.2</b>	<b>Upload Swagger File .....</b>	<b>14</b>
<b>3.5.3</b>	<b>Creating Quicktab instance .....</b>	<b>14</b>

## 1. Introduction

The Apigee HealthAPIx solution provides App (both, mobile and web) developers and digital teams with access to:

A purpose-built developer portal with FHIR-ready APIs, including claims, patient, medication, and condition APIs. Here an app developer can explore documentation and test APIs.

- An open source project with preconfigured API proxies for 16 FHIR APIs.
- Pre-integrated OAuth security framework and other key security functions.
- A reference implementation with a gallery of FHIR-enabled apps for inspiration.

The HealthAPIx solution is deployed on two types of instances.

- Apigee Developer portal - acts as a client to Apigee Edge. The app gallery is also a part of developer portal.
- Apigee Edge which hosts pre-configured FHIR API proxies with OAuth security framework.

The intent of this document is to guide step by step how to deploy the customized Developer Portal instance in the cloud.

While this document steps out the goal to setup a developer portal instance, a preliminary read about developer portal would be useful.

<http://apigee.com/docs/developer-services/content/what-developer-portal>

## 2. Assumptions

This document assumes that Apigee Edge gateway exposing FHIR APIs is already up and running.

The details to setup an Apigee instance with FHIR APIs is covered in [HealthAPIx Design and Deployment v1.2.docx](#) document in \flame\docs folder.

**Prerequisites:** User should have

- Apigee dev-portal admin access,
- pantheon access,
- GitBash 2222 port enabled.

## 3. Getting Started

### 3.1 Flame – Open source project for Health APIx solution.

Flame is an open source project including developer portal source, 16+ FHIR API proxies and the identity and consent management. This document focusses only

on deployment of Developer portal in the cloud. After successful deployment, an app developer should be able to access following:

- Apigee Edge Developer Portal with a customized theme for Healthcare.
- FHIR API Documentation for easy understanding.
- FHIR Sandbox for trying out FHIR APIs
- App Gallery to promote App ecosystem of your API program.

### 3.1.1 Accessing Flame.

Go-to below git hub directory for flame project and clone it to your desktop. Cloned directory contains codes for Dev-portal, FHIR API Proxies and identity management.

Get the API proxy bundle from GIT repository <https://github.com/apigee/flame/tree/master/src/developerportal>.

Use below command:

```
$ git clone <path of git repository>
```

On unavailability of git repo, download code from

<https://drive.google.com/folderview?id=0B-y5CAUxv0zuU2o5ZFNtOUdEMFE&usp=sharing>

**/flame/tree/master/src/developerportal** contains custom portal source code.

In section **3.4. 'Flame customized source code migration to out-of-box portal'**, we will see how to migrate customized dev portal source on a new instance of Apigee out-of-box dev-portal.

### 3.2 Apigee developer portal request.

Firstly, register for Apigee out-of-box Developer Portal on a cloud instance. Use the below URL to register and fill in the form and submit.

<http://apigee.com/docs/content/developer-services-portal-request>

Once your portal is provisioned, you will receive an email from Apigee with further instructions.

See image below.

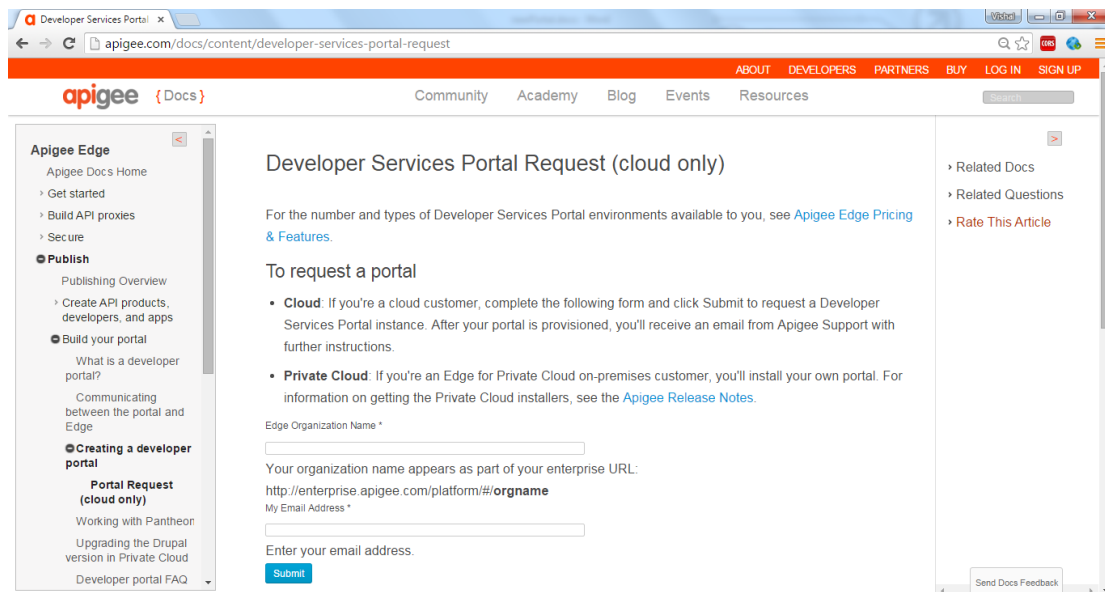


Figure 1.

### 3.3 Pantheon – create your free account.

Pantheon - is a web development platform for open-source Drupal based websites. It is a combination of web development tools in the cloud and web hosting and management services.

URL: <https://pantheon.io>

Register to create a free account on pantheon.

See image below.

