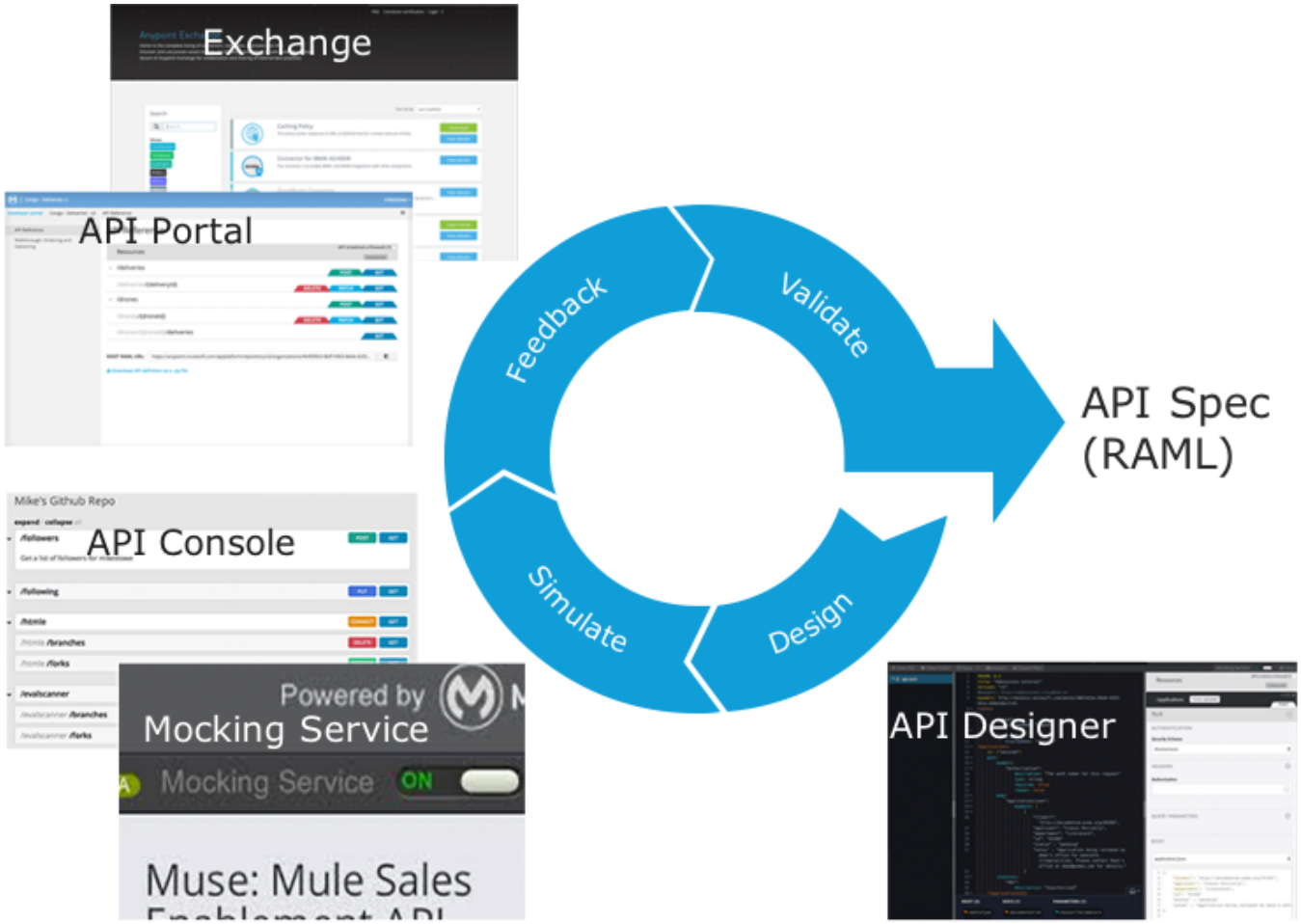
Module 2: Designing APIs



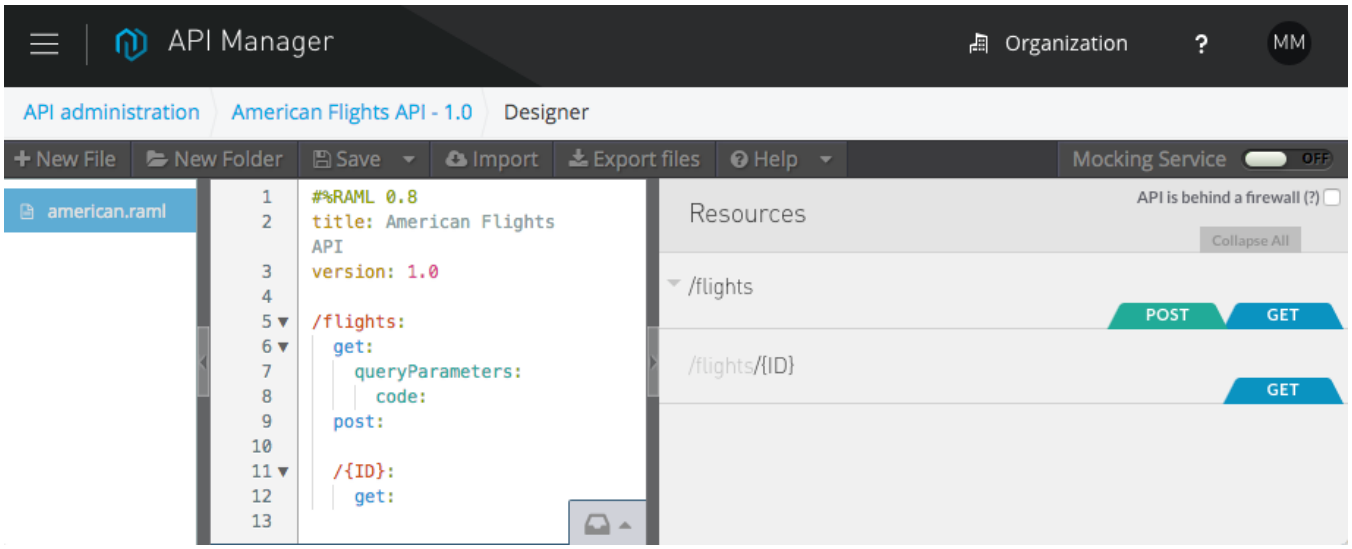
Objectives:

- Define an API with RAML, the Restful API Modeling Language.
- Mock an API to test its design before it is built.
- Create a portal for developers to learn how to use an API.
- Make an API discoverable by adding it to the private Exchange.

Walkthrough 2-1: Use API Designer to define an API with RAML

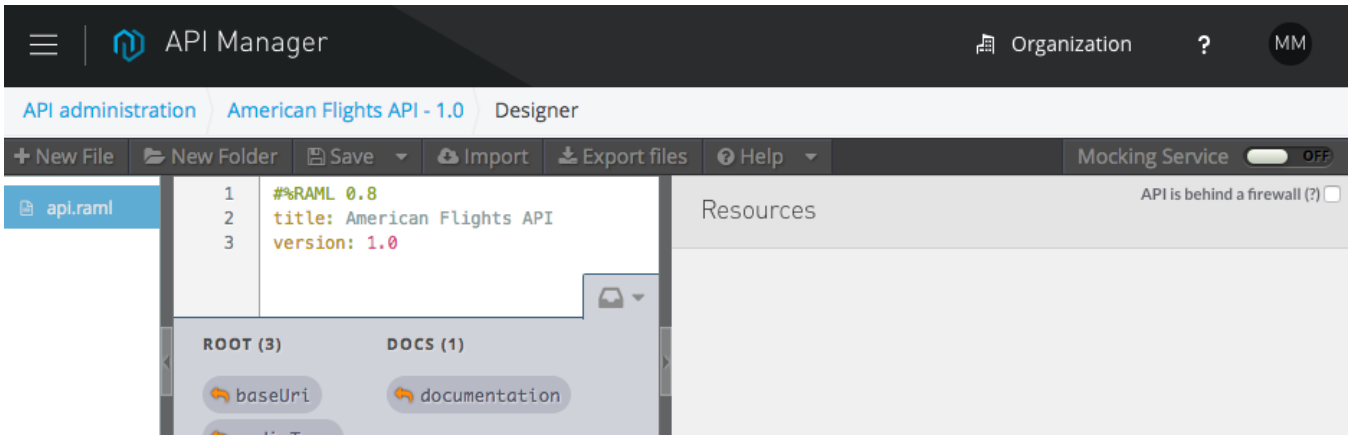
In this walkthrough, you create an API definition with RAML using the API Designer. You will:

- Define resources and nested resources.
- Interact with the API using the API Console.
- Define get and post methods.
- Specify query parameters.



Open the API Designer

1. Return to the API details page for American Flights API in Anypoint Platform.
2. In the API Definition section, click the Define API in API designer link; the API Designer should open.



Note: To change the background color from black to white, press Ctrl+Shift+T.

Add a RAML resource

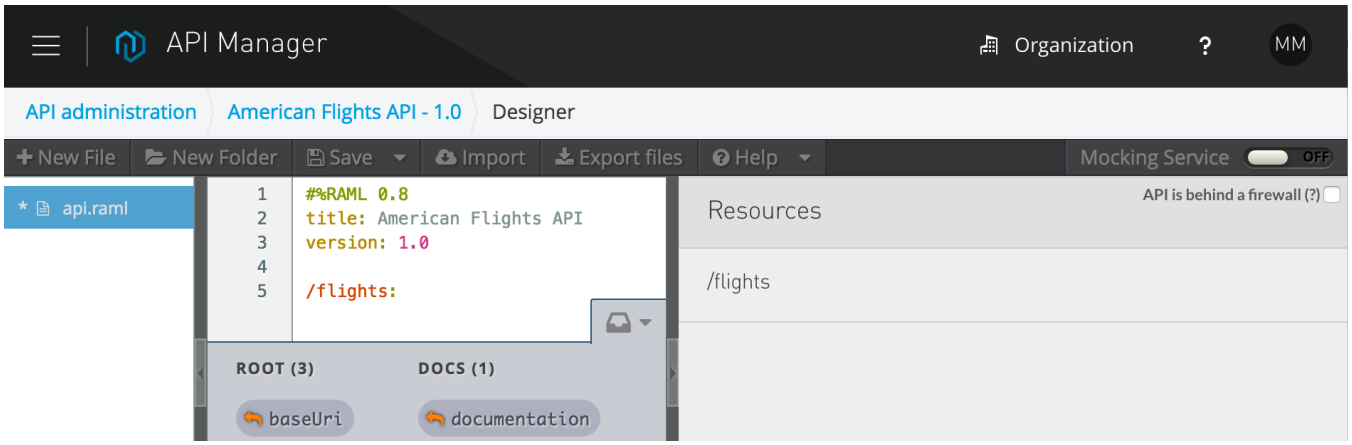
3. Place the cursor on a new line of code at the end of the file.
4. Add a resource called flights.

```
/flights:
```

View the API Console

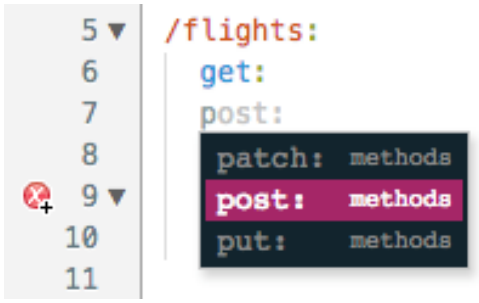
5. Look at the API Console on the right side of the window; you should see the flights resource.

