API URLs

Exported by obd-svc ©2018 MuleSoft, Inc.

API URLs 

The structure of a URI is central to how APIs are organised and categorised within your enterprise domain.A good URI taxonomy helps to categorise your APIs across functional domains, regions, access (public orprivate) and helps define relationships (hierarchical). A good URI also helps to govern the lifecycle of yourAPI through versioning practices.

Recommended URI Structure:

Part Description Example

| Optional. The API environment. An API could be available in a sandbox environment toenable developers to test that API. The {env} part is excluded for production APIs. |
| --- |
| Optional. The access level of the API. This could be public or private. By default the{access} part is excluded for public APIs or simply set to "api". |
| Required. The name of the company or business division for private services. |
| Required. The region of the API |
| Required. The name of the API as defined in the API Manager. This typically presents thebusiness service and should be a short but descriptive name. |
| Required. The version of the API. Depending on requirements, the version can reflect onlymajor versions or include a more hierarchical convention to identify minor versions. |
| Required. The name of the resource that represents the actual object. An API may containmultiple resources. The resource can also be referred to as the API endpoint. |
| Optional. The id of the resource to be fetched/updates. The resource id is optional. |

{env}sandbox

{access}

{company}

{region}

{context}

{versio

api

mytaxis.co.uk

quickbooker

v1

n}

{resou

bookin

rce} gs

{resou

rce-id}

{query

1981927

Optional. The query string can define state transition parameters. page=1

param s}

&sort=+<field>

Exported by obd-svc - 2 - ©2018 MuleSoft, Inc.

API URLs Setting the Base URI 

1 Setting the Base URI

Once the APIs URLs are established, the baseURI defined in the RAML should reflect the DNSentires: http://[env].[access].[company].[region]/[context]/[version] as described in the table above.

Exported by obd-svc - 3 - ©2018 MuleSoft, Inc.

API URLs Filtering 

2 Filtering

In some cases, the API consumer might only need a subset of a collection of resources. This could beaccomplished by using query parameters. For example, to get the list of all shipped orders, the API consumer could use:

GET /orders?state=shipped

Here, the "state" query parameter is used to filter the response.

Exported by obd-svc - 4 - ©2018 MuleSoft, Inc.

API URLs Sorting 

3 Sorting

Similar to filtering, a generic query parameter sort could be used to describe sorting rules. To allowsorting on multiple fields, the query parameter could be designed to take a list of fields instead of a singlevalue.

Next, to allow for ascending and descending sort order, the query parameter could take minus (“-“) as aprefix of each field.

For example, the following request will return all purchase orders sorted by date (descending) and then byproduct (ascending):

GET /orders?sort=-date,product

Exported by obd-svc - 5 - ©2018 MuleSoft, Inc.

API URLs Partial Resources 

4 Partial Resources

In some cases, the consumer might not need all the fields of a resource. To allow for obtaining only apartial resource the API URL could be designed to take a list of fields as a query parameter, and returnonly the fields that are included in that list.

For example, the following request will return only the date and the total of the purchase order: GET /orders/1?fields=date,total

4.1 Aliases

To make the experience of using an API more pleasant for the application developers, the API couldpackage a set of conditions into an easily accessible URL. For example, to return the recently shippedorders, the API could provide the following endpoint: *GET /orders/most-recent*

A resource name should remain short in order to avoid any size limitations.  The base URL should alsocontain no more than 2-3 resources if possible. URIs can be limited in some HTTP stacks.

Exported by obd-svc - 6 - ©2018 MuleSoft, Inc.