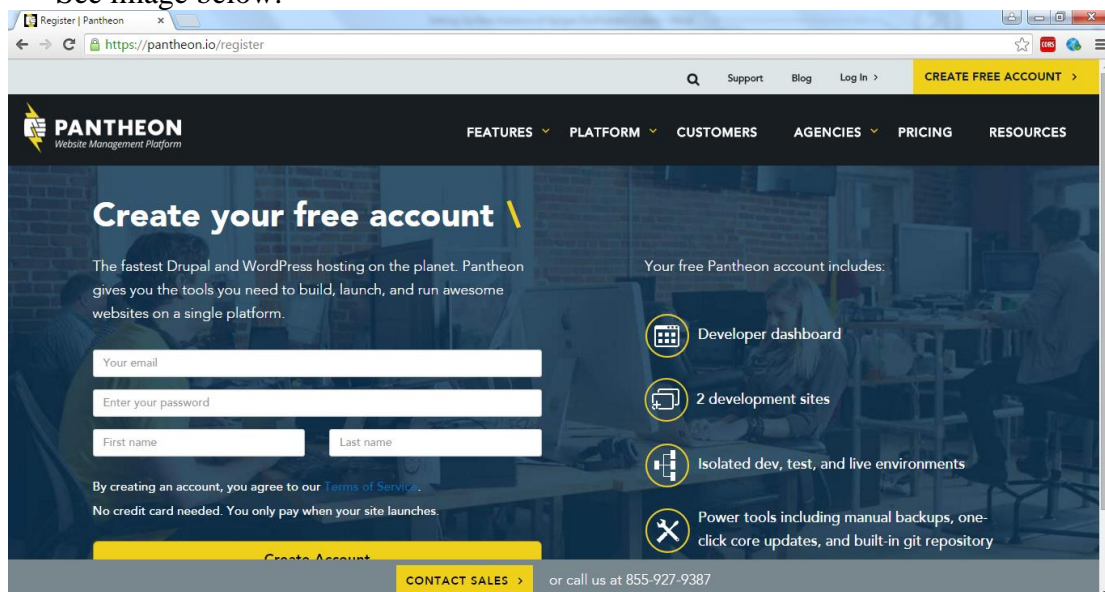


Figure 2.

Apigee out-of-box dev-portal is hosted on pantheon, you just need to connect with Apigee support team to request them to associate your Apigee dev portal with your created pantheon account.

Once it is associated, you will be able to access and interact with your Apigee out-of-box dev-portal code.

See image below –

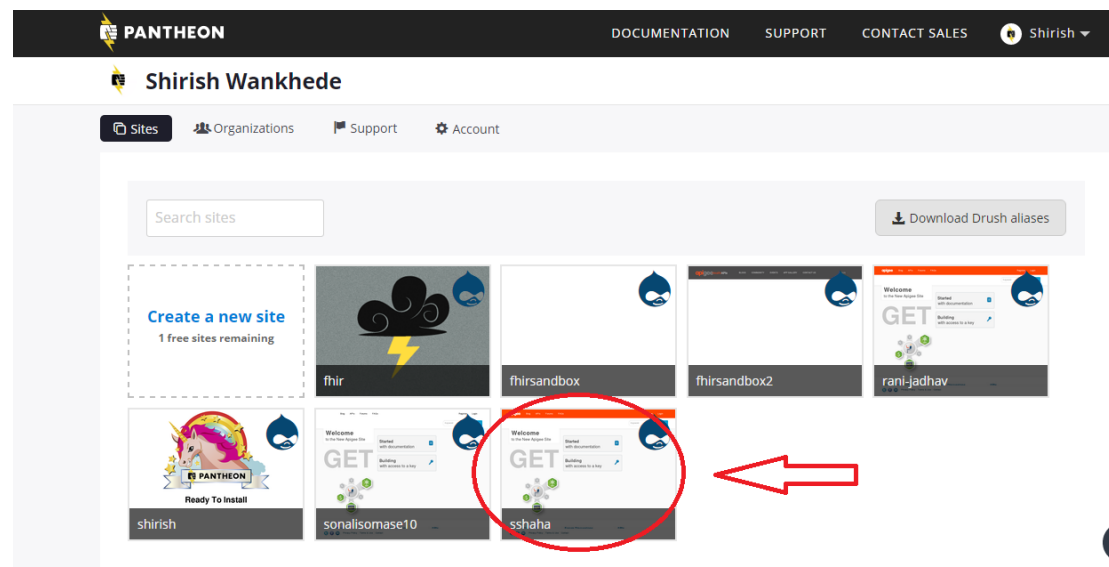


Figure 3.

Above image shows dev-portal (for shirish) now associated with pantheon and portal code can now be accessed using that link.