   Note: If you do not see the API Console, click the arrow located in the middle of the right edge of the web browser window.

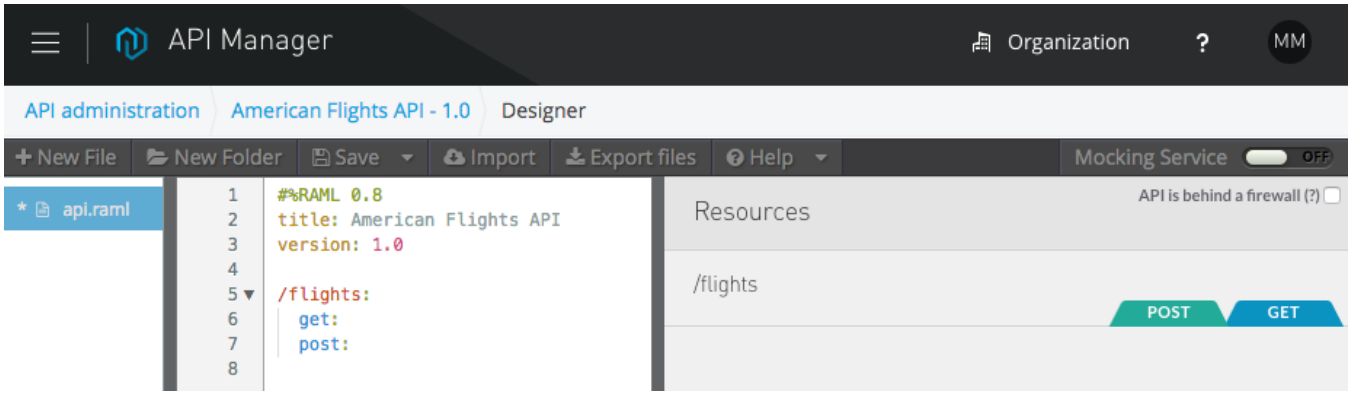


Add RAML methods

6. In the API Designer editor, go to a new line of code.
7. Indent by pressing the Tab key.
8. Press the G key and then the Enter key to add a get method.
9. Look at the API Console; you should see a GET tab for the flights resource.
10. In the editor, press the Enter key and then the P key.
11. In the popup that appears, select post and then press Enter.



12. Look at the API Console; you should see GET and POST tabs for the flights resource.

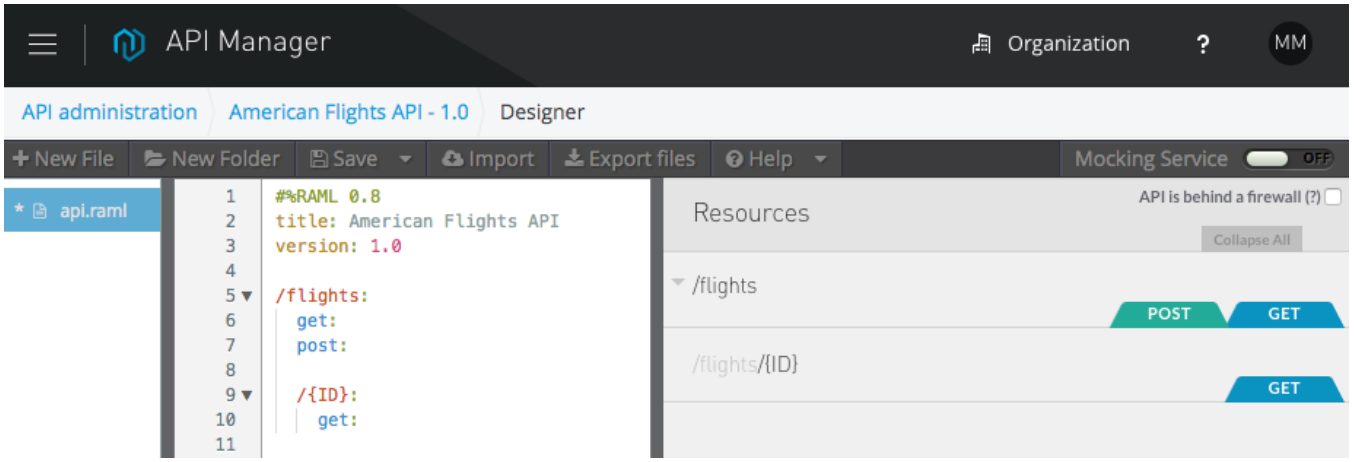


Add a nested RAML resource

13. In the editor, press Enter twice.
14. Make sure you are still under the flights resource (at the same indentation as the methods).
15. Add a nested resource for a flight with a particular ID.

```
/{ID}:
```

16. Add a get method to this resource.
17. Look at the API Console; you should see the nested resource with a GET tab.

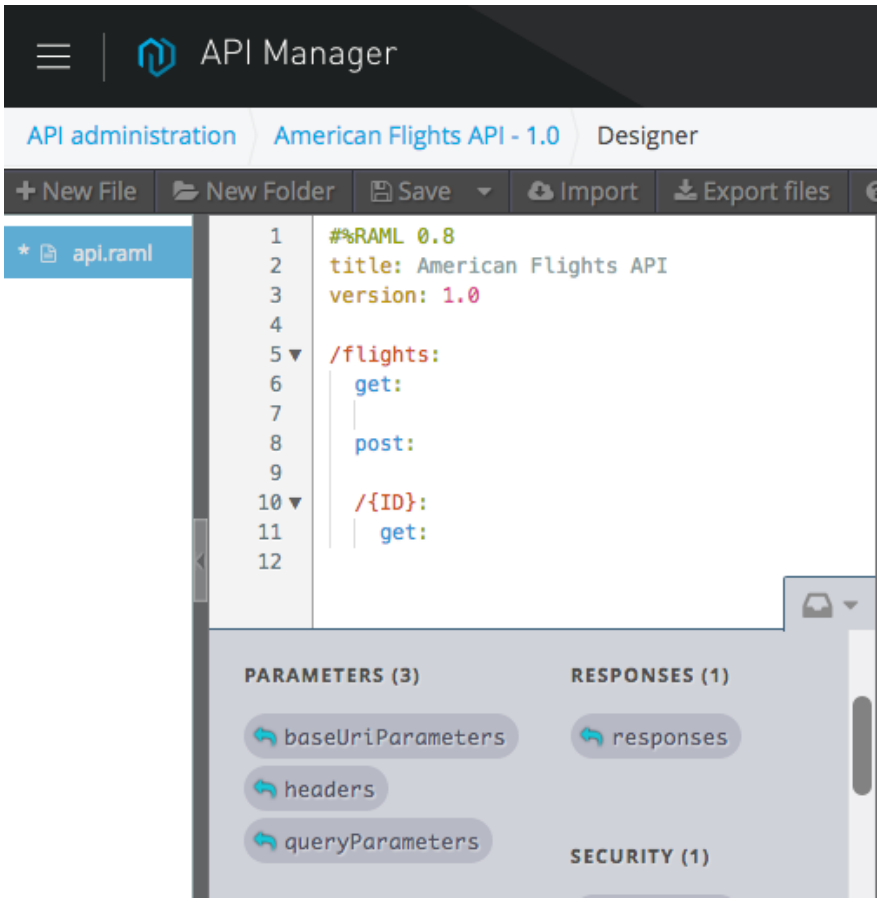


Add a query parameter

18. Locate the API Designer shelf at the bottom of the editor and look at its contents.

    Note: If you don't see the API Designer shelf, it is either minimized or there is an error in your code. To check if it is minimized, go to the bottom of the web browser window and look for its icon. If you see the icon, click it to expand it. If you don't see the icon and you also don't see the API Console, you probably have an error in your code, like a missing RAML definition.

19. In the editor, indent under the /flights get method (not the /flights/{ID} get method).
20. Look at the contents of the API Designer shelf again; the parameters shown should have changed.
21. In the API Designer shelf, click queryParameters.



22. Indent and add a key named code.

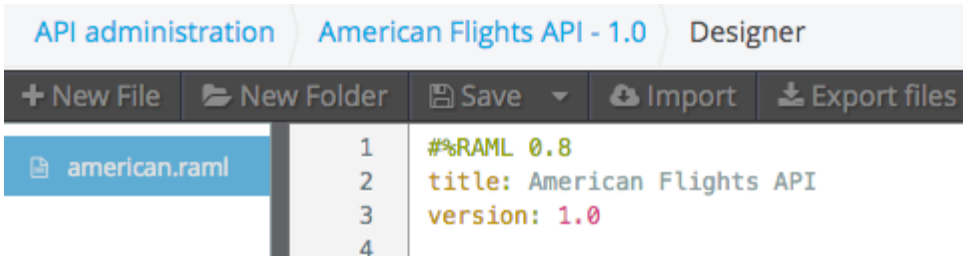
```
#%RAML 0.8
title: American Flights API
version: 1.0

/flights:
  get:
    queryParameters:
      code:
  post:

  /{ID}:
    get:
```

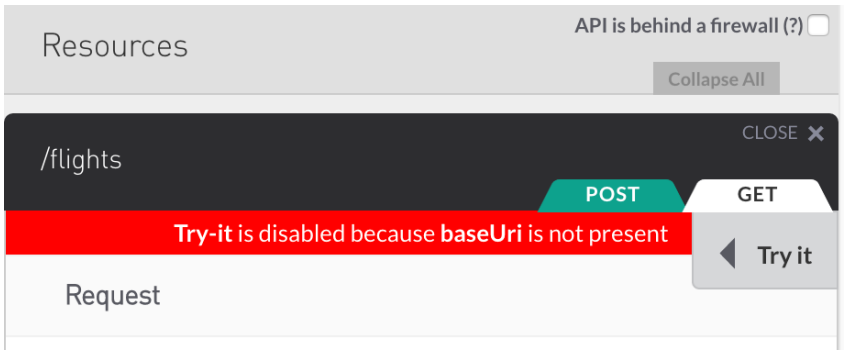
Save and rename the RAML file

23. Click the Save button.
24. Right-click api.raml and select Rename.
25. In the Rename a file dialog box, set the name to american.raml and click OK.



