# **Practical Exercise: Customizing the Developer Portal**

Training Objective

Learn how to customize the look and feel of the Developer Portal.

Business Scenario

PizzaShack wants to customize the look and feel of the Developer Portal to match their company logos and website.

High-Level Steps

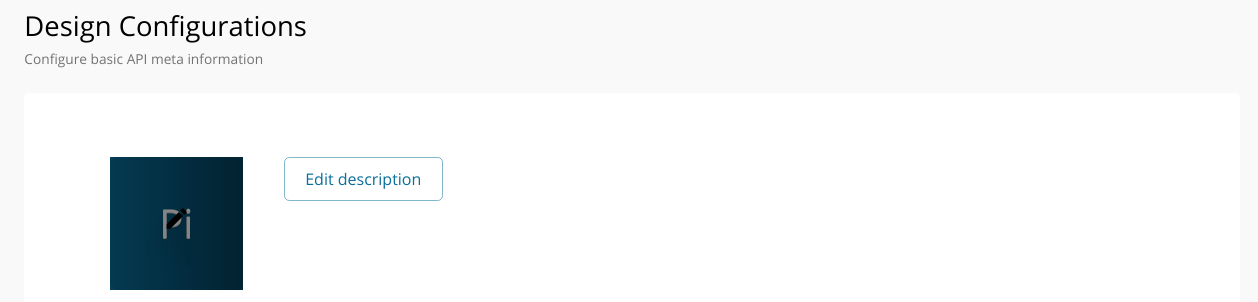
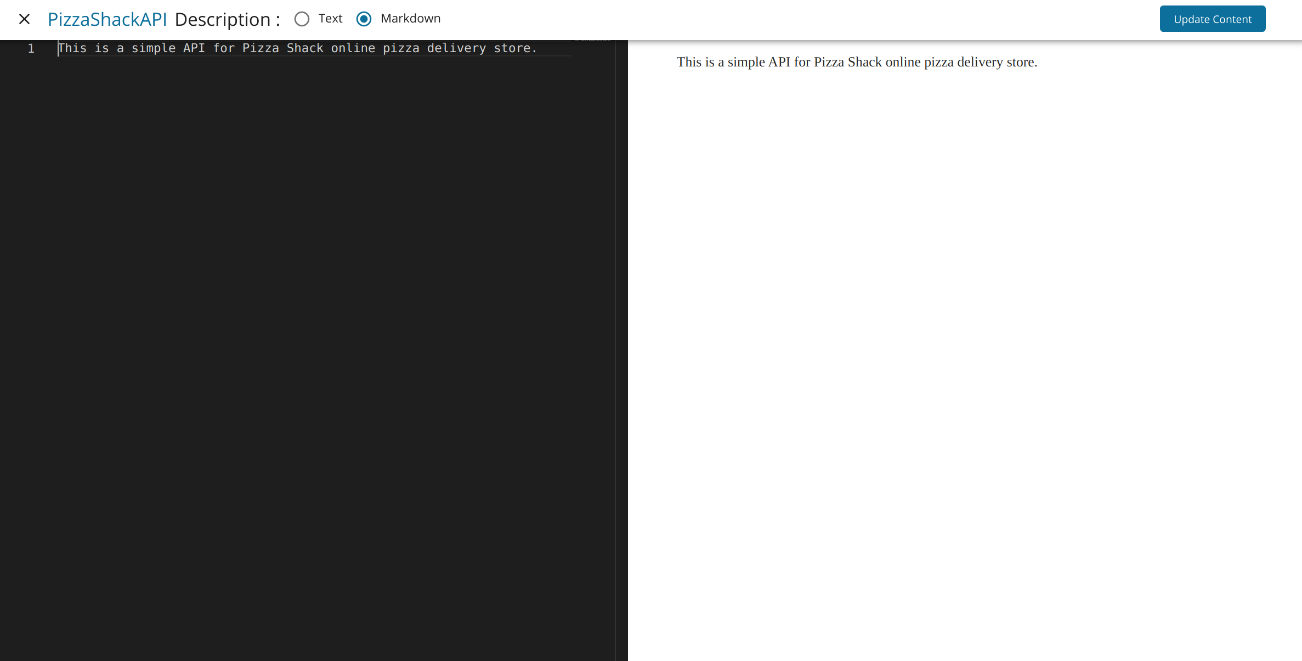
* Customize the Developer Portal by overriding the Developer Portal theme.

Detailed Instructions

Execute the different types of customizations mentioned below to change the look and feel of the Developer Portal, by following the instructions given in the documentation.

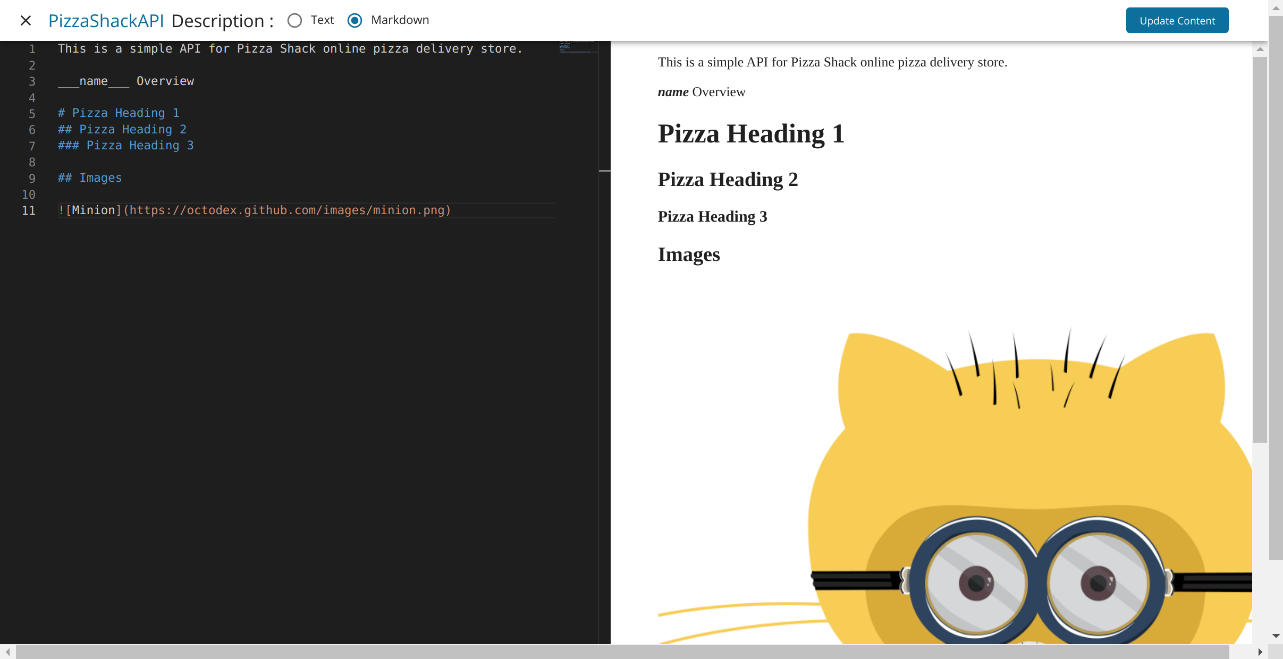
# **Override API Overview page per API**