See image below –

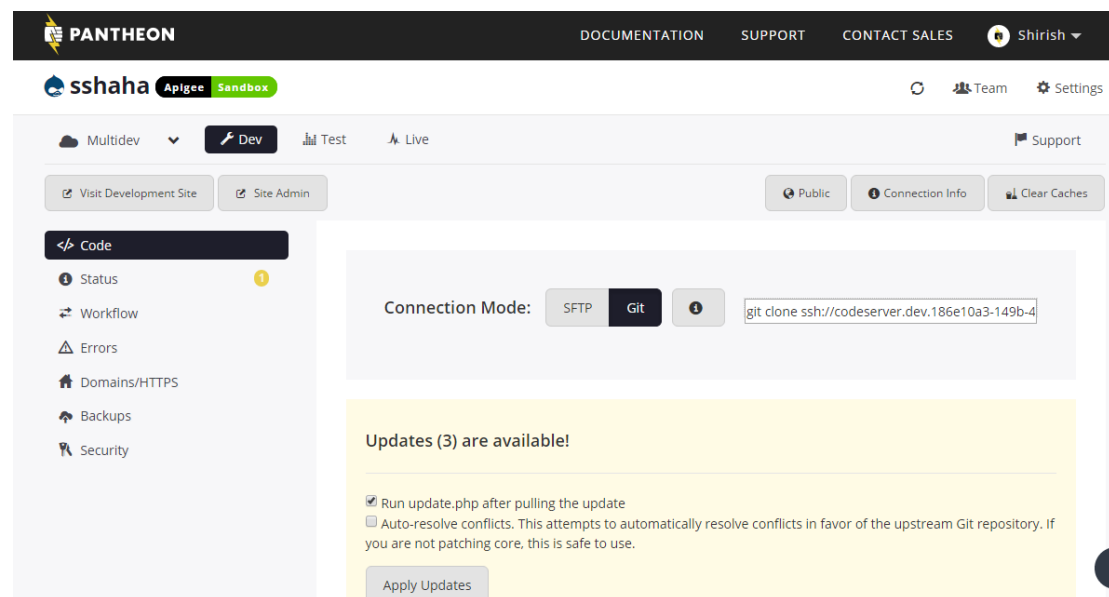


Figure 4.

### 3.4 Flame customized source code migration to out-of-box portal

Initially, before flame code migration to portal, portal UI would look like as below:

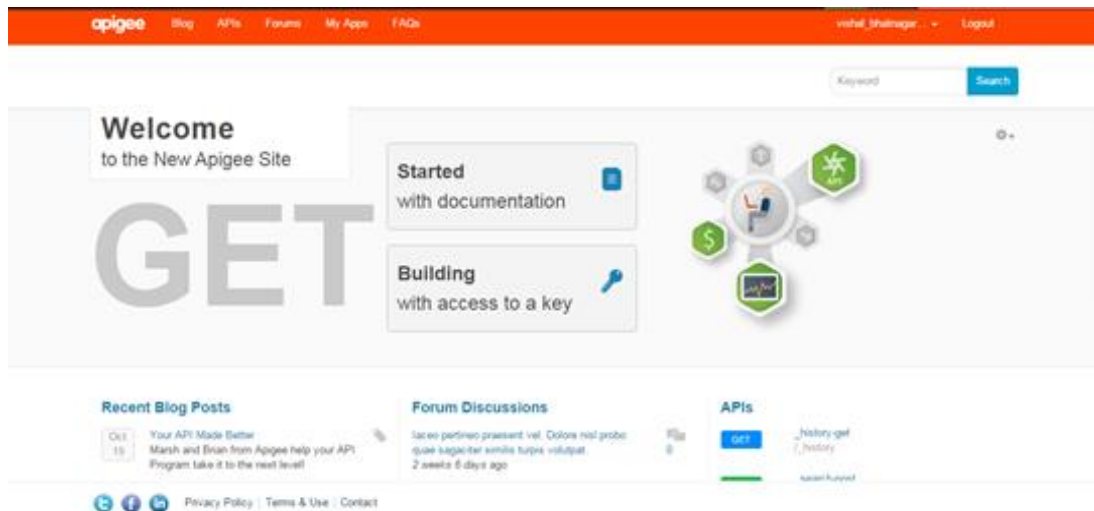


Figure 5.

A Backup of portal's code is recommended before migrating flame code on portal. Login Pantheon -> goto your portal site -> Backups -> and create a new backup. You can select the option to save the backup for a month or for 6 months.

See image below:

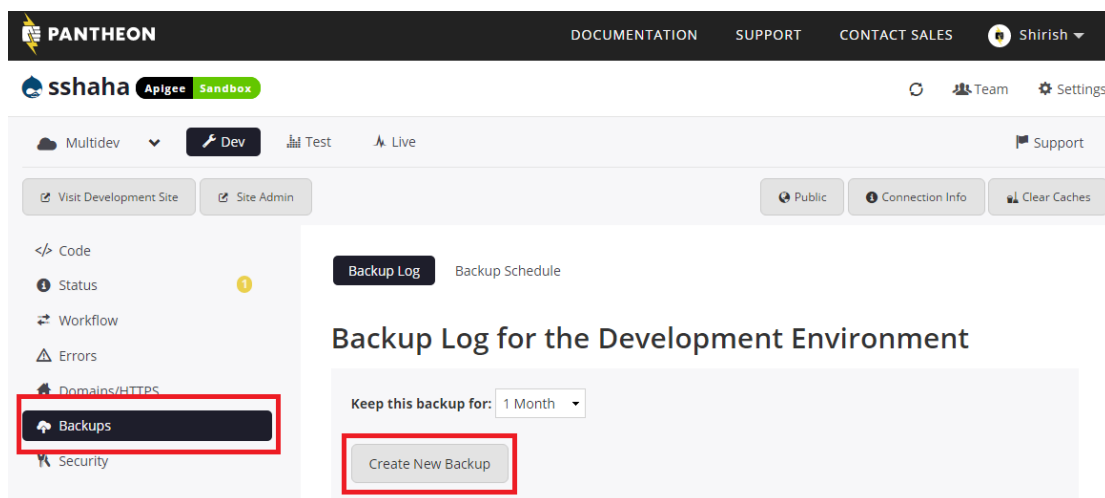


Figure 6.

Next clone default portal's code:  
This document uses git bash client to access Git.  
Get clone url from pantheon.