Try to call an API method using the API Console

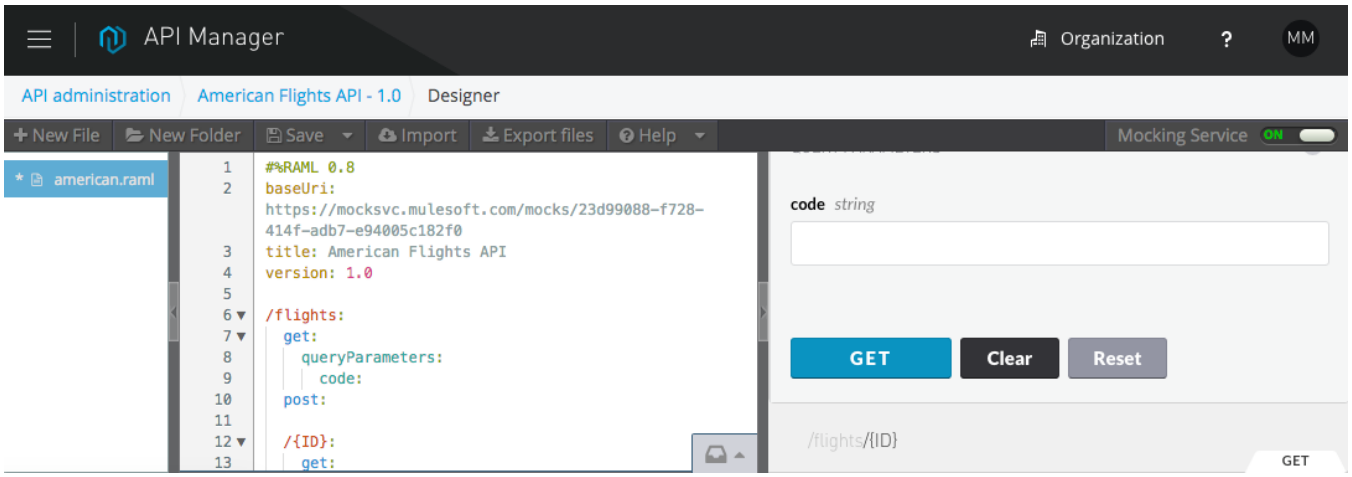
26. In the API Console, click the GET tab for the /flights resource; you should get a message that Try-it is disabled because baseUri is not present.
27. Click the CLOSE button in the upper-right corner.



Walkthrough 2-2: Use the mocking service to test an API

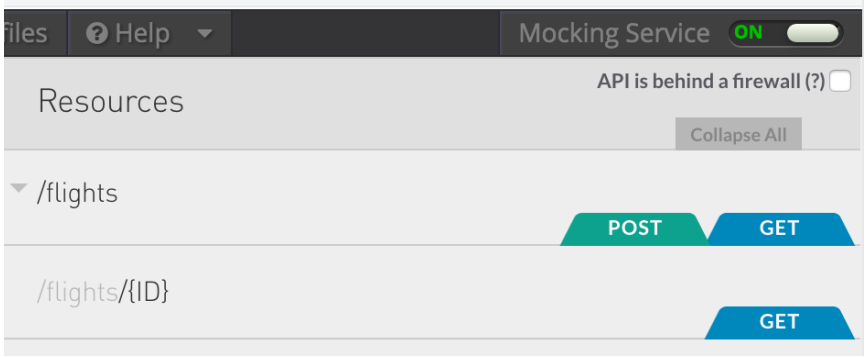
In this walkthrough, you test the API using the Anypoint Platform mocking service. You will:

- Add example responses to the API definition.
- Turn on the the mocking service.
- Use the API Console to make calls to the mocked API.

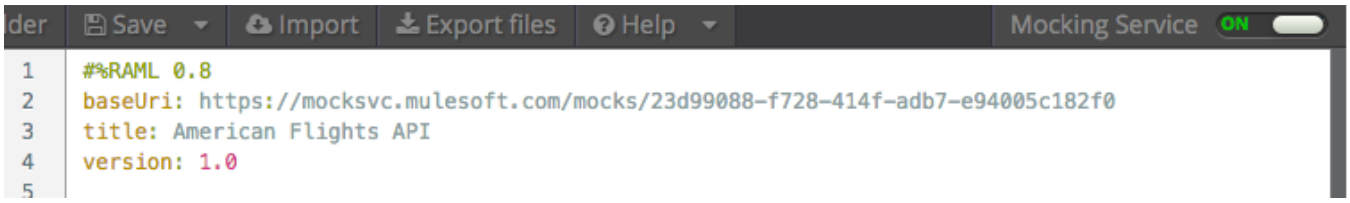


Turn on the mocking service

1. Return to API Designer.
2. Locate the Mocking Service slider in the menu bar above the API Console.
3. Slide it to on.

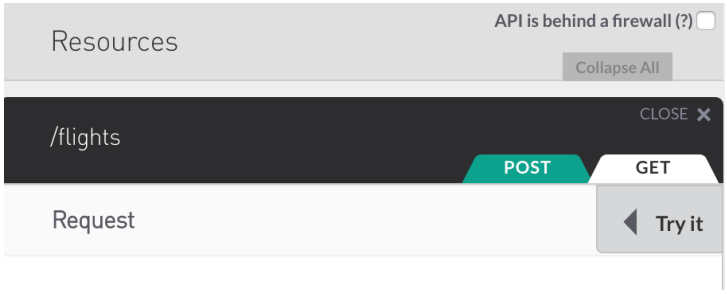


4. Look at the baseUri added to the RAML definition in the editor.

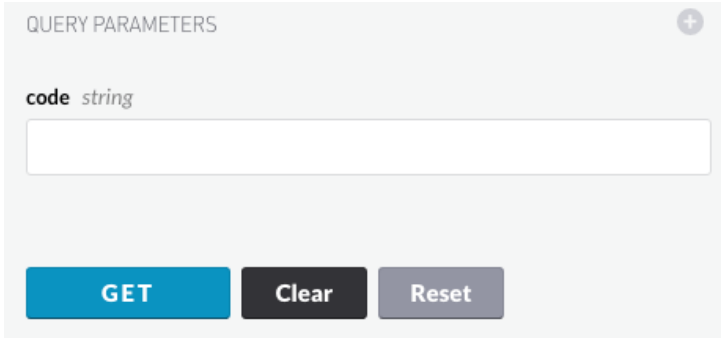


Test the /flights resource

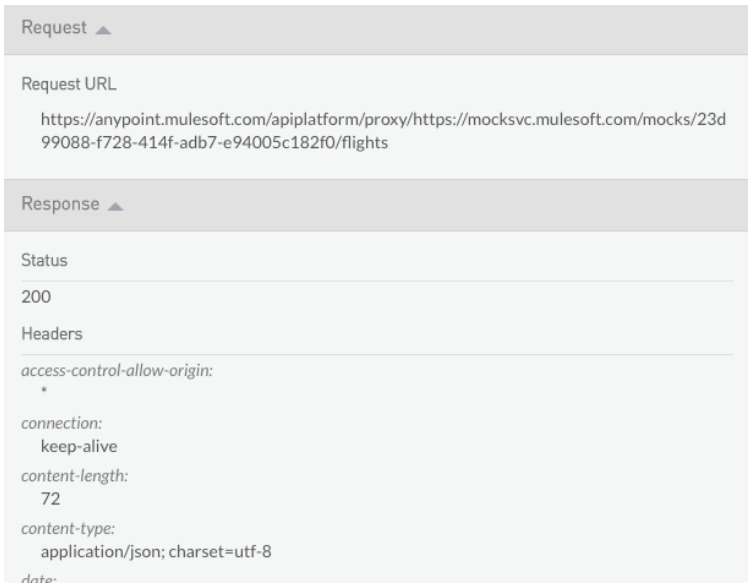
5. In the API Console, click the GET tab for the /flights resource again.
6. Click the Try it button.



7. Locate the text field for the code query parameter.
8. Notice it is not required and has no default or suggested values.
9. Click the GET button.

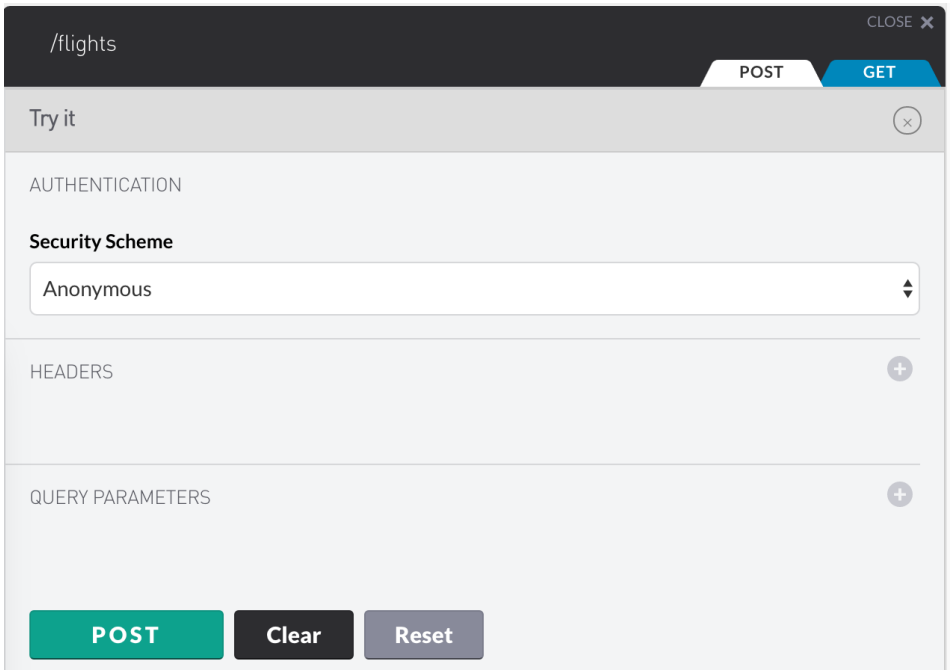


10. Look at the response; you should get a 200 status code and see the content-type is application/json by default.



11. Scroll down and look at the response body; you should get a general RAML message placeholder.

12. Scroll up and click the POST tab.
13. Notice that there is no body field in which to enter the data to post with the request.

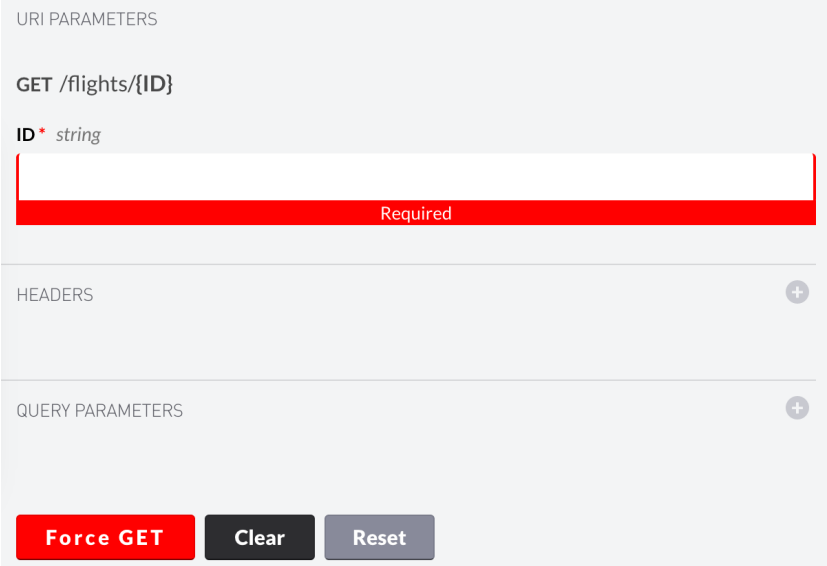


14. Click the POST button.
15. Look at the response; you should get a 200 status code and see the content-type is application/json with a default message.

