

API Design - Best Practices

An effective API Design will have the following characteristics:

- Should be easy to read and work with (have names that are easy to remember).
- Should be difficult to use incorrectly. This is achieved by providing informative feedback to the user and doesn't impose strict rules on the end user.
- Any API should be complete and concise, ie., it should be possible to build complete applications using it. This is achieved by building an API incrementally.

A resource is an object that is important enough to be referenced to itself. A resource has data, maybe relations with other resources and methods that work with it. Resources are grouped together as collections. A URL points to a resource / collection.

- The base URL should be a noun. If it refers to a resource, it should be singular and if it refers to a collection, it should be plural.
- The noun used should describe the resource / collection.
- Having nouns makes it semantically meaningful to use HTTP methods with resources. Eg - get(/photos/img1)

Responses:

- The API methods must provide descriptive and meaningful messages for both validation and erroneous events. This will lead to easier debugging in case of errors and encouragement in case of successes.
- The errors should be aligned with HTTP error codes.
- All API endpoints that return a GET response should have such a sample response provided in the documentation. This sample response should include some sample data and data type.