It is possible to display a custom Overview content for any API by following the steps given below.

1. Login to the API Publisher and go to the Develop → Portal Configurations → Basic Info section of the API that you want to override.
2. Click **Edit description** to open the embedded editor. 
3. Select the **Markdown** option to switch to the Markdown editor. 
4. Add the markdown content.

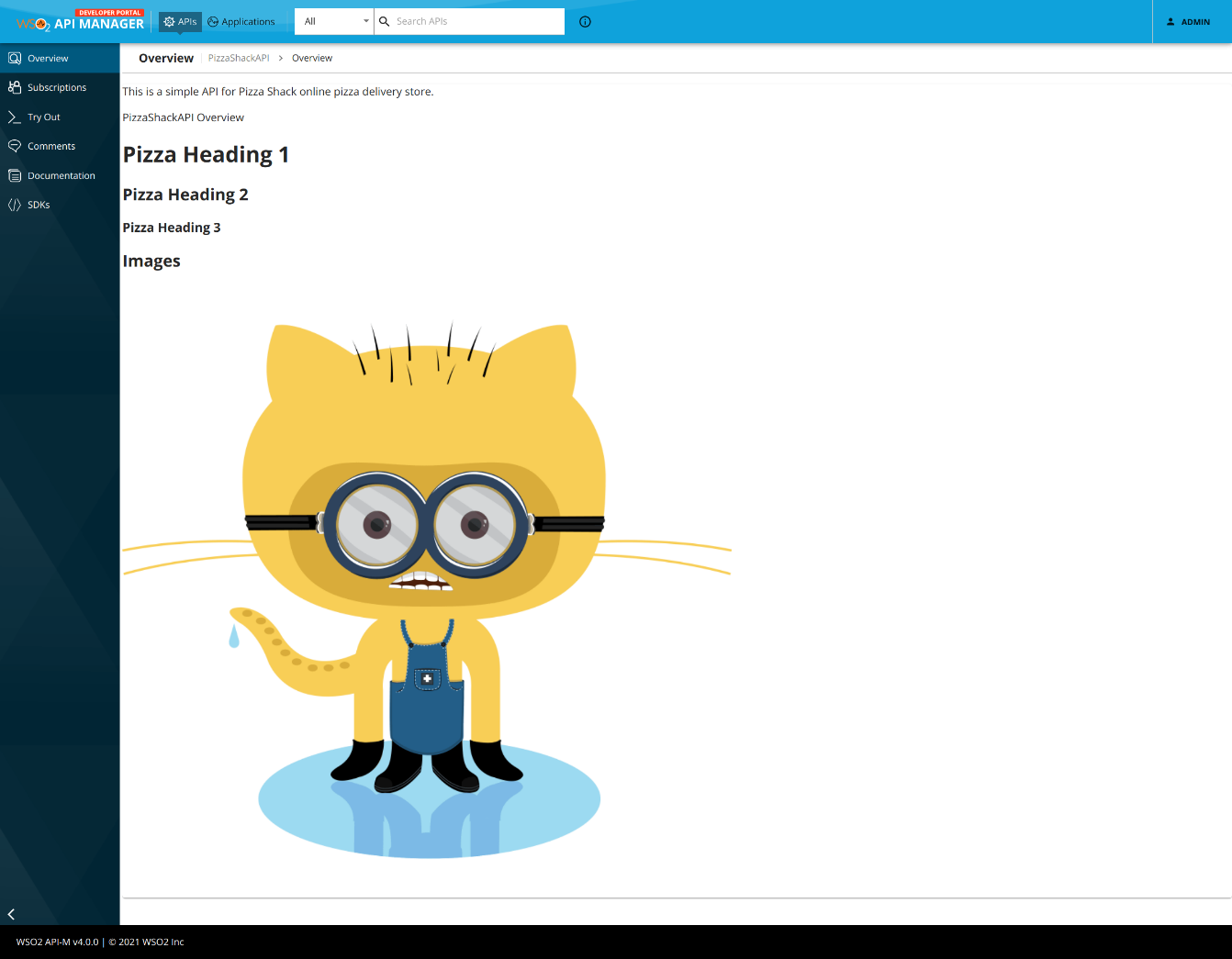
Following keys can be used within the markdown to get some of the API properties.

| Property Name | Key to use in markdown |
| --- | --- |
| name | \_\_\_name\_\_\_ |
| authorizationHeader | \_\_\_authorizationHeader\_\_\_ |
| avgRating | \_\_\_avgRating\_\_\_ |
| context | \_\_\_context\_\_\_ |
| id | \_\_\_id\_\_\_ |
| lifeCycleStatus | \_\_\_lifeCycleStatus\_\_\_ |
| provider | \_\_\_provider\_\_\_ |
| type | \_\_\_type\_\_\_ |
| version | \_\_\_version\_\_\_ |



Above screen demonstrates the use of \_\_\_name\_\_\_ key to display API name within the markdown content.

1. Click **Update Content**. Then click **Save** at the bottom of the Design Configurations form.
2. Go to the Developer Portal and select the API for which the markdown content was added. Overview for the selected API will be rendered from the markdown content.