See image below:



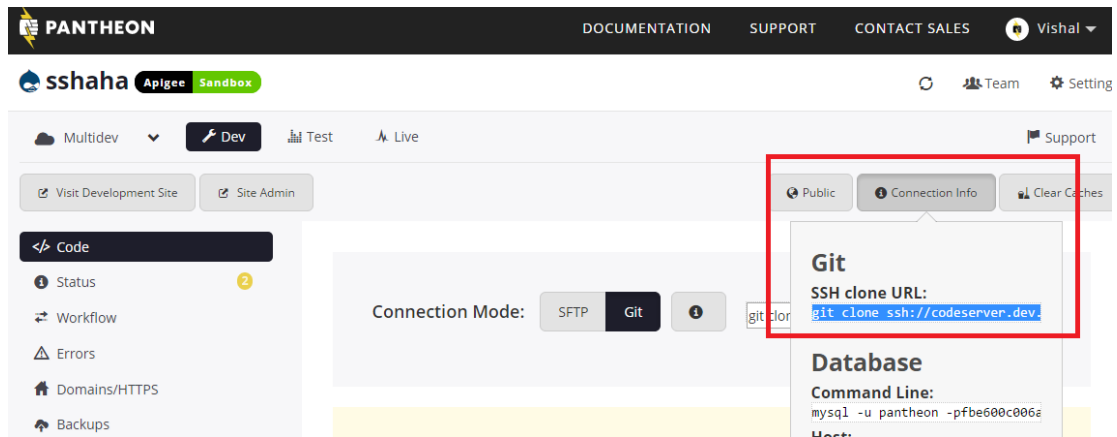


Figure 7.

Now open git bash. Check out the latest code.

1. If the code is not present on your local system, Clone the latest code. Use the following commands.
  - Go to the folder where you want to clone the code.
  - Use the git clone command (highlighted in RED)
  - If it prompts for security validation then type “yes” and proceed.
2. If the code is already present on your local system, then check out updated code using following commands.
  - Go to home folder of the code. (e.g. cd to \$<home\_folder>)
  - \$git pull
  - Enter the password if it's prompted.

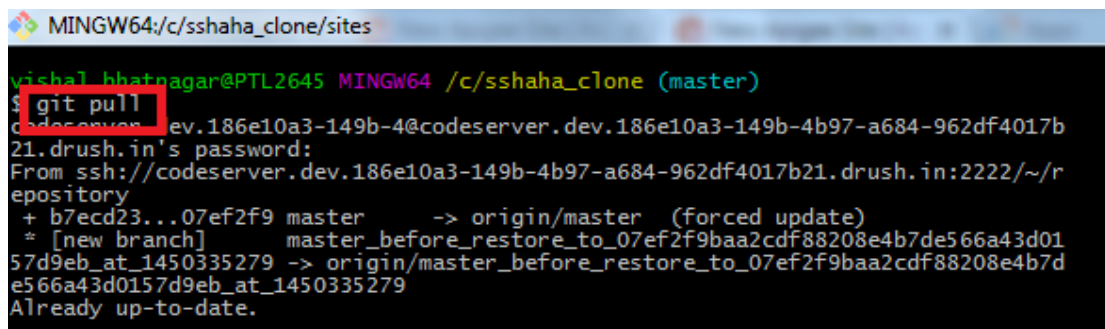


Figure 8.

This command will clone “sshaha” portal’s out-of-box code on your local system in your <home\_folder> directory as <home\_folder> /sites /

The ‘all’ folder from default portal code now has to be replaced with the ‘all’ folder in flame.

Copy `/flame/tree/master/src/developerportal/all/` to  
`/<home_folder>/sites/all/`

Now we need to push back the custom changes.

This will be achieved by running below commands in GitBash

### Commands :

Open GitBash.

```
$ cd {root_folder}/sites
```

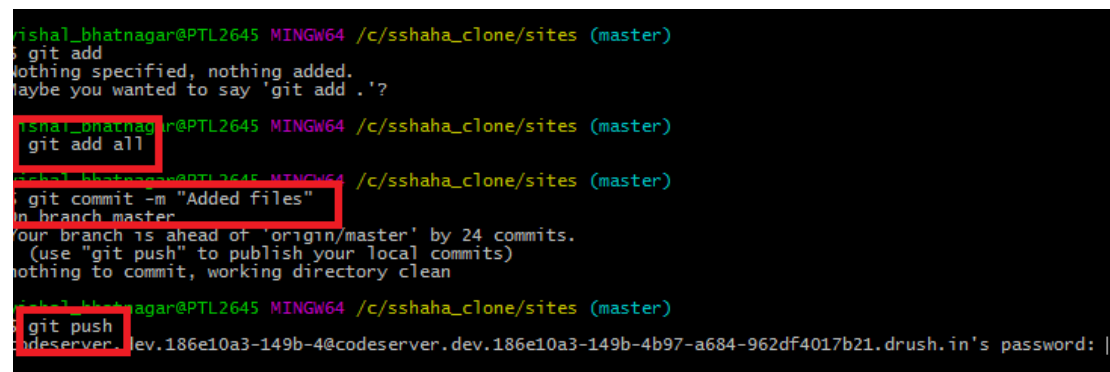
```
$ git add all
```

```
$ git commit -m "—comments--"
```

```
$ git push
```

If password is prompted then type your GitBash password and “Enter” key button.

See image below:



```
ishal_bhatnagar@PTL2645 MINGW64 /c/sshaha_clone/sites (master)
$ git add
nothing specified, nothing added.
maybe you wanted to say 'git add .'
ishal_bhatnagar@PTL2645 MINGW64 /c/sshaha_clone/sites (master)
$ git add all
ishal_bhatnagar@PTL2645 MINGW64 /c/sshaha_clone/sites (master)
$ git commit -m "Added files"
[master 24 commits ahead of origin/master]
nothing to commit, working directory clean
ishal_bhatnagar@PTL2645 MINGW64 /c/sshaha_clone/sites (master)
$ git push
dev.186e10a3-149b-4@codeserver.dev.186e10a3-149b-4b97-a684-962df4017b21.drush.in's password: |
```

Figure 9

Above commands will help in pushing changes to portal.