Test the /flights/{ID} resource

16. Scroll down to the bottom of the API Console.
17. Click the GET tab for the /flights/{ID} resource.
18. Click the Try it button.

19. Click the GET button; you should get a message that the ID is required.

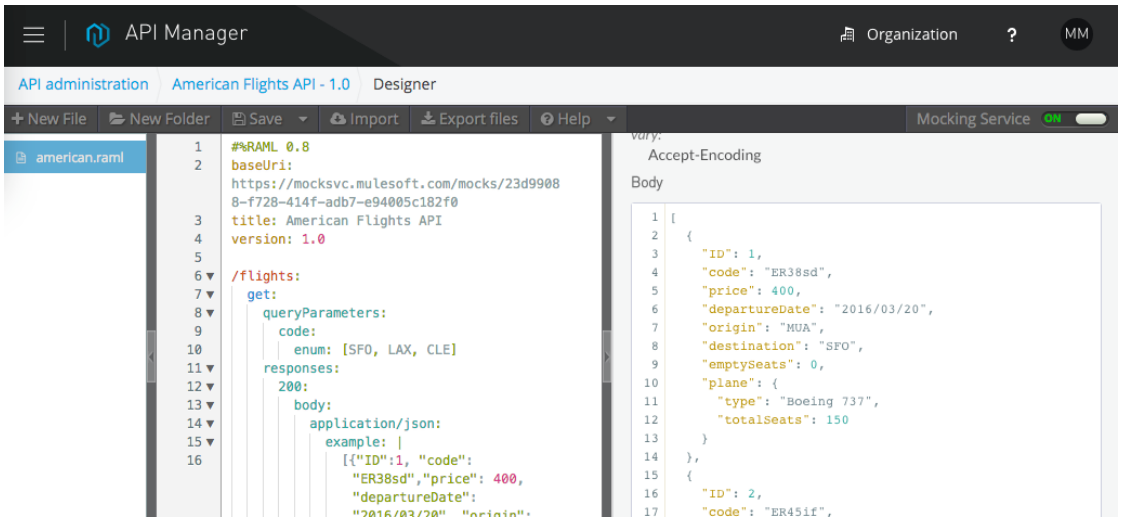


20. In the ID text box, enter a value of 10.
21. Click the GET button.
22. Look at the response; you should get a 200 status code and see the content-type is application/json with a default message.

Walkthrough 2-3: Add request and response details

In this walkthrough, you add information about each of the methods to the API definition. You will:

- Specify data types for get and post method responses.
- Add example JSON responses.
- Specify possible query parameter values.
- Test the API and get the sample responses.



View the specification for the data to be returned by the API

1. In your computer's file browser, navigate to your student files folder and open the schemas-and-examples folder.
2. Open american-flights-example.json and look at the JSON.
3. Copy the JSON.

Specify details for the /flights get method response

4. Return to API Designer.
5. Indent under the /flights get method (at the same level as queryParameters).
6. In the API Designer shelf, click responses.

7. Indent and add a 200 response.

```
/flights:
  get:
    queryParameters:
      code:
    responses:
      200:
```

8. Indent and add a body parameter from the shelf.
9. Indent and add application/json from the shelf or the popup that appears.

```
/flights:
  get:
    queryParameters:
      code:
    responses:
      200:
        body:
          application/json:
  post:
```

Add sample data for the /flights get method response

10. Indent and add an example element.
11. Add a space and then a | after the example element.
12. Indent under example and paste the example code you copied from the american-flights-example.json file.

```
/flights:
  get:
    queryParameters:
      code:
    responses:
      200:
        body:
          application/json:
            example: |
              [{"ID":1, "code": "ER38sd","price": 400, "departureDate": "2016/03/20", "origin":
               "MUA", "destination": "SFO", "emptySeats": 0, "plane": {"type": "Boeing 737",
               "totalSeats": 150}}, {"ID":2,"code": "ER45if", "price": 345.99, "departureDate":
               "2016/02/11", "origin": "MUA", "destination": "LAX", "emptySeats": 52, "plane":
               {"type": "Boeing 777", "totalSeats": 300}}]
```

Specify details for the /flights/{ID} get method response

13. For the /flights/{ID} get method, add a 200 response with application/json data.

14. Indent under application/json and add an example element with a space and then a | after it.
15. Return to the course snippets.txt file and copy the American Flights API - /flights/{ID} get response example.
16. Return to API Designer and indent and paste the example code you copied.

```
/{ID}:
  get:
    responses:
      200:
        body:
          application/json:
            example: |
              {"ID":1, "code": "ER38sd","price": 400, "departureDate": "2016/03/20", "origin":
               "MUA", "destination": "SFO", "emptySeats": 0, "plane": {"type": "Boeing 737",
               "totalSeats": 150}}
```

Specify details for the /flights post method response

17. For the /flights post method, add a 201 response with application/json data.
18. Indent under application/json and add an example element with a space and then a | after it.
19. Return to the course snippets.txt file and copy the American Flights API - /flights post response example.
20. Return to API Designer and indent and paste the example code you copied.

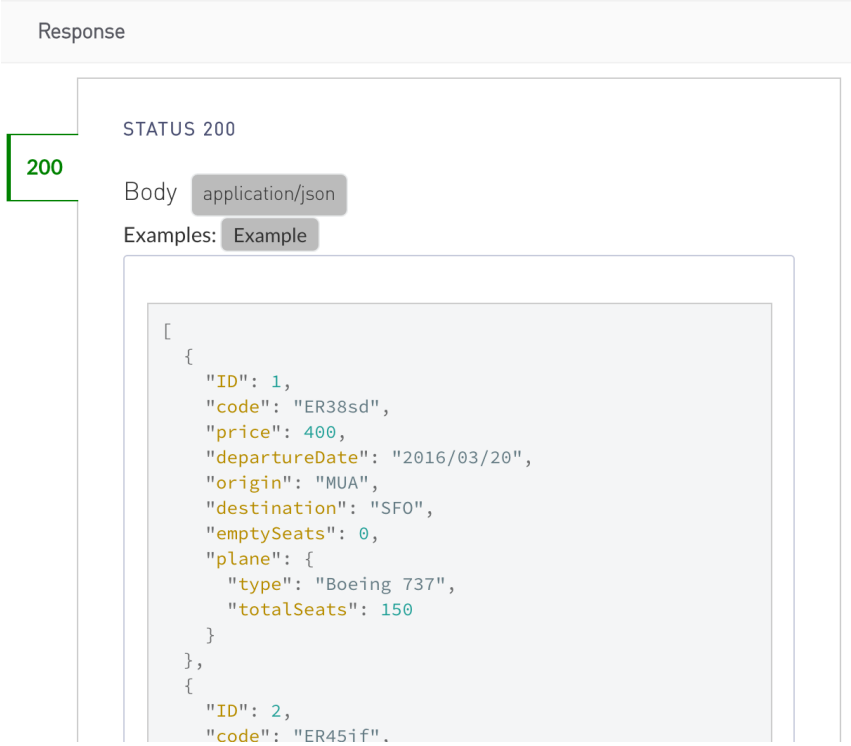
```
post:
  responses:
    201:
      body:
        application/json:
          example: |
            {"message": "Flight added (but not really)"}
```

21. Save the file.

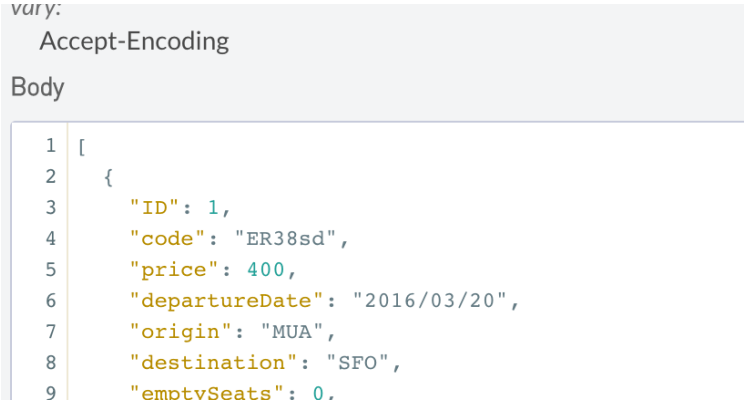
Get the sample data when making calls to the mocked API

22. In the API Console, click the GET tab for the flights resource again.

23. Scroll down and locate the new response section with the 200 response information with the example data.



24. Scroll up and click the Try it button.
25. Click the GET button; you should see the example data returned.



26. Scroll up and click the POST tab for the flights resource.
27. Click the close button for Try it.
28. Look at the request section and notice there is no information about the data that should be sent with the request – the data to be posted.

Specify details for the /flights post method request

29. Return to /flights in the editor.
30. Indent directly under the /flights post method so you are above and at the same level as responses.
31. Add a body element.
32. Indent and add application/json.
33. Indent and add an example element with a space and then a | after it.
34. Indent.
35. Return to the course snippets.txt file and copy the American Flights API - /flights post request body example.
36. Return to API Designer and indent and paste the example code you copied.

```
18 ▼   post:
19 ▼     body:
20 ▼       application/json:
21           example: |
22             {"code": "ER38sd","price": 400, "departureDate": "2016/03/20", "origin": "MUA",
               "destination": "SFO", "emptySeats": 0, "plane": {"type": "Boeing 737", "totalSeats":
               150}}
23 ▼     responses:
24 ▼       201:
25 ▼         body:
```

Specify details for the /flights get method request

37. In the /flights get method, indent under the code query parameter and look at the possible parameters in the shelf.
38. Click enum.
39. Set enum to an array of values including SFO, LAX, and CLE.