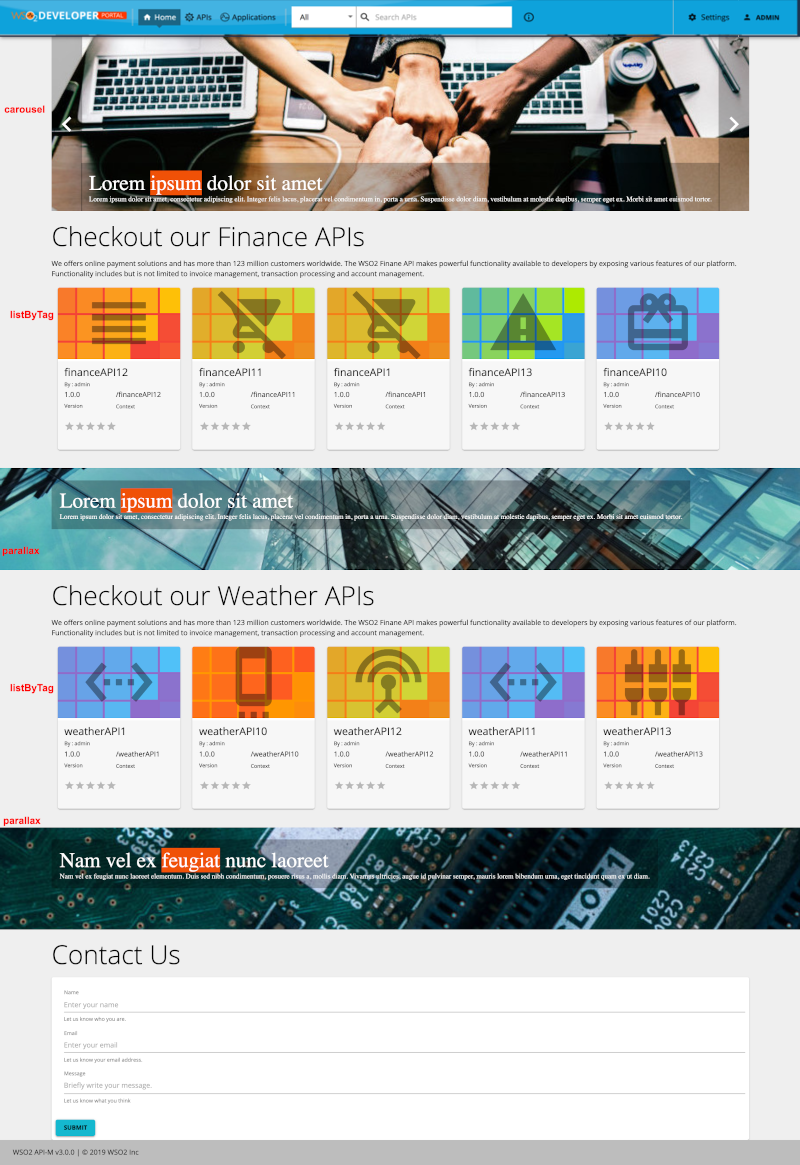
# **Enable or Disable Home Page**

It is a common requirement to have a Landing Page if the developer portal is required to present corporate branding. The default Landing Page is the API listing page. But when we enable the Home Page, there will be an additional Landing Page. It can be customized based on the design requirements by configuring the defaultTheme.js file.

The defaultTheme.js file has all the parameters defining the look and feel of the developer portal. To learn more about defaultTheme.js refer [here](https://apim.docs.wso2.com/en/latest/reference/customize-product/customizations/customizing-the-developer-portal/overriding-developer-portal-theme/#global-theming).

### **The Home Page has Four Sections**

1. Carousel
2. First Description and the listing of APIs filtered by a given tag (provided via the theme file)
3. Second description and the listing of APIs filtered by a given tag (provided via the theme file)
4. "Contact Us" section



## **Steps to Configure the Landing Page**

By using defaultTheme.js as a reference , you could customize these link tabs by modifying userTheme.json file. For that

1. Open <API-M\_HOME>/repository/deployment/server/webapps/devportal/site/public/theme/userTheme.json file in a text editor and set the attributes accordingly.

Following JSON is an example for a userTheme.js to define the look and feel, and the behavior of the landing page. You can set the attributes (components) such as carousel, listByTag, parallax and contact as shown in the below example. (Refer to the above screenshot to identify the components referred by the attribute names)

``` js { "custom": { "landingPage": { "active": true, "carousel": { "active": true, "slides": [ { "src": "/site/public/images/landing/01.jpg", "title": "Lorem ipsum dolor sit amet", "content": "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer felis lacus, placerat vel condimentum in, porta a urna. Suspendisse dolor diam, vestibulum at molestie dapibus, semper eget ex. Morbi sit amet euismod tortor." }, { "src": "/site/public/images/landing/02.jpg", "title": "Curabitur malesuada arcu sapien", "content": "Curabitur malesuada arcu sapien, suscipit egestas purus efficitur vitae. Etiam vulputate hendrerit venenatis. " }, { "src": "/site/public/images/landing/03.jpg", "title": "Nam vel ex feugiat nunc laoreet", "content": "Nam vel ex feugiat nunc laoreet elementum. Duis sed nibh condimentum, posuere risus a, mollis diam. Vivamus ultricies, augue id pulvinar semper, mauris lorem bibendum urna, eget tincidunt quam ex ut diam." } ] }, "listByTag": { "active": true, "content": [ { "tag": "finance", "title": "Checkout our Finance APIs", "description": "WSO2 offers online payment solutions and have more than 123 million customers worldwide. The WSO2 Finance API makes powerful functionality available to developers by exposing various features of the platform. Functionality includes but is not limited to invoice management, transaction processing, and account management.", "maxCount": 5 }, { "tag": "weather", "title": "Checkout our Weather APIs", "description": "WSO2 offers online payment solutions and have more than 123 million customers worldwide. The WSO2 Finance API makes powerful functionality available to developers by exposing various features of the platform. Functionality includes but is not limited to invoice management, transaction processing, and account management.", "maxCount": 5 } ] }, "parallax": { "active": true, "content": [ { "src": "/site/public/images/landing/parallax1.jpg", "title": "Lorem ipsum dolor sit amet", "content": "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer felis lacus, placerat vel condimentum in, porta a urna. Suspendisse dolor diam, vestibulum at molestie dapibus, semper eget ex. Morbi sit amet euismod tortor." }, { "src": "/site/public/images/landing/parallax2.jpg", "title": "Nam vel ex feugiat nunc laoreet", "content": "Nam vel ex feugiat nunc laoreet elementum. Duis sed nibh condimentum, posuere risus a, mollis diam. Vivamus ultricies, augue id pulvinar semper, mauris lorem bibendum urna, eget tincidunt quam ex ut diam." } ] }, "contact": { "active": true } } } }

