**API Design Review Checklist**

Contents

[URI Format 2](#_Toc17724765)

[URI Base Path 2](#_Toc17724766)

[API Versioning 2](#_Toc17724767)

[Resource Naming 2](#_Toc17724768)

[Resource Identification 2](#_Toc17724769)

[Modeling 3](#_Toc17724770)

[Common Objects 3](#_Toc17724771)

[Request and Response 4](#_Toc17724772)

[Verb Usage 4](#_Toc17724773)

[Specification Completeness 5](#_Toc17724774)

[Business Reporting Considerations 5](#_Toc17724775)

[Architecture/Call Sequence Diagram 5](#_Toc17724776)

[API Certification 5](#_Toc17724777)

[API Naming 6](#_Toc17724778)

**Authors: APIFIRST Team**

**Digital Architecture**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version & Date** | **Created/Changed by** | **Change details** | **Reviewed/Approved By** |
| **1.0/ July 17, 2019** | **Maneesh Moota** | **Initial version** |  |
| **1.1/ Aug 21, 2019** | **Melissa Johnston** | **Enriched** |  |
| **1.2/Aug 26, 2019** | **Maneesh Moota** | **Added API Naming rule** | **Hirok Choudhury** |

### URI Format

|  |  |  |
| --- | --- | --- |
| **Requirements** | **Mandatory** | **Status** |
| Anything in the URI should use lowercase only |  |  |
| Use hyphens, avoid underscore |  |  |
| No forward slash (/) at the end |  |  |
| No file extensions such as xml or json |  |  |
| Query parameters’ names should not contain spaces or underscores. Use dash (-) instead |  |  |

### URI Base Path

|  |  |  |
| --- | --- | --- |
| **Requirements** | **Mandatory** | **Status** |
| Define base path using common terms so that navigation to the resource is intuitive, e.g. <https://api.se.com/sales/v1/quotes>  where “/sales/v1” is the base path in this example. |  |  |
| Limit base path qualifiers to at most 2 levels e.g. /capability/[sub-process] or /capability/[business-unit] |  |  |

### API Versioning

|  |  |  |
| --- | --- | --- |
| **Requirements** | **Mandatory** | **Status** |
| Identify the major.minor version in the OAS Version field |  |  |
| Include only the major version in the base path. Locate it at the end. e.g “/sales/v1” |  |  |
| Increment the major version if there is a “breaking change” requiring a consumer-side upgrade |  |  |
| Increment the minor version if there is an enhancement where a consumer-side upgrade is optional |  |  |
| Do not maintain more than 3 major versions of an API, best practice is to limit it to 2 |  |  |

### Resource Naming

|  |  |  |
| --- | --- | --- |
| **Requirements** | **Mandatory** | **Status** |
| Use Noun, not verb |  |  |
| Use Plural, not singular |  |  |
| Use Hyphens, not underscores |  |  |

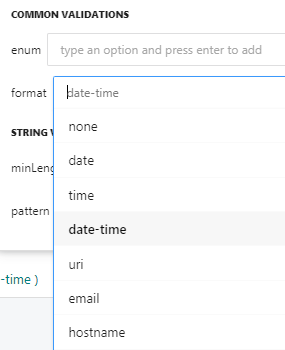
### Resource Identification

|  |  |  |
| --- | --- | --- |
| **Requirements** | **Mandatory** | **Status** |
| Use a path parameter rather than a query parameter to identify an individual resource |  |  |
| Limit number of path parameters to at most 2 e.g. /quotes/{quote-id}/quote-items/{quote-item-id} |  |  |
| Each resource definition must include a required field which is its unique identifier, typically a technical id. |  |  |