Open the Dev-Portal on the browser and login with Admin credentials.

Further we need to do few manual configurations.

### 1.Enable FHIR responsive theme:

Click on Appearance -> Enable and set default hyperlink as shown in screenshot.

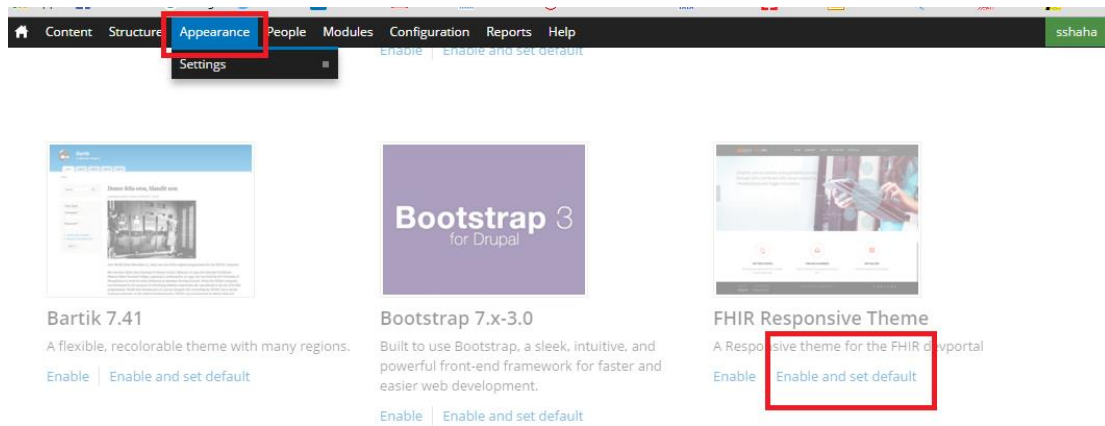


Figure 10

## 2.Enable FHIR required Modules (FHIR Blocks and FHIR Custom):

Go to Module tab -> Search FHIR

You will get list of FHIR Modules. Enable **FHIR Blocks** and **FHIR Custom** Modules and click on “**Save Configuration**” Button.

Click on “**Continue**” button to enable dependent module automatically.

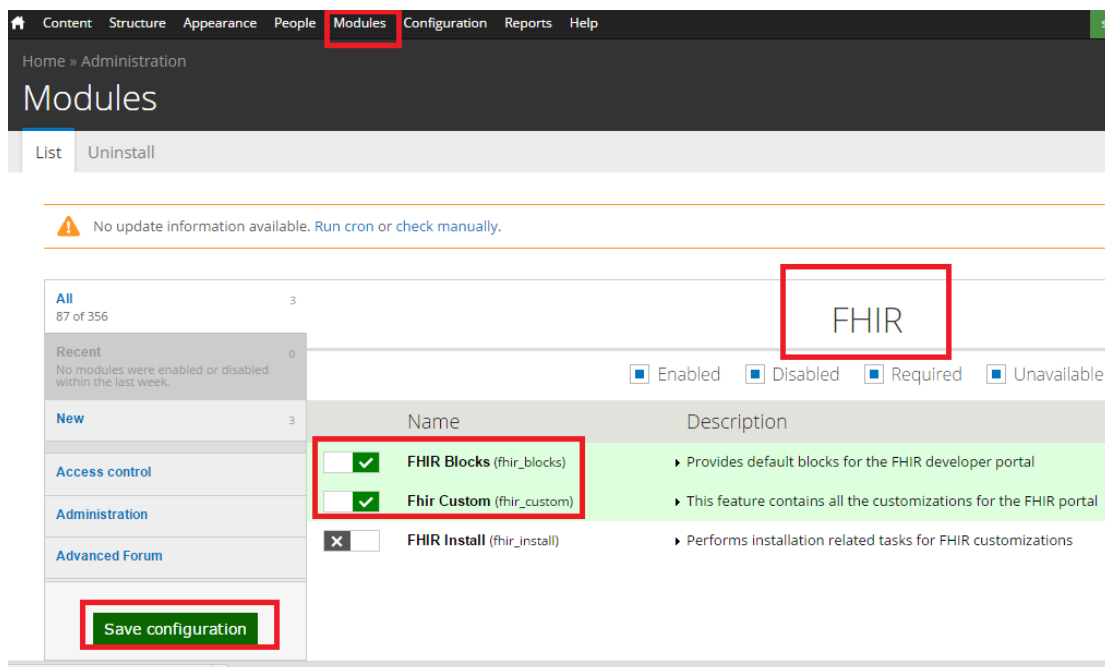


Figure 11

## 3.Clear Cache :

To clear cache Go to **Home** -> **Flush all caches** as shown in below screen shot.