```

| Option | type | Values | Description |
| --- | --- | --- | --- |
| active | boolean | true, false | If true the feature is enabled. If false(default), the feature is disabled |
| carousel.active | boolean | true, false | If true( default ) the carousel section is visible. If false carousel section is hidden. |
| carousel.slides | array |  | Provide the content for the carousel animation at the page top. Slide is a json of following format. |
| listByTag.active | boolean | true, false | If true( default ) the listByTag section is visible. If false listByTag section is hidden. |
| listByTag.content | array |  | Provide the content for the API grouping. It's an array of JSON where the JSON of following format. |
| parallax.active | boolean | true, false | If true( default ) the parallax scrolling section is visible. If false parallax scrolling section is hidden. |
| contact.active | boolean | true, false | If true( default ) the contact us section is visible. If false contact us section is hidden. |

1. Refresh the Developer Portal to view the changes.

The Landing Page provides a head start for developers who try to rebrand the developer portal for their needs. If the requirements are much complicated, then you need to override the relevant components. The steps to override only specific react components can be found [here](https://apim.docs.wso2.com/en/latest/reference/customize-product/customizations/advanced-ui-customization/).

# **Enable or Disable Footer**

The footer section is visible by default. You can hide the footer by configuring the defaultTheme.js file.

The defaultTheme.js file has all the parameters defining the look and feel of the developer portal. To learn more about defaultTheme.js refer [here](https://apim.docs.wso2.com/en/latest/reference/customize-product/customizations/customizing-the-developer-portal/overriding-developer-portal-theme/#global-theming).

1. Open the <API-M\_HOME>/repository/deployment/server/webapps/devportal/src/main/webapp/site/public/theme/defaultTheme.js file in a text editor and set the themes.light.custom.footer.active attribute as false to hide the footer.
2. Refresh the Developer Portal to view the changes.

### **The following attributes are available for the footer**

**const** Configurations = {

custom: {

footer: {

active: true,

text: '',

background: '#000',

color: '#fff',

},

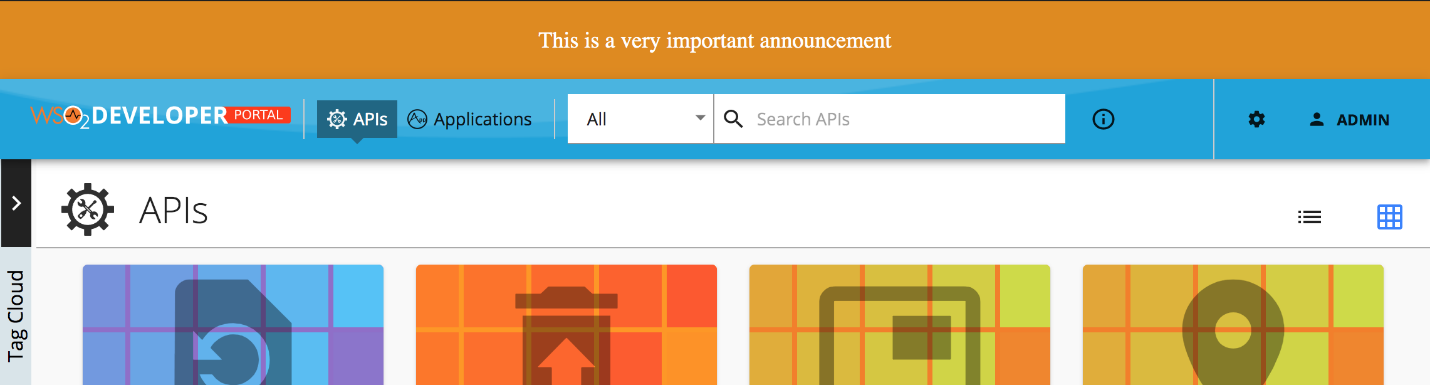
},

};

| Option | type | Description |
| --- | --- | --- |
| active | boolean | Footer is visible if true (default). Hidden if false. |
| text | string | Leave empty to show the default WSO2 Text. Provide custom text to display your own thing. |
| background | string | Set the background color of the footer |
| color | string | Set the font color for the footer text |

# **Enable or Disable Banner**

The banner section is hidden by default. The banner section can be used to show an announcement to the developer portal users as follows.



You can show a banner by configuring the defaultTheme.js file.

The defaultTheme.js file has all the parameters defining the look and feel of the developer portal. To learn more about defaultTheme.js refer [here](https://apim.docs.wso2.com/en/latest/reference/customize-product/customizations/customizing-the-developer-portal/overriding-developer-portal-theme/#global-theming).

1. Open the <API-M\_HOME>/repository/deployment/server/webapps/devportal/src/main/webapp/site/public/theme/defaultTheme.js file in a text editor and set the themes.light.custom.banner.active attribute as true to show the banner.
2. Refresh the Developer Portal to view the changes.