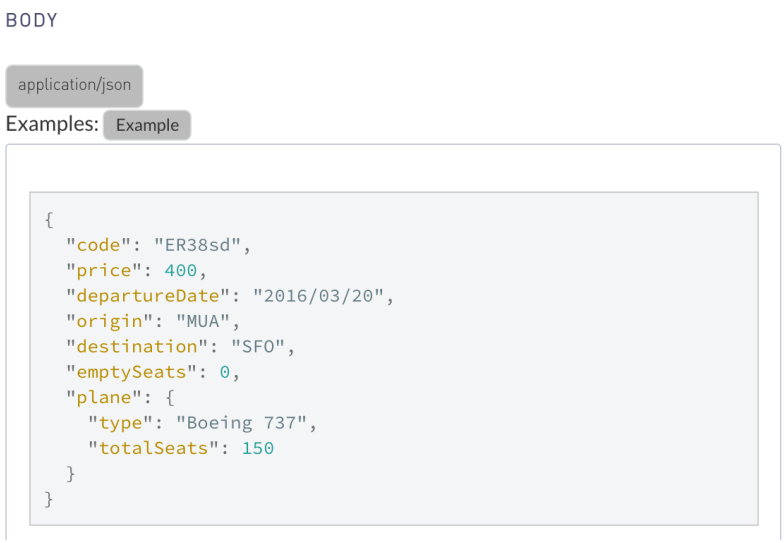
```
6 ▼ /flights:
7 ▼   get:
8 ▼     queryParameters:
9         code:
10          enum: [SFO, LAX, CLE]
11 ▼    responses:
```

40. Save the file.

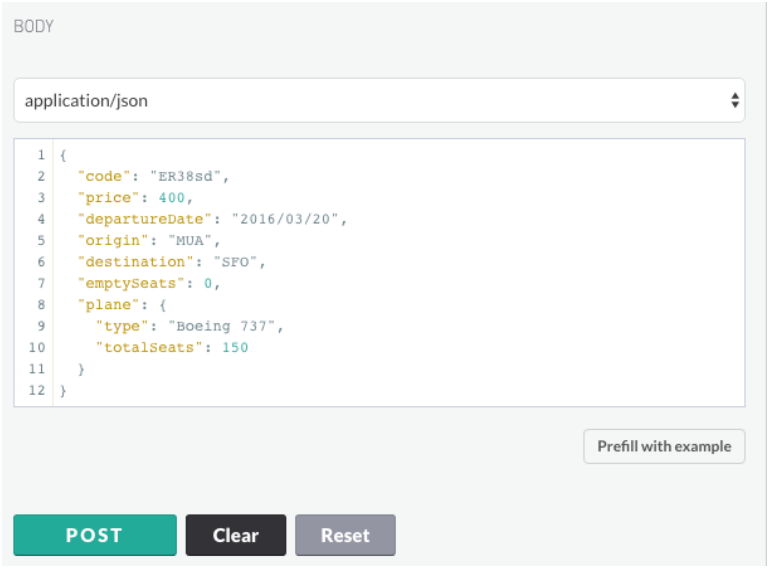
Use the sample data when making calls to the mocked API

41. In the API Console, click the GET tab for the /flights resource again.
42. Click the Try it button.

43. Locate the code query parameter; it should now have a drop-down menu with a list of possible values.
44. Select a value and click the GET button; you should see all the example data still returned.
45. Scroll up and click the Close button.
46. Click the POST tab for the /flights resource.
47. Look at the new body section.



48. Click the Try it button.
49. Locate the text area for entering JSON for the body to post; it should be prefilled with the example JSON.

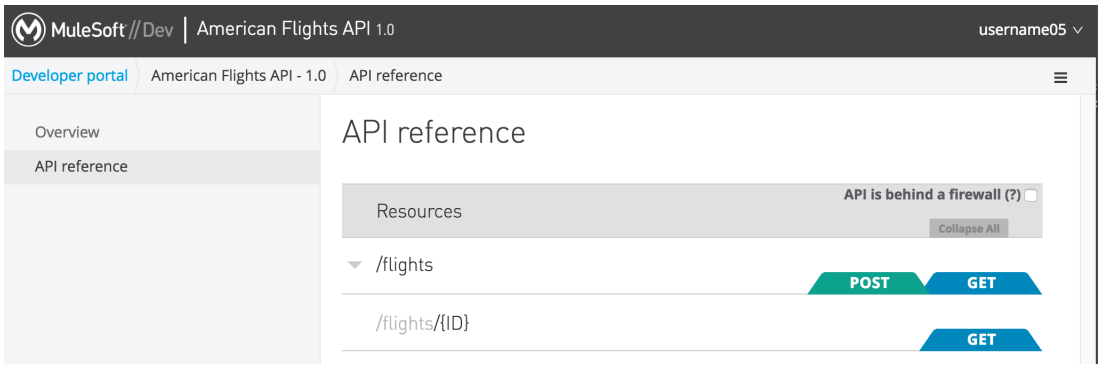


50. Click the POST button; you should get a 201 response with a message that the flight was added (but not really).

Walkthrough 2-4: Create an API portal

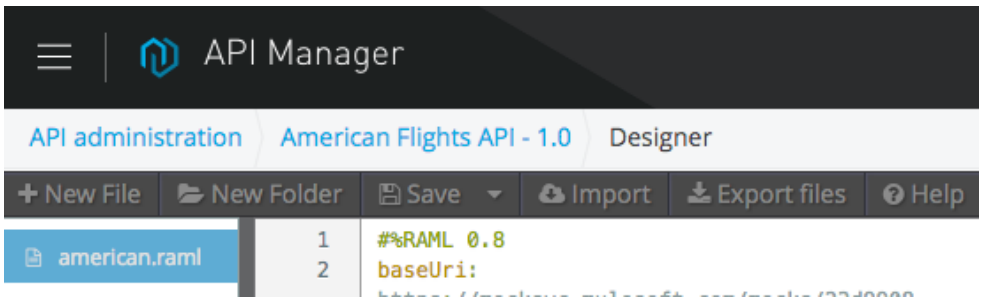
In this walkthrough, you create an API portal for developers to locate, learn about, and try out the API. You will:

- Create an API portal.
- Add content to the portal.
- Customize the portal.
- View the resulting developer portal.

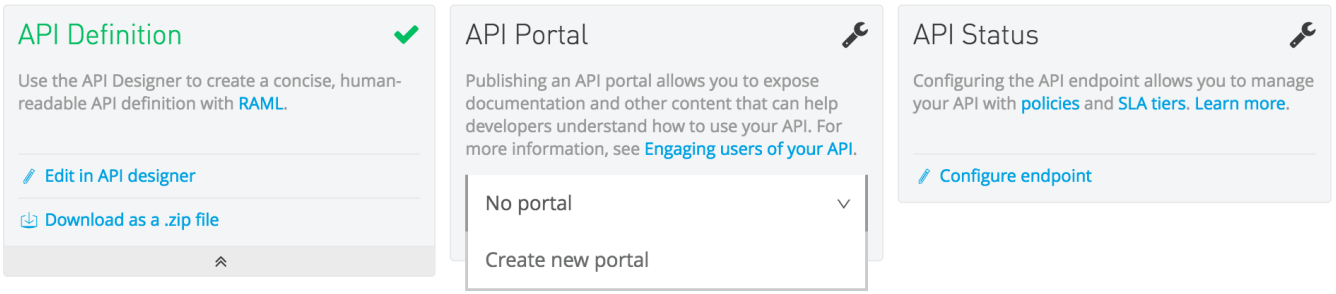


Create a portal

1. Return to API Designer.
2. Click the American Flights API – 1.0 link in the top-left corner.

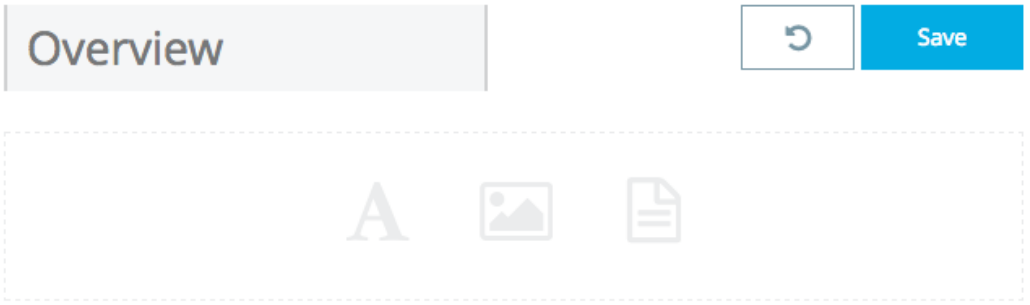


3. On the API details page, locate the API Portal section.
4. In the drop-down menu, select Create new portal; the API Portal Designer should open.

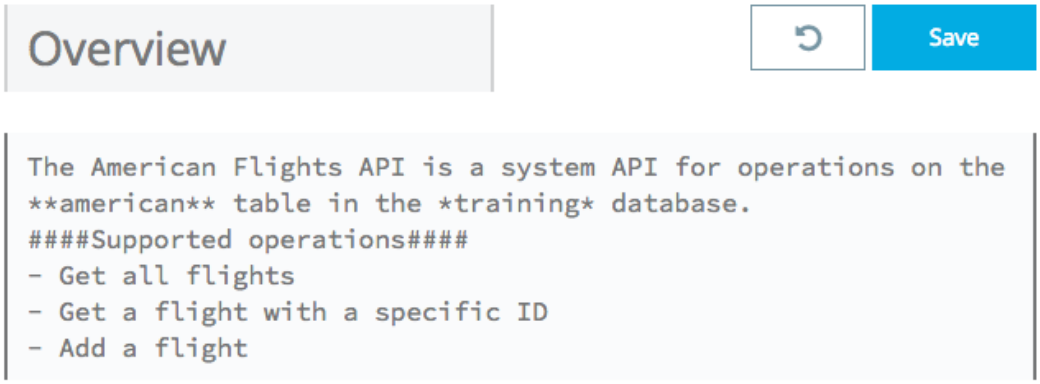


Customize the portal's home page

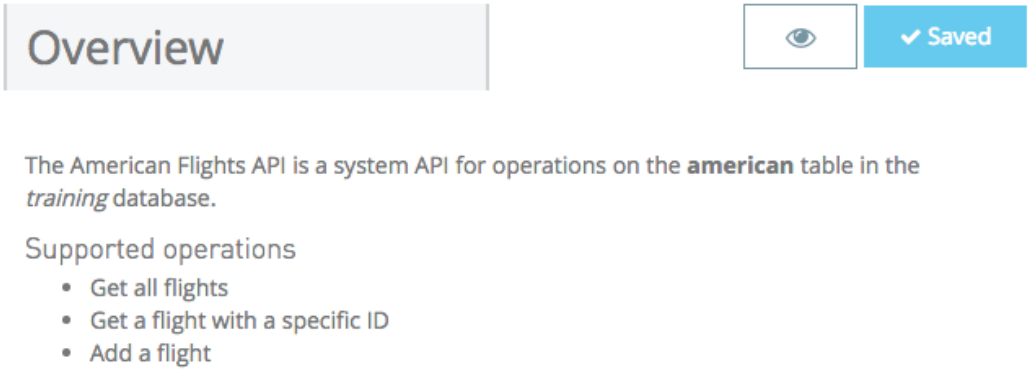
5. In the API Portal Designer, click the Home text in the page and change it to Overview.
6. Return to the course snippets.txt file and copy the text for the API Portal overview text.
7. Return to the API Portal Designer and click the A symbol in the middle of the page.



