MADAPI Downstream System API Standardization

Restful Standards   
Reference:   
https://restfulapi.net/  
<https://restfulapi.net/rest-api-design-tutorial-with-example/>  
  
  
REQUESTS

* **GET** requests should only be used to fetch resources. Resources in this context apply to all manner of object references or abstractions of data that exist on the downstream including customers, products, documents etc.
* **GET** parameters like search parameters should be available as either query parameters or headers.
* **POST** requests should only be used to create resources. If there are cases where a request performs 2 actions of create, replace or modify, the primary function of the endpoint or use should determine whether a post is used.
* **PATCH** requests like POST should be used to modify existing resources.
* **PUT** requestsshould be used to replace existing resources.
* **DELETE** requests should be used to remove resources.
* Request parameters must ensure consistency in typography either CamelCase or snake\_case. This applies also to headers and query parameters.

RESPONSES

* **Two approaches** may be used.
* **Approach 1:** Requestsmust return standard http responses for successes, pending actions and failures. Successes and pending actions are usually represented with 2xx http status codes. These are to be clearly communicated in documentation. Failures are represented with 3xx – 5xx codes.
* **Approach 2:** All Successes and failures on accessing or manipulating resources are sent with http status code 2xx, however the payload of the response must include codes (success and failure) which indicate the actual response. A mapping document must be provided alongside this. 5xx and 4xx codes in this case will be used to represent failures on either accessing the downstream in itself or during service unavailability service.
* Response parameters must ensure consistency in typography either CamelCase or snake\_case. This applies also to header parameters.
* Relevant response payload should be sent at all times. Do not overload the response payload with data not relevant to the Use Case.

ERROR RESPONSES

* Error messages and descriptionsmust be representative of the actual failures. Suggested fixes or solutions can be included in the description. E.g. TransactionId Length exceeds allowable limit of 20 alphanumeric characters.

RESOURCE URIs

* Relationships between resources must be clearly expressed in the resource uri used in accessing those resources.
* e.g. https://baseurl.com/v1/customers/{customerId}/subscriptionse