### **The following attributes available for the banner**

**const** Configurations = {

custom: {

banner: {

active: true,

style: 'text',

image: '/site/public/images/landing/01.jpg',

text: 'This is a very important announcement',

color: '#ffffff',

background: '#e08a00',

padding: 20,

margin: 0,

fontSize: 18,

textAlign: 'center',

},

},

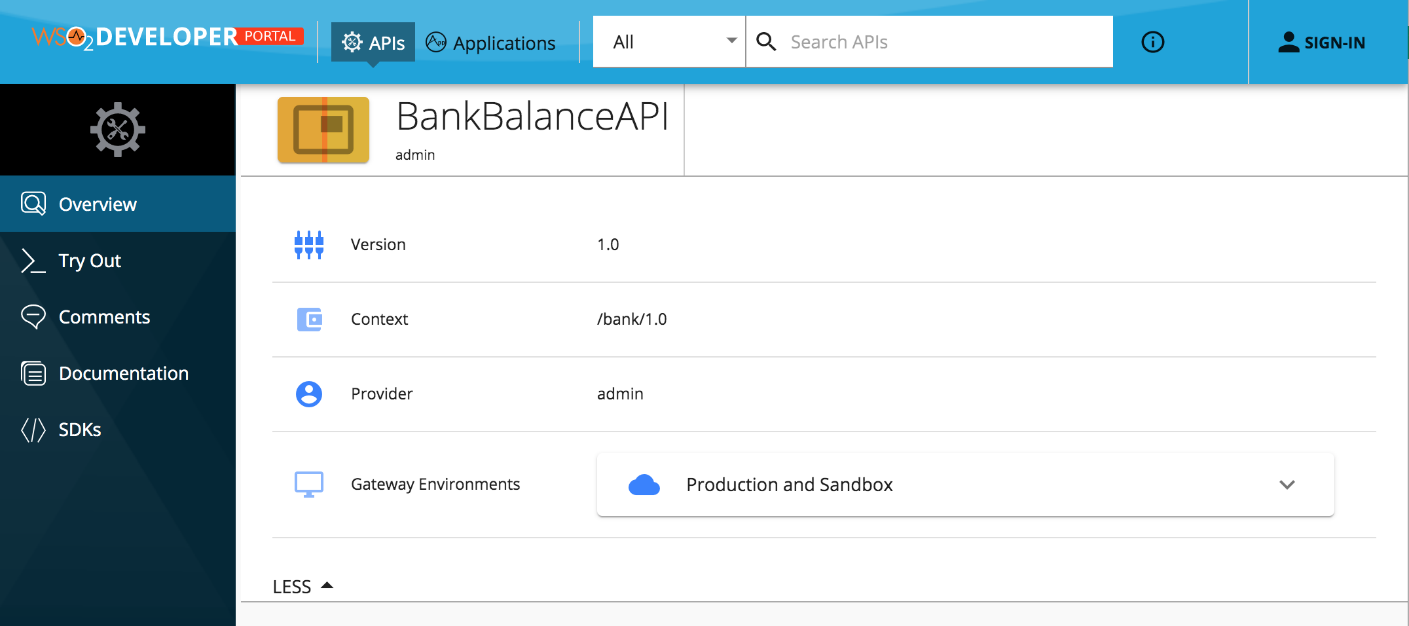
};

| Option | type | Description |
| --- | --- | --- |
| active | boolean | Banner is visible if true. Hidden if false (default). |
| style | string | Style value can be 'image' or 'text' (default). If text it will display the 'banner.text' value else it will display the 'banner.image'. |
| image | string | Path to banner image |
| text | string | Text to be shown |
| color | string | Color of the banner text |
| background | string | Background color of the banner |
| padding | integer | CSS padding of the banner |
| margin | integer | CSS margin of the banner |
| fontSize | integer | CSS font-size attribute of the banner text |
| textAlign | string | Text align direction |

# **Styling API Details Left Menu**

The API details left menu can be customized to match with your design needs by configuring the defaultTheme.js file.

The defaultTheme.js file has all the parameters defining the look and feel of the developer portal. To learn more about defaultTheme.js refer [here](https://apim.docs.wso2.com/en/latest/reference/customize-product/customizations/customizing-the-developer-portal/overriding-developer-portal-theme/#global-theming).



You can change the themes.light.custom.leftMenu attributes to change the left menu styling. Note that, these changes will effect the same way to application details left menu as well.

1. Open the Open <API-M\_HOME>/repository/deployment/server/webapps/devportal/site/public/theme/userTheme.json file in a text editor and update the attributes.
2. Refresh the Developer Portal to view the changes.

### **The following attributes are available for the leftMenu**

{

"custom": {

"leftMenu": {

"position": "vertical-left",

"style": "icon left",

"iconSize": 24,

"leftMenuTextStyle": "uppercase",

"width": 180,

"background": "#012534",

"backgroundImage": "/site/public/images/leftMenuBack.png",

"leftMenuActive": "#00597f",

"leftMenuActiveSubmenu": "#0d1723",

"activeBackground": "#191e46",

"rootIconVisible": true,

"rootIconSize": 42,

"rootIconTextVisible": false,

"rootBackground": "#000"

}

}

}

The above JSON defines the default look and feel.

We can change the menu to take different positions. For an example, the following configuration sets the menu as a toolbar by just changing the values of the above JSON as follows.

