



Module 7: Documenting and Testing APIs

Goal



☰

API summary

▼ Documentation

ACME Bank Headline

ACME Banking API Home

> Types

▼ Resources

☰

ACME Banking API enables developers to build applications that make use of the information from resource methods implemented in the API.

This API contains functionality that allows developers to retrieve and manipulate *customer*, *account* and *transaction* information.

Expires:

type: datetime

description: |

Sets a date in RFC 1123 format from which the cached resource should no longer be consi

If both the Expires header and max-age in the Cache-Control header are set, max-age wil

example: Tue, 17 Jul 2017 09:30:41 GMT

format: rfc2616

12 /customers:

13 get:

14 description: Retrieve a list of customers

15 displayName: Get all customers

16 headers:

17 Accept?:

✕ —

Objectives

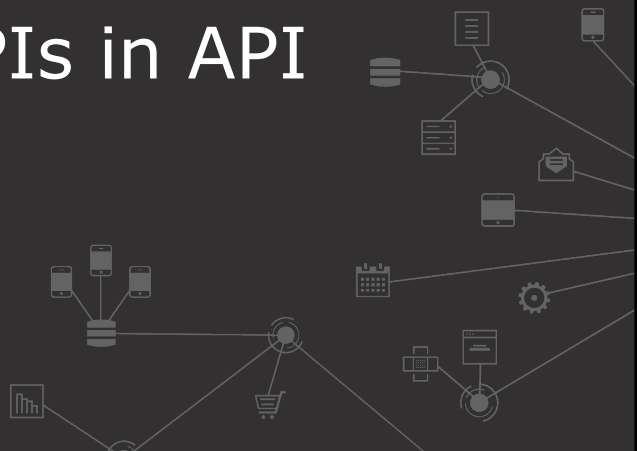


- Add documentation and description nodes to RAML definitions
- Use the mocking service to create API endpoints
- Use the API console to test API endpoints

All contents © MuleSoft Inc.

3

Documenting APIs in API designer



Documenting APIs using Anypoint Platform



- Three ways to document APIs
 - Provide documentation and description nodes in the RAML API definition
 - This will be added to the API console
 - API Portal (covered in module 8)
 - API Notebook (covered in module 8)

Creating documentation in RAML 1.0



- Documentation can be
 - Added inside the RAML API definition
 - Referenced as another RAML file outside the API definition, as a fragment
- A documentation fragment consists of a title and content node
 - Serves as a user guide and reference for the API
 - Explains how the API works by providing technical and business context
- Multiple documentation fragments can be referenced inside a RAML API definition

Walkthrough 7-1: Add a documentation fragment to a RAML API definition



- Create a documentation fragment
- Link the documentation fragment to the main RAML definition file
- Link multiple documentation fragments to the documentation node in the RAML API definition

```

1  #%RAML 1.0 DocumentationItem
2  title: ACME Banking API Home
3  content: |
4  **ACME Banking API** enables developers to build

```

☰
✕ —

ACME Banking API enables developers to build applications that make use of the information from resource methods implemented in the API.

This API contains functionality that allows developers to retrieve and manipulate *customer*, *account* and *transaction* information. Check out the [API Portal](#) for more details.

All contents © MuleSoft Inc.

7

Adding description and displayName nodes to API definitions



- A description node
 - Is a substantial, human-friendly description of a resource, method, datatype, etc.
 - Can be a string and can be formatted using Markdown
- A displayName node
 - Is an alternate, human friendly name for a resource
 - It can be used with resource methods, example and type nodes too

```

12  /customers:
13    get:
14      description: Retrieve a list of customers
15      displayName: Get all customers
16      headers:
17        Accept?:

```

All contents © MuleSoft Inc.

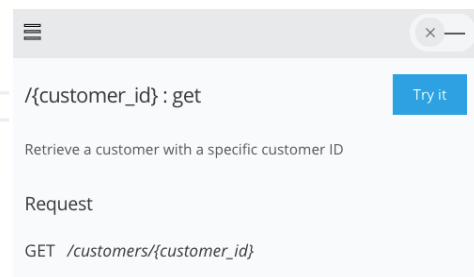
8

Walkthrough 7-2: Add description nodes to resource methods



- Add descriptions to resource methods
- Add displayName to resource methods
- Add descriptions to URI parameters
- Add description and example nodes to request and response headers

```
12 /customers:  
13 get:  
14   description: Retrieve a list of customers  
15   displayName: Get all customers  
16   headers:  
17     Accept?:
```



All contents © MuleSoft Inc.

9

Testing APIs in API designer



Enabling the mocking service



- Mocking service is a part of the API console
- Creates a mocking server to test API endpoints using the API console
- When the mocking service is turned on, a `baseUri` node is appended to the RAML API definition
 - It forms the base of the URLs of each of the API resources
 - `baseUri` can contain the mock server URL or actual API implementation URL

All contents © MuleSoft Inc.

11

Walkthrough 7-3: Use the mocking service in API console to test an API



- Enable the mocking service in the API console
- Test the resource methods in the API console to see the response information

The screenshot displays the MuleSoft API console interface. On the left, a sidebar menu includes 'API summary', 'Documentation', 'ACME Bank Headline', and 'ACME Banking API Home'. A 'Mocking Service' button is visible. The main panel shows a successful API test result with a status of '200 OK' and a response time of '334.72 ms'. Below this, the response body is displayed as a JSON object:

```

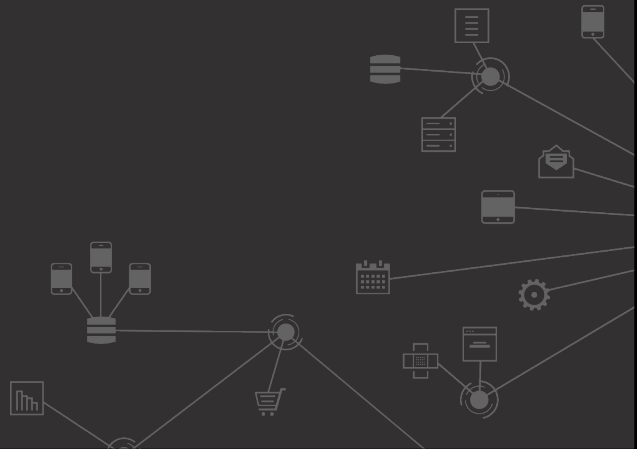
{
  "customerID": "8f19cb50-3f57-4d38",
  "firstName": "John",
  "lastName": "Doe",
  "displayName": "John Doe",
  "address(Object)": {
    "addressLine1": "1234 Lane",
    "addressLine2": "Apt.#620",
    "city": "San Francisco",
    "state": "California"
  }
}

```

All contents © MuleSoft Inc.

12

Summary



Summary



- There are several ways of documenting APIs – API Portal, API Notebook and adding documentation fragments in API definition
- Documentation added to the RAML definition appears in the API console
- Documenting allows key stakeholders to give valuable feedback during API lifecycle design
- API console and mocking service allows rapid prototyping of the API design