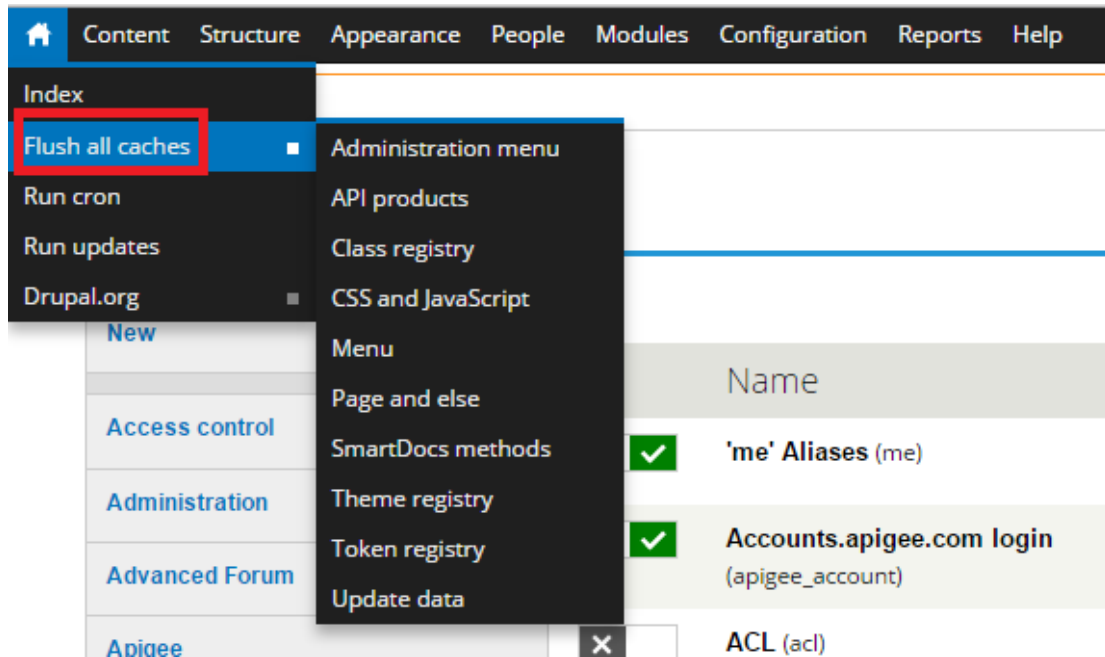


Figure 12

#### 4. Enable FHIR Install Module:

Go to Module tab -> Search FHIR

You will get list of FHIR Modules. Enable **FHIR Install** Module and click on “**Save Configuration**” Button.

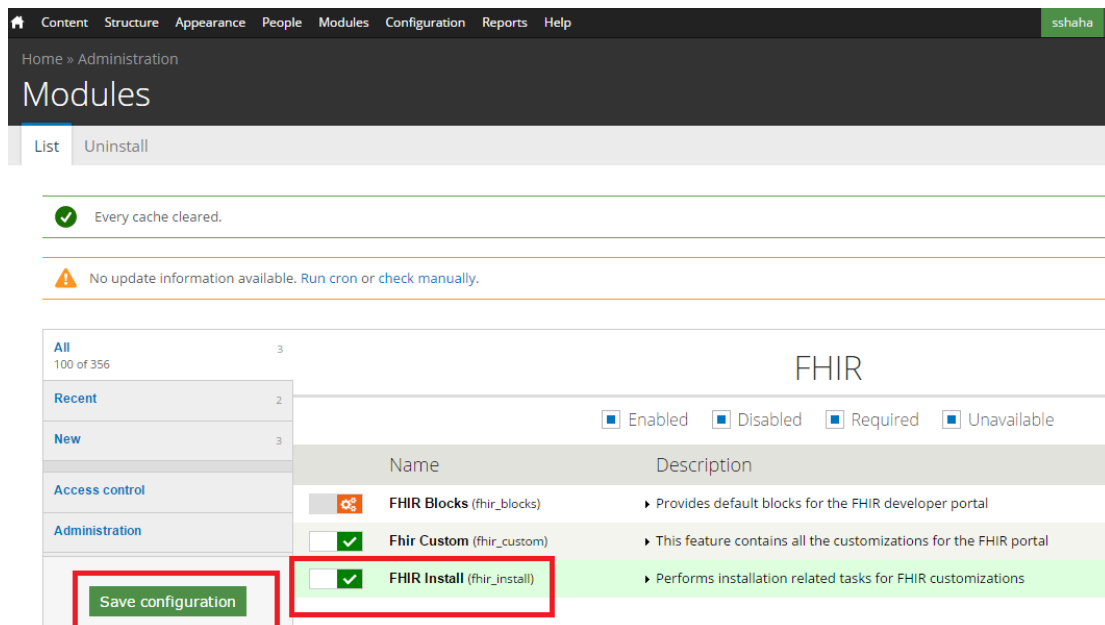


Figure 13

### 5. Clear Cache Again :

To clear cache goto Home -> Flush all caches

Here we go, now refresh your portal site and you will see all change of flame code on your portal.

See image below:

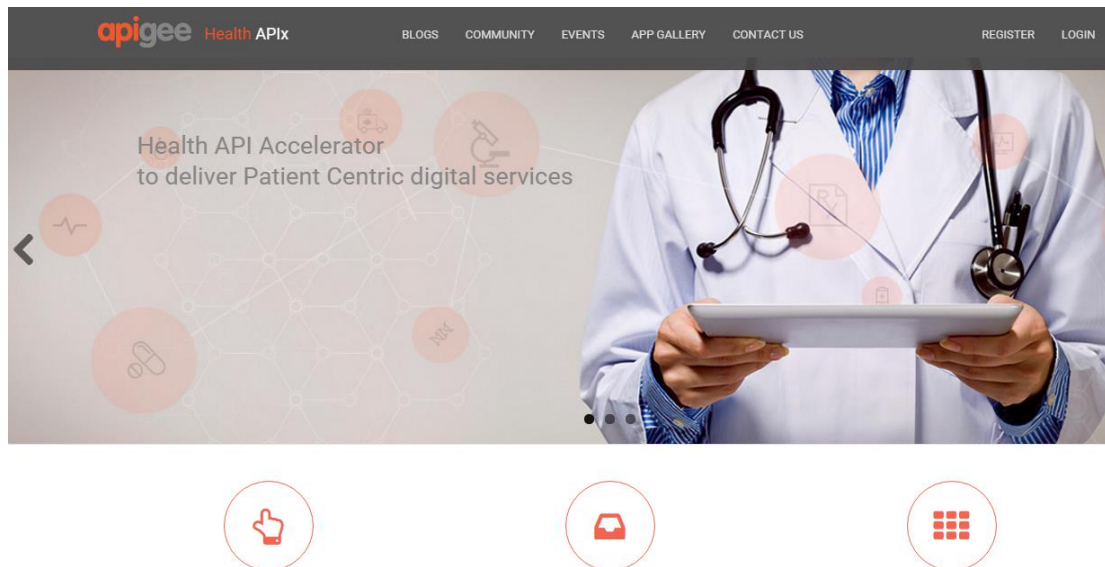


Figure 14

## 3.5 Theming of SmartDocs Pages:

### 3.5.1 Applying HBR Template

- Go to Configuration -> SmartDocs -> Click on “Choose File” button to select the .hbr file from cloned/downloaded flame source code i.e. `flame/src/developerportal/smartdocTemplate.hbr`

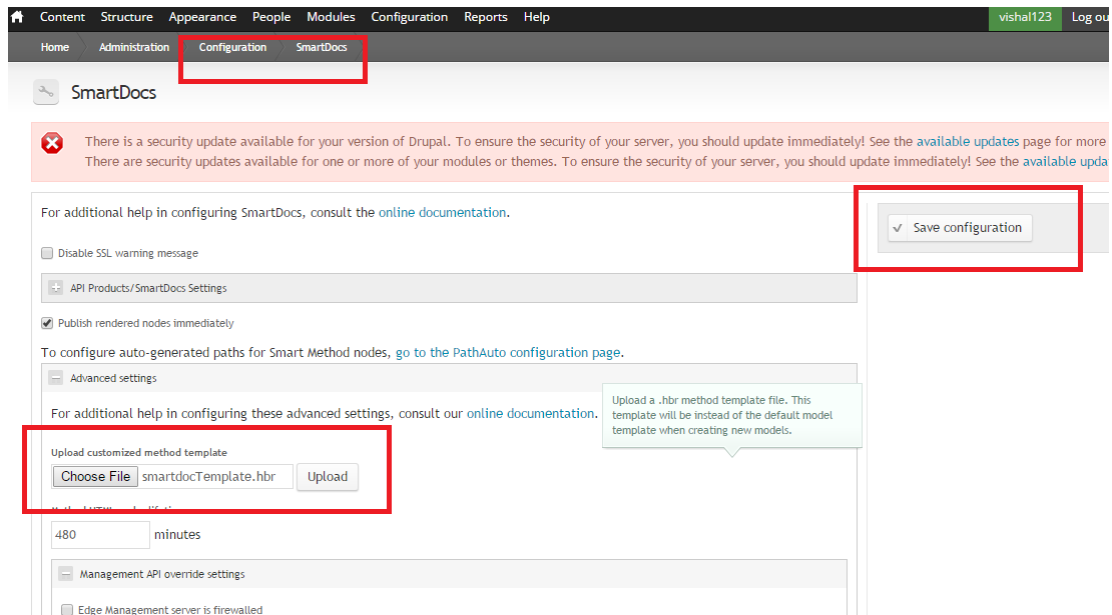


Figure 15

- Click Upload Button
- Save Configuration
- Clear Cache
- Check the changes reflecting on SmartDoc page.

### 3.5.2 Upload Swagger File

- Refer Document “flame/docs/HealthAPIx\_Deploy\_SmartDocs\_v1.0.docx”

### 3.5.3 Creating Quicktab instance

- Go to modules -> Enable Quicktab Module
- Go to Structures -> QuickTabs -> Add QuickTabs Instance
- Fill the details as highlighted in **RED**

The screenshot shows the 'Quicktabs' configuration page in Drupal. The title is 'FHIR APIs'. The machine name is 'fhir\_apis'. The renderer is set to 'accordion'. The 'Collapsible' checkbox is checked. The default tab is set to '- None -'.

There is a security update available for your version of Drupal. To ensure the security of your server, you should update immediately. There are security updates available for one or more of your modules or themes. To ensure the security of your server, you should

Title \*  
FHIR APIs Machine name: fhir\_apis [Edit](#)

Renderer  
accordion

☐ History

JQuery UI options

☐ Autoheight  
☒ Collapsible

Default tab  
- None -

Figure 16

- Add Tabs for all APIs (e.g. Vaccination API) as shown in Fig 17.  
(Ref. entries in **RED** color box)
- Click “Save” Button.

The screenshot shows the 'Quicktabs' configuration page in Drupal, specifically the 'Vaccination' tab configuration. The 'Tab title' is 'Vaccination'. The 'Tab type' is 'block'. The 'Tab content' is 'vaccination\_methods' with the display 'frontpage: Front-page Block'. The 'Save' button is highlighted.

JQuery UI options

☐ Autoheight  
☒ Collapsible

Default tab  
- None -

☐ Hide empty tabs

Tab title	Tab type	Tab content
Vaccination	<input checked="" type="radio"/> block <input type="radio"/> view <input type="radio"/> node <input type="radio"/> qtabs <input type="radio"/> callback	Select a view vaccination_methods display frontpage: Front-page Block
	<input checked="" type="radio"/> block <input type="radio"/> view <input type="radio"/> node <input type="radio"/> qtabs <input type="radio"/> callback	Select a block Apigee SSO (apigee_sso_ui:0) <input checked="" type="checkbox"/> Hide the title of this block

☒ Save

Figure 17

- Configure to display Quicktab block on SmartDoc Page.