{

"custom": {

"leftMenu": {

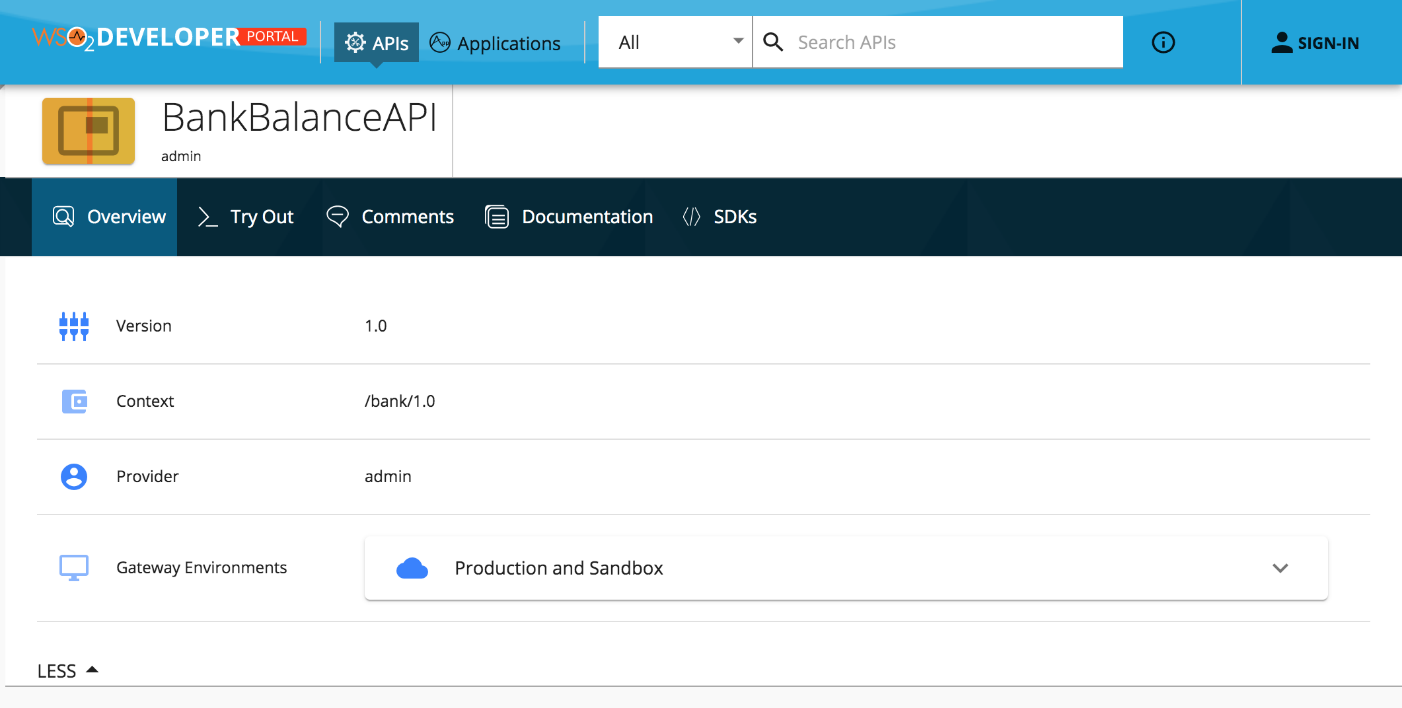
"position": "horizontal",

"rootIconVisible": true

}

}

}



The following will set the menu to the right hand side and will disable the icons.

{

"custom": {

"leftMenu": {

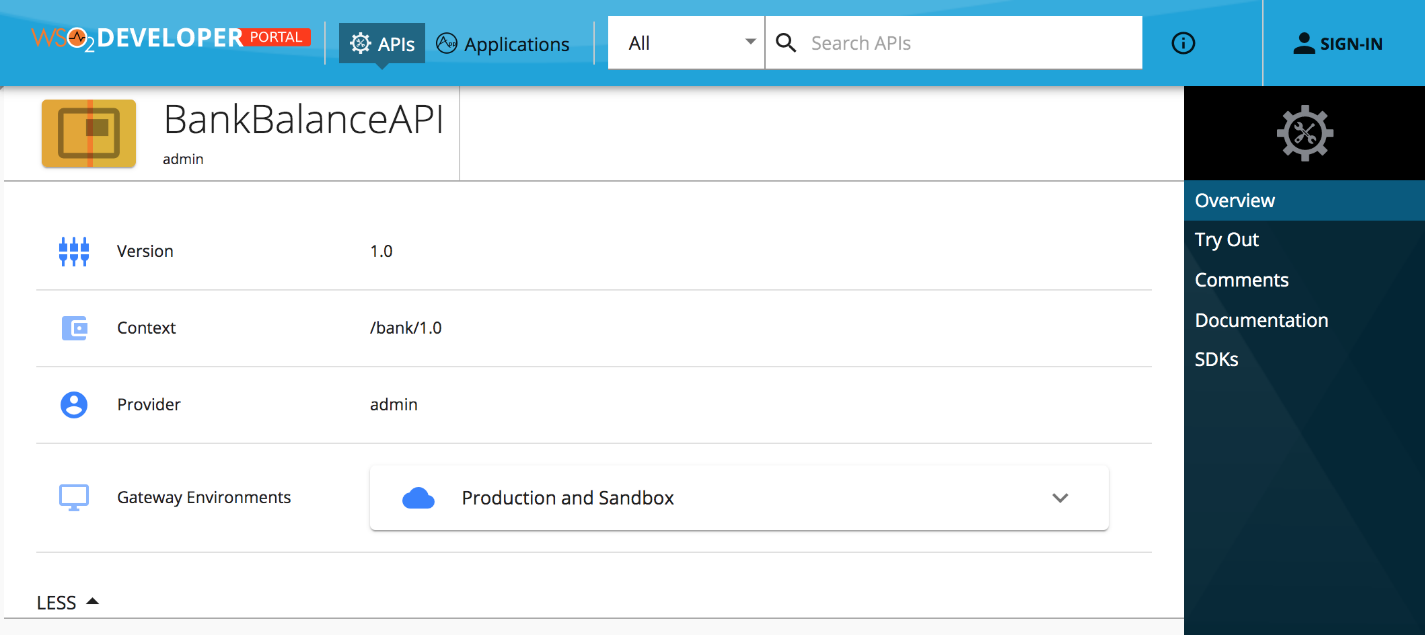
"position": "vertical-right",

"style": "no icon"

}

}

}



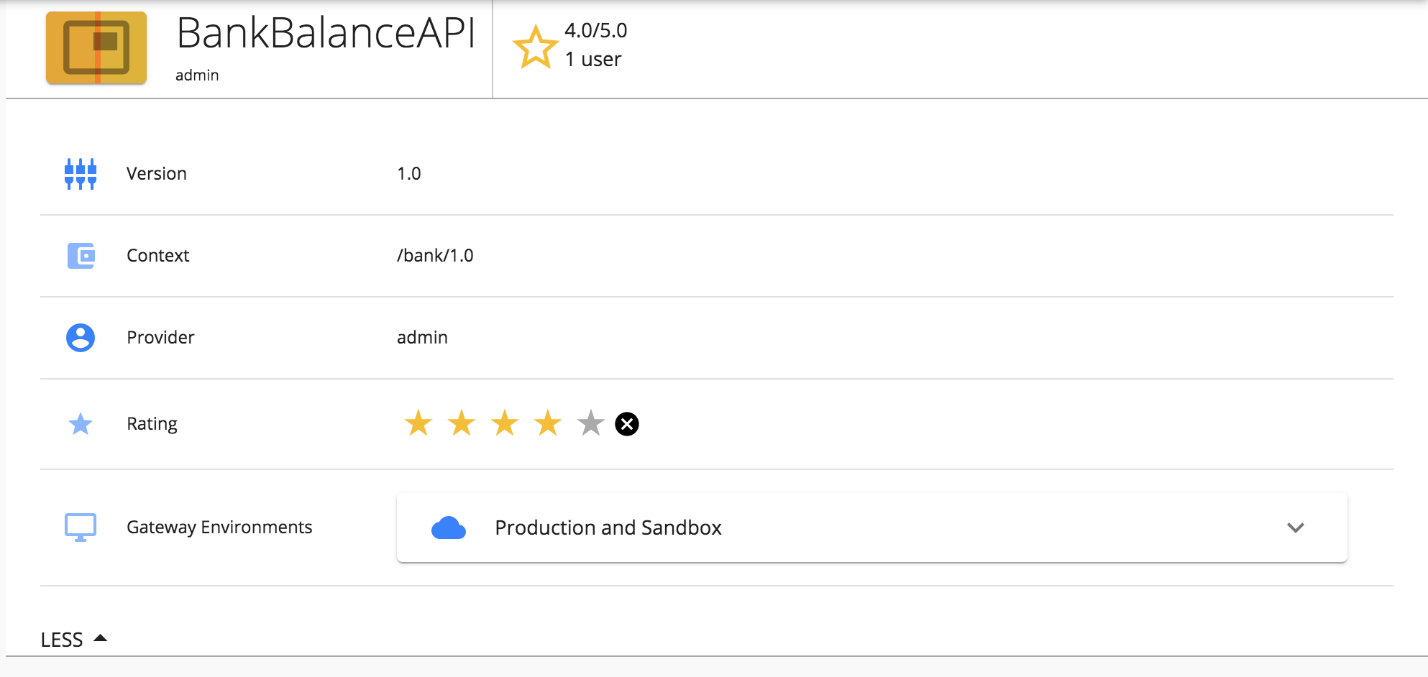
| Option | type | Description |
| --- | --- | --- |
| position | string | Sets the position of the menu. Accepts 'horizontal', 'vertical-left', 'vertical-right' |
| style | string | Sets the menu icon position and visibility. Accepts'icon top', 'icon left', 'no icon', 'no text' |
| iconSize | integer | Icon size in pixels |
| leftMenuTextStyle | string | Set the font style for the menu text |
| width | integer | Defines the menu width |
| background | string | Set the background for the left menu |
| backgroundImage | string | Set a background image to the left menu ex: '/site/public/images/leftMenuBack.png' |
| leftMenuActive | string | Set the background color of the menu when it's selected |
| leftMenuActiveSubmenu | string | Set the background color for submenu elements. Example use can be found at the Application details page |
| rootIconVisible | boolean | Set the top Icons visibility. By default set to true, Set it to false to hide it. |
| rootIconSize | integer | Define the size of root icon. The value is considered in pixels when rendering. |
| rootIconTextVisible | boolean | Set the visibility of the root icon text. By default it's hidden. |
| rootBackground | string | Set the background color of the root icon containing element |