### 

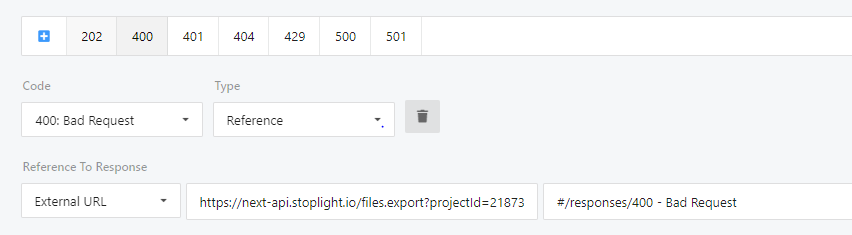
### Modeling

|  |  |  |
| --- | --- | --- |
| **Requirements** | **Mandatory** | **Status** |
| Use object modelling to define the structure of resources |  |  |
| Use simple nouns as object names, avoiding RPC-style endings such as “input”/”output” or “request”/”response” |  |  |
| Use camelCase for attribute names |  |  |
| Use the OAS Array type linked to an individual $ref object rather than defining a named array object. |  |  |
| Enable AViD format validation for fields that hold date, datetime, time, email, hostname or uri values |  |  |



### Common Objects

|  |  |  |
| --- | --- | --- |
| **Requirements** | **Mandatory** | **Status** |
| Use AViD “Reference” type to link each individual error response type to the related error object in the shared common.oas2 file via an External URL. |  |  |



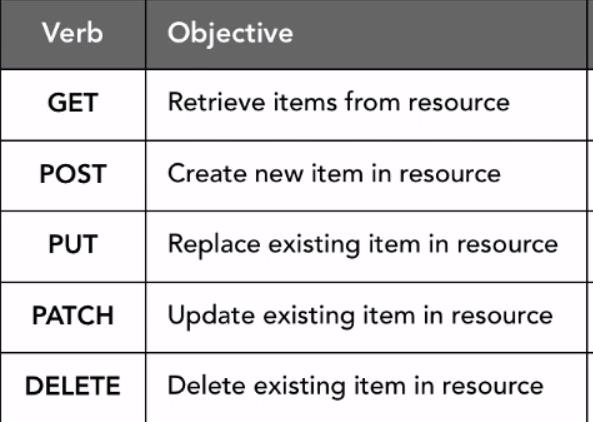
|  |  |  |
| --- | --- | --- |
| **Requirements** | **Mandatory** | **Status** |
| Use AViD “Reference” type to link to standardized common objects in the shared common.oas2 file via an External URL. For example, Currency, Address, Contact, Link and Page Links. |  |  |

### Request and Response

|  |  |  |
| --- | --- | --- |
| **Requirements** | **Mandatory** | **Status** |
| POST, PUT or PATCH must have a defined request body |  |  |
| POST must return status code 201 when a resource has been successfully created. |  |  |
| GET must have a defined response body associated with status code 200 |  |  |
| Limit response error codes to the subset defined in AViD “Common Resources” project, common.oas2 file |  |  |
| In the relevant API operation description, describe any API-specific response error code and error message value that differs from the default descriptions. |  |  |

### Verb Usage

|  |  |  |
| --- | --- | --- |
| **Requirements** | **Mandatory** | **Status** |
| Use the correct verb for the intended objective |  |  |



### Specification Completeness

|  |  |  |
| --- | --- | --- |
| **Requirements** | **Mandatory** | **Status** |
| Add API-specific descriptions at all levels: API overview, headers, parameters, operations, objects, attributes |  |  |
| Add examples for each object and operation |  |  |

### Business Reporting Considerations

|  |  |  |
| --- | --- | --- |
| **Requirements** | **Mandatory** | **Status** |
| Include fields which will help business users understand API-based reporting, for example include Account Name, Resource Name/Number (as distinct from technical id), Resource Status, Currency amounts. |  |  |
| Identify the subset of fields within the API that have business analytics significance.  Note that statistics for these fields will be collected by the APIM platform for reporting purposes. |  |  |

### Architecture/Call Sequence Diagram

|  |  |  |
| --- | --- | --- |
| **Requirements** | **Mandatory** | **Status** |
| Architecture/Call sequence diagram created and added to Technical Documents? |  |  |

### API Certification

|  |  |  |
| --- | --- | --- |
| **Requirements** | **Mandatory** | **Status** |
| API certification is mandatory for an external facing API and/or an API which has applied for a product label. Passing certification is a pre-requisite for a Production release. |  |  |

### API Naming

|  |  |  |
| --- | --- | --- |
| **Requirements** | **Mandatory** | **Status** |
| API name should not contain any System of record and technical implementation details. No references to Backend. It is up to the Business team, PM and the API Owner to come up with an API name. |  |  |