- a. Go to Structures -> Blocks -> Search for Created Quicktab name e.g. "FHIR\_API"
- b. Select the region as a "SECONDARY"
- c. Click on "Configure" hyperlink as shown in below screenshot as BLUE

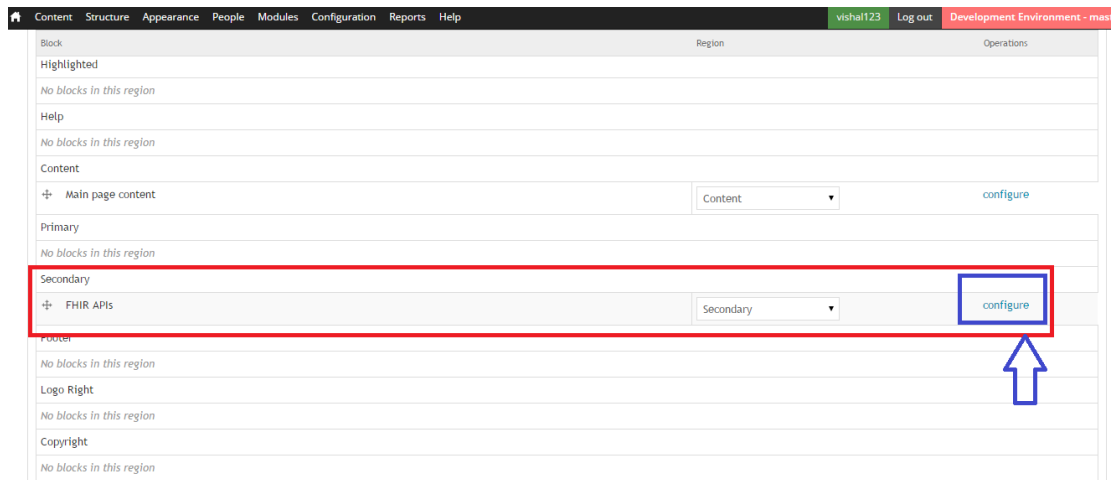


Figure 18

- d. Fill the following details for "FHIR APIs" block as shown in below image (Figure 19 & 20)
  - I. Css : all-patient-left-nav
  - II. FHIR Responsive theme : Secondary
  - III. Content Type : SmartDocs Method
  - IV. Click on "Save" button.



Content Structure Appearance People Modules Configuration Reports Help

Home Administration Structure Blocks Configure block

'FHIR APIs' block

There is a security update available for your version of Drupal. To ensure the security of your site, please update Drupal. There are security updates available for one or more of your modules or themes.

Block title

CSS class(es)

all-patient-left-nav

Region settings

Specify in which themes and regions this block is displayed.

FHIR Responsive Theme (default theme)

Secondary

Rubik (administration theme)

- None -

Figure 19

Content Structure Appearance People Modules Configuration Reports Help

visal12

Region settings

Specify in which themes and regions this block is displayed.

FHIR Responsive Theme (default theme)

Secondary

Rubik (administration theme)

- None -

Apigee Responsive

- None -

Save block

Visibility settings

Pages

Not restricted

Content types

SmartDocs Method

Roles

Not restricted

Users

Not customizable

Show block for specific content types

Apps

Article

Audio

Basic page

FAQ

Forum topic

Image

Video content

Slides

☒ SmartDocs Method

Video

Webform

Figure 20.

- Resultant screen should be as below.

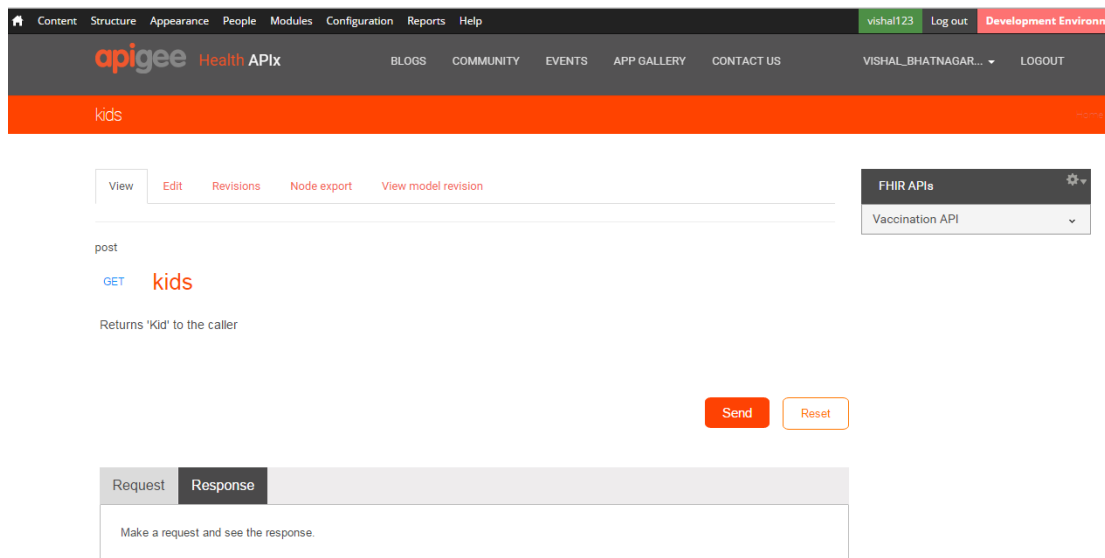


Figure 21

This completes configuration of Smartdocs.

Following all sections and steps above would enable a customized developer portal similar to <https://healthapix.apigee.com/>

Have fun!