With these configurations, we tried to handle most of the use cases for rebranding. But if someone wants to do a change that is not supported, for an example change the icons or make the menu collapsible, then they need to override the relevant React component. Refer to the [Advanced Customization](https://apim.docs.wso2.com/en/latest/reference/customize-product/customizations/advanced-ui-customization/) for more information.

# **Styling API and Application Details Info Section**

The "API Details Info" section which is shown below can be customized according to your design needs by configuring the defaultTheme.js file.

The defaultTheme.js file has all the parameters defining the look and feel of the developer portal. To learn more about defaultTheme.js refer [here](https://apim.docs.wso2.com/en/latest/reference/customize-product/customizations/customizing-the-developer-portal/overriding-developer-portal-theme/#global-theming).

[](https://apim.docs.wso2.com/en/latest/assets/img/learn/styling-api-details-info-section1.png)

Edit the attributes in themes.light.custom.infoBar by following the below steps to change the styling of "API Details Info" section and the "Application Details Info" section.

1. Open the <API-M\_HOME>/repository/deployment/server/webapps/devportal/site/public/theme/userTheme.json file in a text editor and update the attributes.
2. Refresh the Developer Portal to view the changes.

### **The following attributes are available for infoBar**

The JSON sample code that is given below defines the default look and feel.

{

"custom": {

"infoBar": {

"height": 70,

"background": "#ffffff",

"showThumbnail": true,

"starColor": "#f6bf21",

"sliderBackground": "#ffffff",

"iconOddColor": "#347eff",

"iconEvenColor": "#89b4ff",

"listGridSelectedColor": "#347eff",

"tagChipBackground": "#7dd7f5"

}

}

}

#### Example: Change the colors of UI elements

{

"custom": {

"infoBar": {

"height": 70,

"background": "#000",

"showThumbnail": false,

"starColor": "#ff1a1a",

"sliderBackground": "#000",

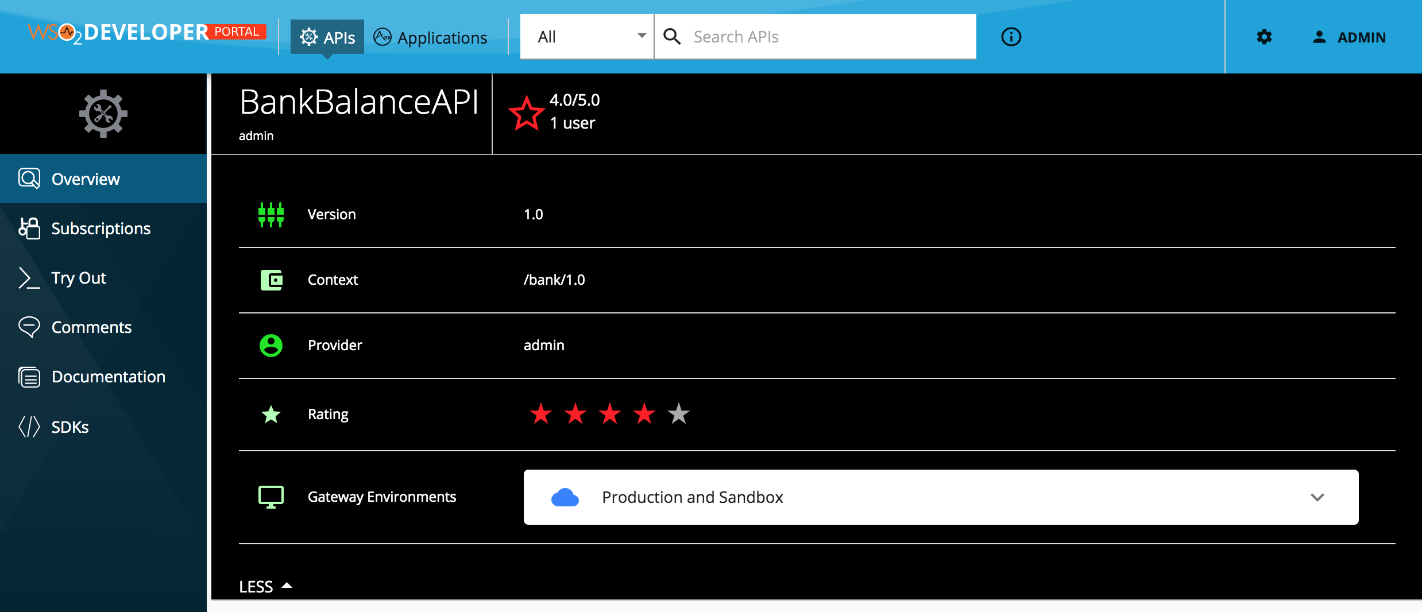
"iconOddColor": "#00e600",

"iconEvenColor": "#b3ffb3"