8. In the text area that appears, paste the text.
9. Look at the markdown syntax being used to format the text.

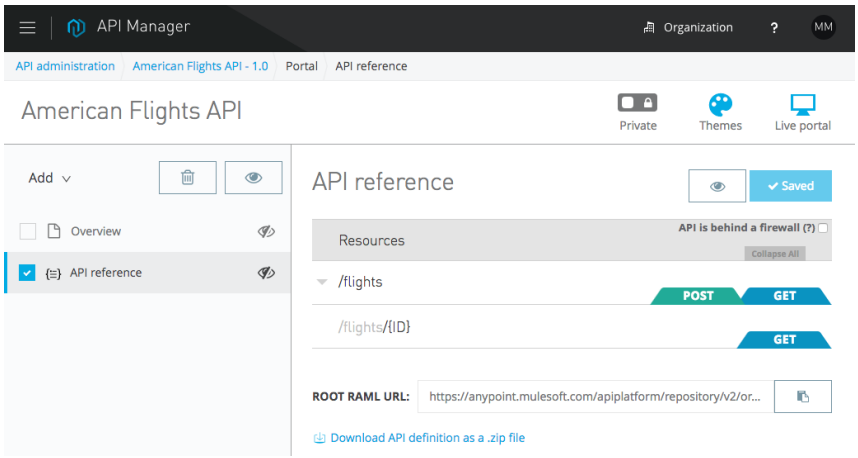


10. Click outside the text area and look at the formatted content.
11. Click the Save button.



Explore the API reference page

12. In the left-side navigation, click the API reference link.
13. Explore the API, just as you did in the API Console in API Designer.



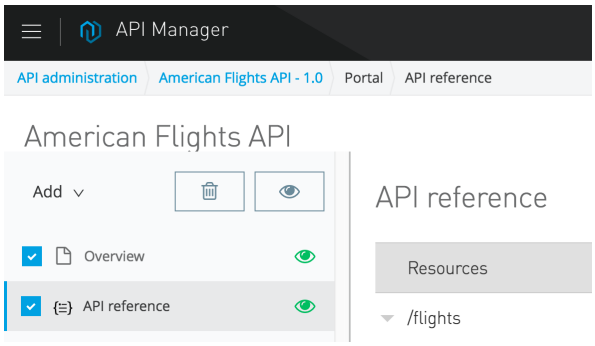
Save the RAML URL

14. Locate the ROOT RAML URL at the bottom of the API reference.
15. Click the Copy button located to the right.
16. Return to the course snippets.txt file and paste it under Your American Flights RAML URL.

```
46    * Your American Flights RAML URL
47    https://anypoint.mulesoft.com/apiplatform/repository/v2/
48
49    * Your American Flights API Portal URL
```

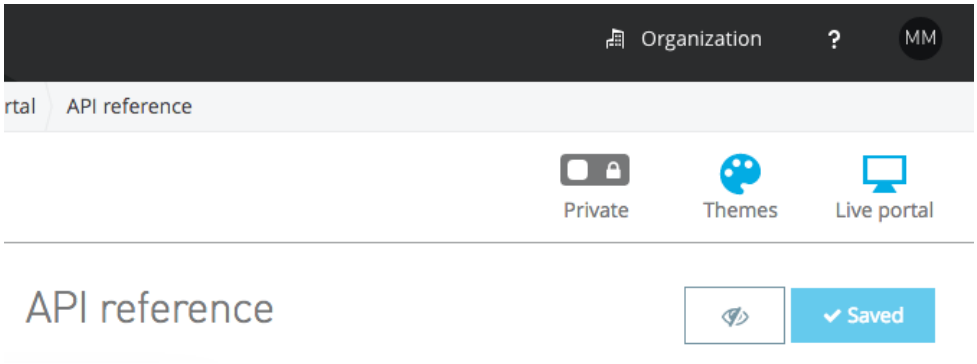
Make pages visible

17. Return to the API Portal Designer.
18. In the left-side navigation, check the checkboxes to the left of the Overview and API reference pages.
19. Click the Set to visible button (the eye); the eye icons next to both pages should now be green.

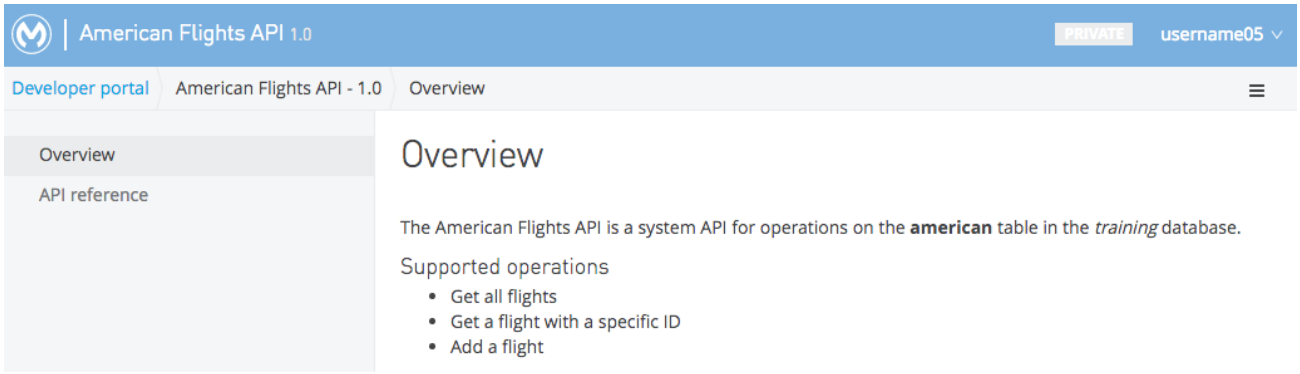


View the developer portal

20. Click the Live portal button.

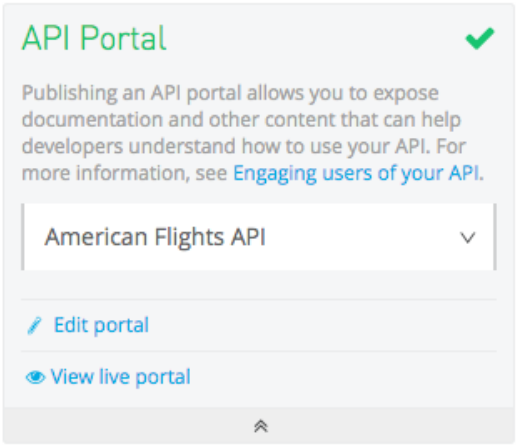


21. Examine the API Portal that opens in a new browser tab or window.



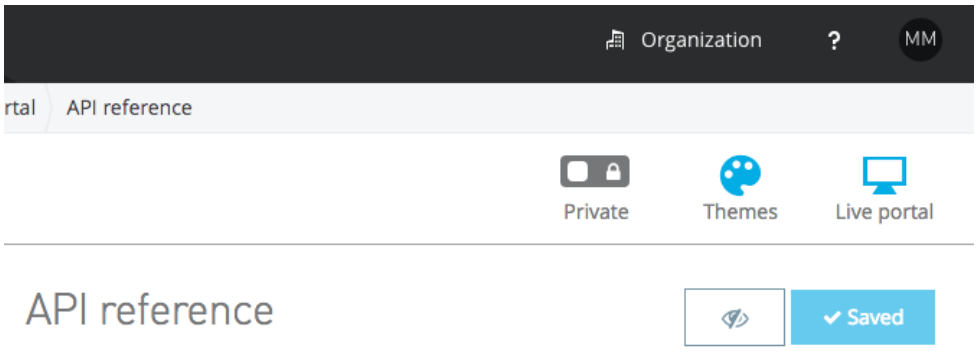
22. In the left-side navigation, click API reference.
23. Explore the API.
24. In the main menu bar, click the menu button (the three horizontal lines) located all the way to the right and select Manage API; you should return to the API Detail page in Anypoint Platform.

