**Project Title:**

Build a simple RESTful API web service using Python Flask and Swagger UI

**Motivation:**

First things first, A RESTful API is an application program interface ([API](https://searchapparchitecture.techtarget.com/definition/application-program-interface-API)) that uses HTTP requests to GET, PUT, POST and DELETE data. A RESTful API -- also referred to as a RESTful web service or REST API -- is based on representational state transfer ([REST](https://searchapparchitecture.techtarget.com/definition/REST-REpresentational-State-Transfer)) technology, an architectural style and approach to communications often used in [web services](https://searchapparchitecture.techtarget.com/definition/Web-services) development. The REST design does not require a specific format for the data provided with the requests. In general data is provided in the request body as a [JSON](http://en.wikipedia.org/wiki/JSON) blob, or sometimes as arguments in the [query string](http://en.wikipedia.org/wiki/Query_string) portion of the URL.

|  |  |  |
| --- | --- | --- |
| **HTTP Method** | **Action** | **Examples** |
| GET | Obtain information about a resource | http://myRetail.com/api/products (retrieve products list) |
| GET | Obtain information about a resource | http:// myRetail.com/api/product/123456 (retrieve a product with id = 123456) |
| POST | Create a new resource | http:// myRetail.com/api/products (create a new product, from data provided with the request) |
| PUT | Update a resource | http:// myRetail.com/api/products/123456 (update a product with id = 123456, from data provided with the request) |
| DELETE | Delete a resource | http:// myRetail.com/api/products/123456 (delete a product with id = 123456) |

This project adheres to the REST guidelines then becomes an exercise in identifying the resources that will be exposed and how they will be affected by the different request methods.

The goal for this exercise is to create an end-to-end Proof-of-Concept for a products API, which will create a RESTful service that can retrieve product and price details by ID, and aggregate product data from multiple sources and return it as JSON to the caller.