}

}

}

[](https://apim.docs.wso2.com/en/latest/assets/img/learn/styling-api-details-info-section2.png)

### **Attribute options**

| Option | type | Description |
| --- | --- | --- |
| height | integer | Height of the top most row with the API name up to the API link, given in pixels. |
| background | string | Background color of the component |
| showThumbnail | boolean | Show/hide the API thumbnail. Set to visible, by default. |
| starColor | string | Set the color of the star rating. |
| sliderBackground | string | Set the background color of the 'SHOW'/'HIDE' collapsible section at the bottom. |
| iconOddColor | string | Change the color of the icons displayed on the left of the odd-numbered rows. |
| iconEvenColor | string | Change the color of the icons displayed on the left of the even-numbered rows. |
| listGridSelectedColor | string | Define the color of a selected icon (Grid/List) view of the API listing page. |
| tagChipBackground | string | Change the background color of the tags. |

You can handle most of the use cases with regard to rebranding using these configurations. However, if you need to make a change that is not supported by the defaultTheme.js file, then you need to override the relevant React component. For more information, see [Advanced Customization](https://apim.docs.wso2.com/en/latest/reference/customize-product/customizations/advanced-ui-customization/) for more information.

### **Known issues and workarounds**

The API details page tags are not visible- <https://github.com/wso2/product-apim/issues/7849>.

As a workaround, add the following to the <API-M\_HOME>/repository/deployment/server/webapps/devportal/src/main/webapp/site/public/theme/defaultTheme.js file.

**const** Configurations = {

custom: {

tagWise: {

active: true,

},

},

};

# **Changing the Logo and Header Styles**

The header section can be customized to match different design needs by configuring the defaultTheme.js file.

The defaultTheme.js file has all the parameters defining the look and feel of the developer portal. To learn more about defaultTheme.js refer [here](https://apim.docs.wso2.com/en/latest/reference/customize-product/customizations/customizing-the-developer-portal/overriding-developer-portal-theme/#global-theming).

The following is the default look and the configuration. The default header of the API Manager Developer Portal is shown below.



1. Open the <API-M\_HOME>/repository/deployment/server/webapps/devportal/site/public/theme/userTheme.json file in a text editor and set the attributes accordingly as shown below which customizes the logo and the header of the developer portal.

js { "custom": { "appBar": { "logo": "/site/public/images/logo.svg", "logoHeight": 19, "logoWidth": 208, "background": "#0fa2db", "backgroundImage": "/site/public/images/appbarBack.png", "searchInputBackground": "#fff", "searchInputActiveBackground": "#fff", "activeBackground": "#1c6584", "showSearch": true, "drawerWidth": 200 } } }

| Option | type | Description |
| --- | --- | --- |
| logo | string | Relative path to logo |
| logoHeight | integer | Logo height in pixels |
| logoWidth | integer | Logo width in pixels |
| background | string | Background color of the header |
| backgroundImage | string | Set the background image to the header. Ex: '/site/public/images/appbarBack.png' |
| searchInputBackground | string | Set the background color for the search input |
| searchInputActiveBackground | string | Set the background color for the search input |
| activeBackground | string | Background color of the selected header menu item |
| drawerWidth | integer | Small resolutions will collopse the top menu in to a toggle drawer. This property sets the it's width in pixels |

1. Refresh the Developer Portal to view the changes.

#### Example

We can change the logo and the header background as follows by changing the parameters. An example is shown below.

{

"custom": {

"appBar": {

"logo": "/site/public/images/logo-black.png",

"logoHeight": 66,

"logoWidth": 200,

"background": "#a10207",

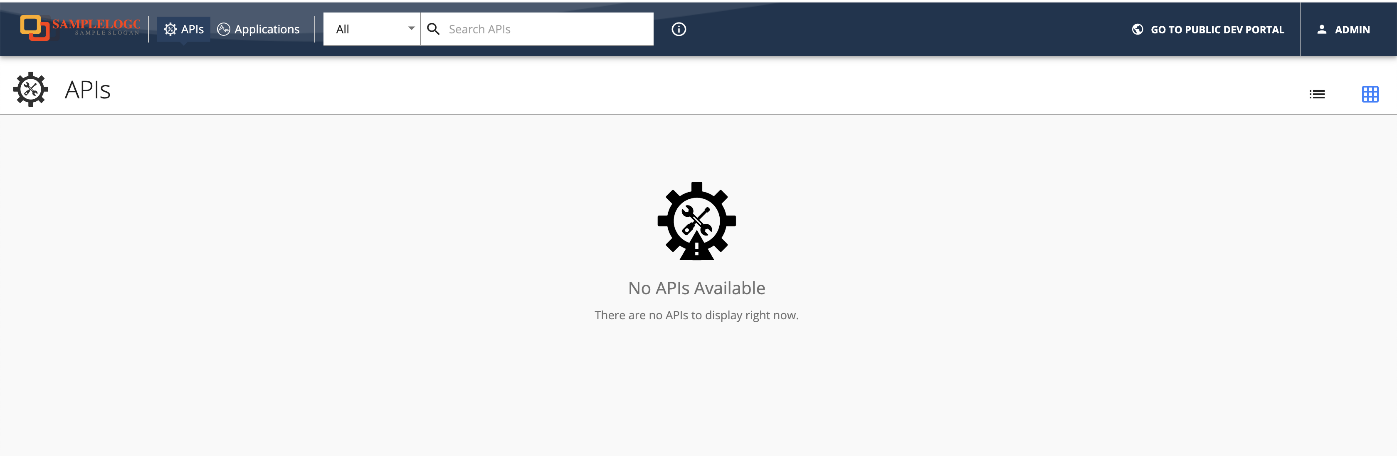
"activeBackground": "#ffd500"

}

}

}

Make sure you restart the server for the changes to take effect.



Expected Outcome

As a result of this exercise, you will be able to customize the Developer Portal according to your requirements.