25. In the API Portal section, click the Edit portal link.



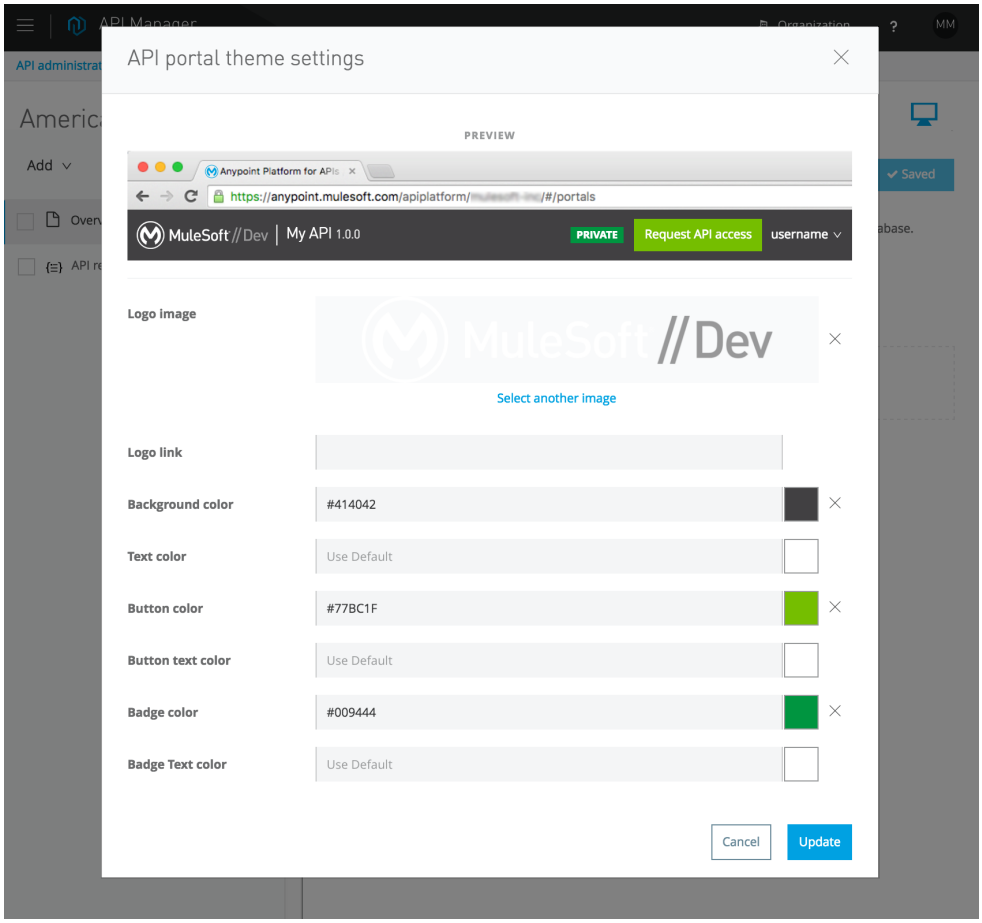
Skin the portal

26. In the API Portal Designer, click the Themes button; an API portal theme settings dialog box should open.

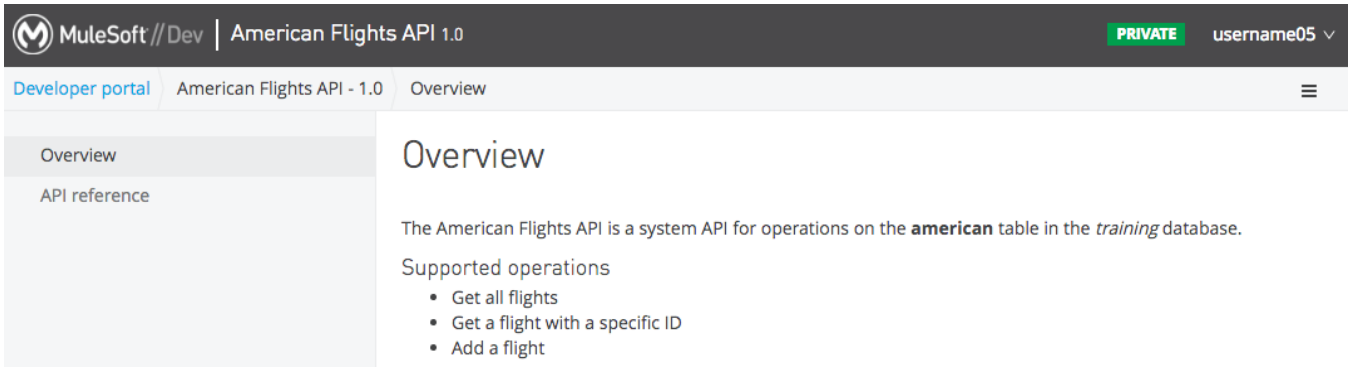


27. In your computer's file browser, navigate to the student files and locate the PNG file in the resources folder.
28. Drag the PNG file from the student files to the API Portal theme settings dialog box and drop it in Logo image area.
29. Click in the Background color text box.
30. In the color pop-up that appears, click in the color bar on the right to select a color and then click in the color box to select a shade.
31. Look at the preview at the top of the dialog box and make sure the color looks good with the logo.
32. Change any other colors you want.

33. Click the Update button.



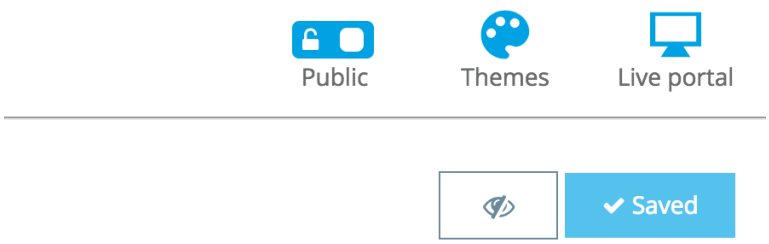
34. In the API Portal Designer, click the Live portal button again; you should see the new portal skin.



35. Look at the upper-right corner and locate the PRIVATE badge.
36. Close the tab or web browser window.

Make the portal public

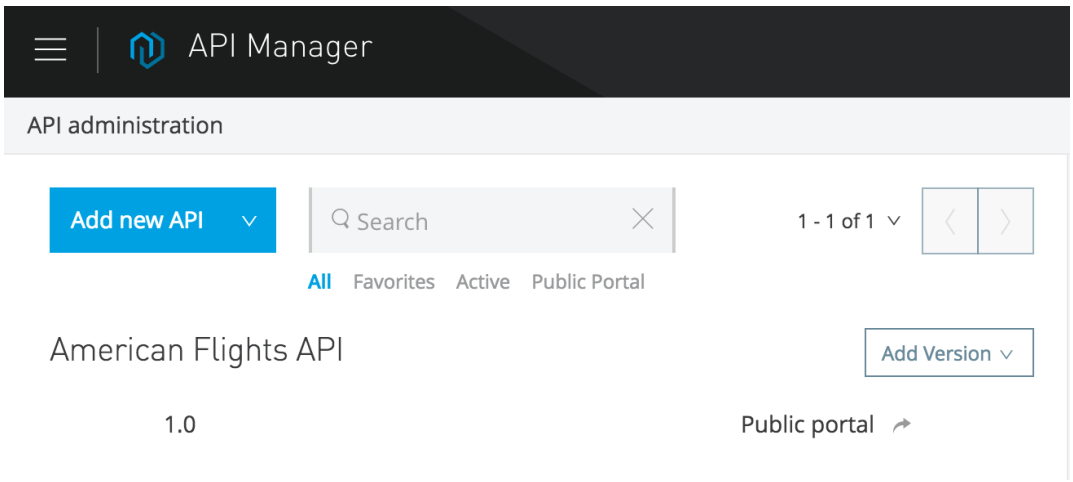
37. Return to the API Portal Designer.
38. Click the Private slider in the upper-right corner; it should now say Public.



or operations on the **american** table in the *training* database.

Save the portal URL

39. Click the API Administration link in the upper-left corner of the API Portal Designer.
40. Locate the Public portal link next to the version of the API.



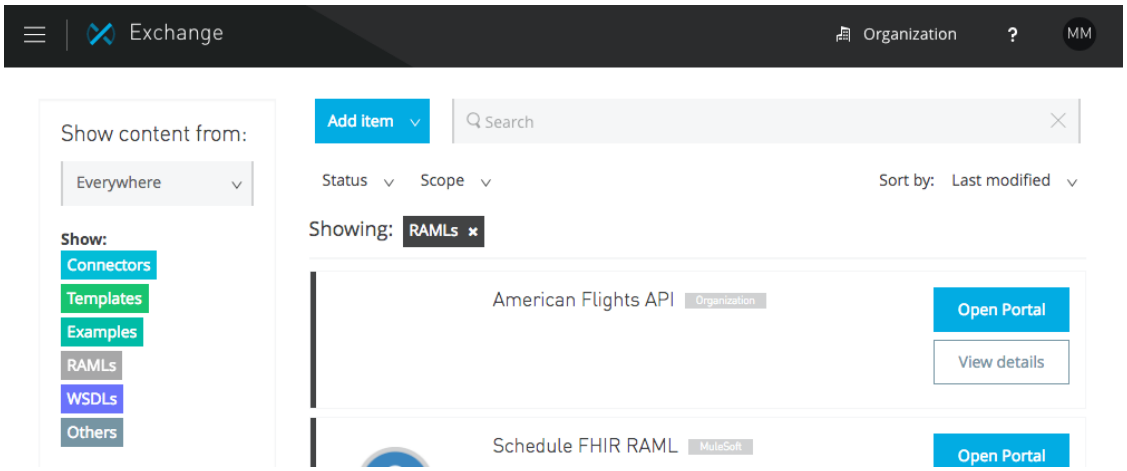
41. Right-click the link and select Copy Link Address (or your browser equivalent).
42. Return to the course snippets.txt file and paste it under Your American Flights API Portal URL.
43. Save the file.

```
46    * Your American Flights RAML URL
47    https://anypoint.mulesoft.com/apiplatform/repository/v2/orga
48
49    * Your American Flights API Portal URL
50    https://anypoint.mulesoft.com/apiplatform/organization-5/#/p
```

Walkthrough 2-5: Add an API to the Anypoint Exchange

In this walkthrough, you enhance the discoverability of an API by adding it to the private Anypoint Exchange. You will:

- Give yourself permission to publish items on the Exchange.
- Add a new RAML API to the private Exchange.
- Submit an item for approval.
- Approve and publish an item to the private Exchange.



Review private Exchange settings

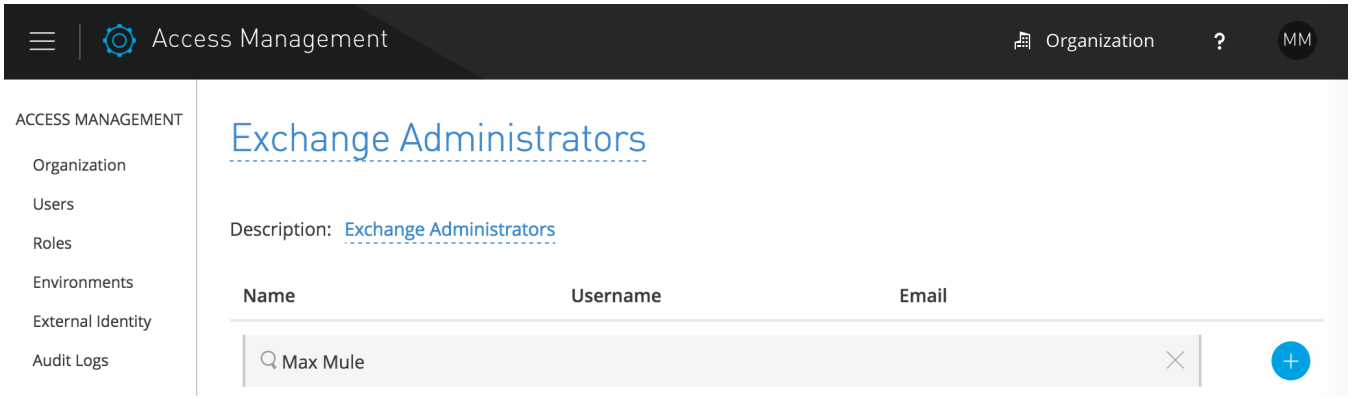
1. Return to Anypoint Platform.
2. In the main menu, select Access Management.
3. In the left-side navigation, click Exchange.
4. Review the information.

Make yourself an Exchange Administrator

5. In the left-side navigation, click Roles.
6. View the Exchange roles.

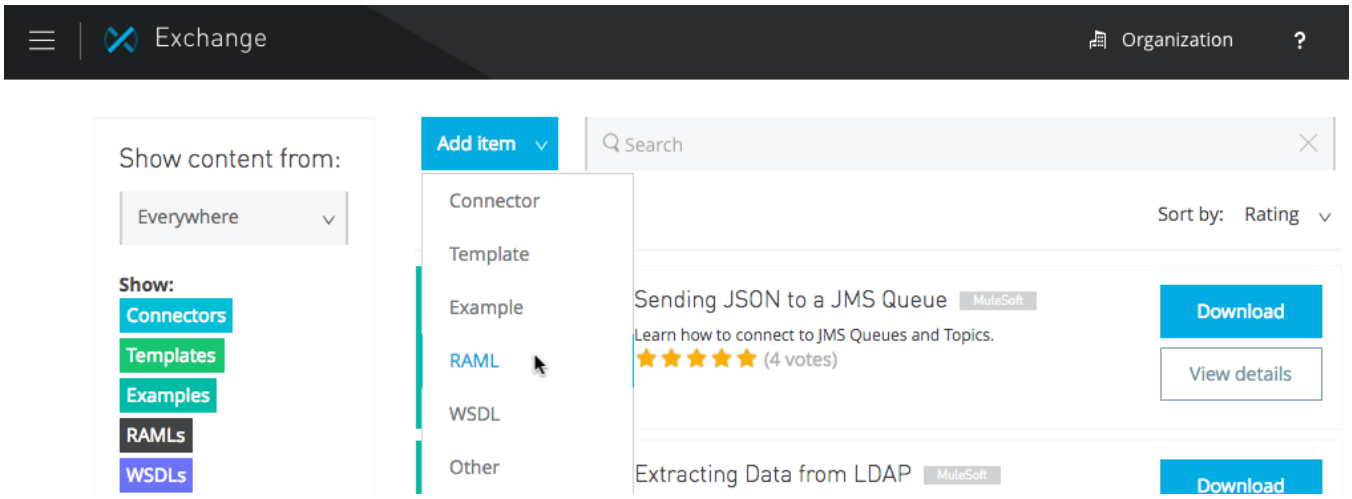
Exchange Administrators	0	Exchange Administrators
Exchange Contributors	0	Exchange Contributors
Exchange Viewers	0	Exchange Viewers
Organization Administrators	1	Organization Administrators
Portals Viewer	0	Viewer of all portals in the organization

7. Click Exchange Administrators.
8. Click in the Add a user text box and select yourself from the drop-down menu that appears.
9. Click the add button.

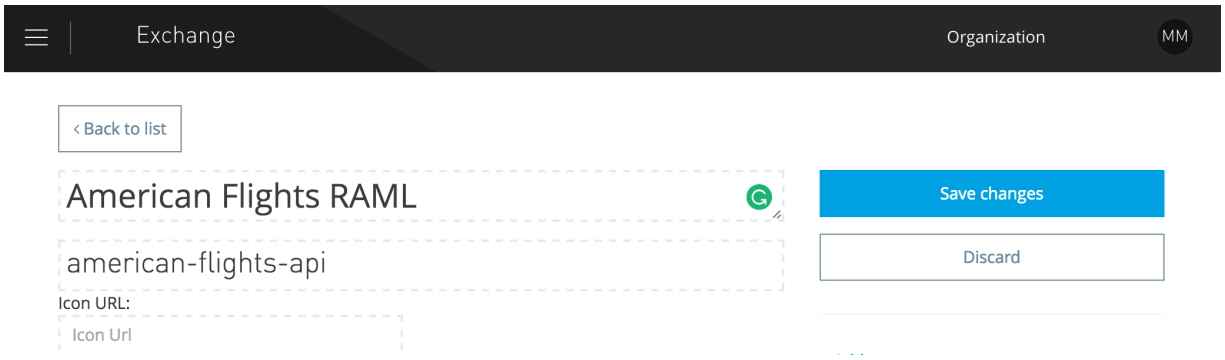


Create a new item to add to the Exchange

10. In the main menu, select Exchange.
11. In the Exchange, click the Add item button and select RAML.

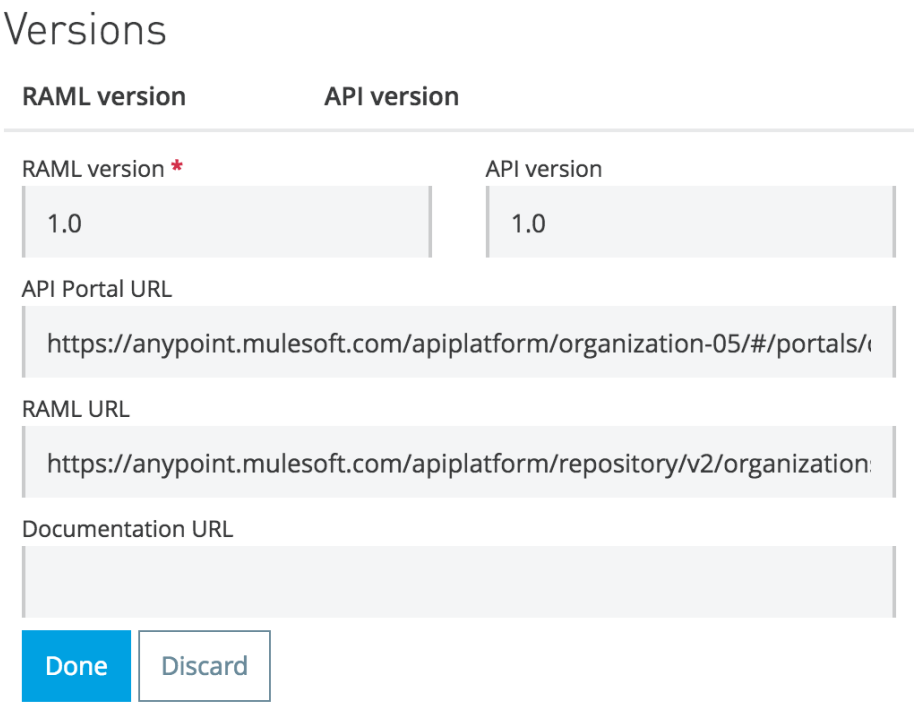


12. Set the item name to American Flights RAML.



13. Scroll down to the versions section and click the Add version button.
14. Enter the following information:

    - RAML version: 1.0
    - API version: 1.0
    - API Portal URL: *Use the URL for your portal that you saved in the course snippets.txt file*
    - RAML URL: *Use the URL for your RAML that you saved in the course snippets.txt file*

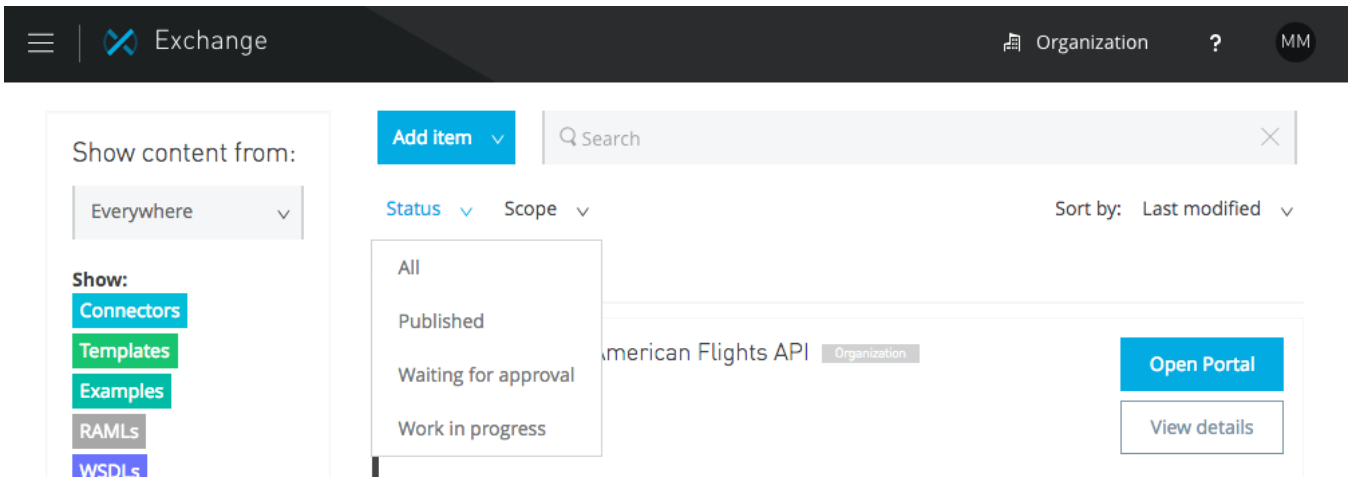


15. Click Done.
16. Scroll up and click the Save new item button.

Locate the item

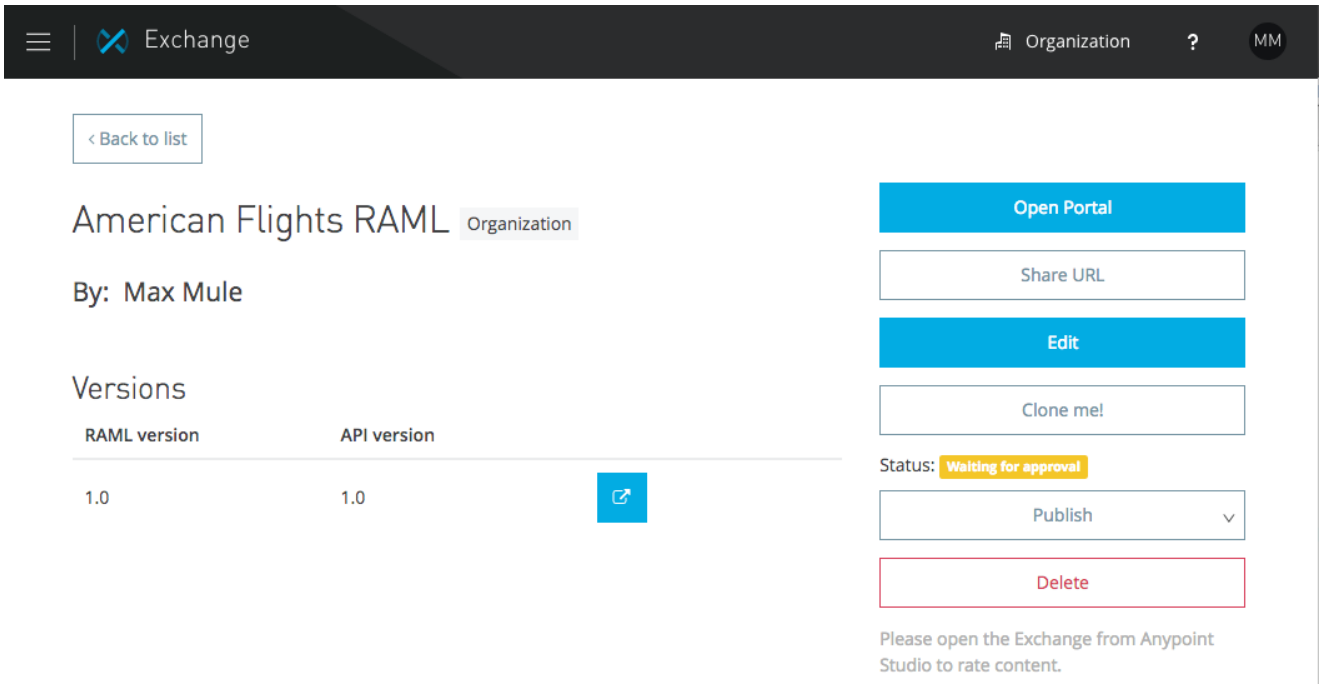
17. Click the Back to list link.
18. Click the RAMLs button in the upper-left corner of the Exchange.
19. In the Status menu beneath the Add item button, select Waiting for approval; you should see nothing listed.

20. Change the Status menu to Work in progress; you should your RAML.



Publish the item

21. Click the View details button for the American Flights RAML.
22. Locate the Work in progress status.
23. Click the drop-down button on the Publish button and select your organization.



Note: Exchange contributors would have a Request to Publish instead of a Publish button and then an Exchange administrator would have to approve the item.

View the new item in the Exchange

24. In the Status menu beneath the Add item button, select Published; you should see your RAML listed.
25. Change the Status menu to All; you should see your RAML listed.