**Introduction to API**

API (Application Programming Interface) is a software intermediary which is used for communication between two applications. Examples of this are present everywhere online, and an API is being used every time you check the weather, use a social media application, and send an instant message.

For instance, let's say you wish to check the weather for a certain area. The information for this weather is stored on the server, along with the information of weather for every other area. You need to communicate to the server and inform it what information you want to see from its vast banks. This is where the API comes in. You tell the API, your intermediary, what area you want the weather for, and it will go to the server to request that information. It then delivers the server's response, and presents you with the weather in an easily readable way.

**REST**

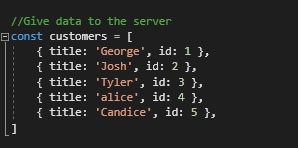
REST (Representational State Transfer) is an architectural approach to designing web services. As such, it is not a protocol or standard, and can be implemented in a multitude of ways. A RESTful API uses a representation of the state of a resource, which it transfers to the requester.

There are several criteria which are required for an API to be considered RESTful. They are as follows:

* A client-server architecture, with requests managed through HTTP (Hyperlink Text Transfer Protocol).
* Stateless client-server communication. No requests are connected, and no client information is stored between them.
* Cacheable data that streamlines client-server interactions
* A uniform interface between components so that information is transferred in a standard form. This requires:
  + Resources are identifiable and separate from the representations sent
  + Resources can be manipulated by the client, and the representation contains enough information to do so
  + Self-descriptive messages returned to the client have enough information to describe how the client should process it (For instance, understandable error feedback.)
  + Hypertext/hypermedia is available, meaning that hyperlinks can be used by the clients to find all available actions they may take
* A layered system that organizes each type of server involved in the retrieval of requested information. This is invisible to the client.

**The Application**

This is a simple application demonstrating usage of a REST API with CRUD (Create, Read, Update, and Delete), and tested using Postman.

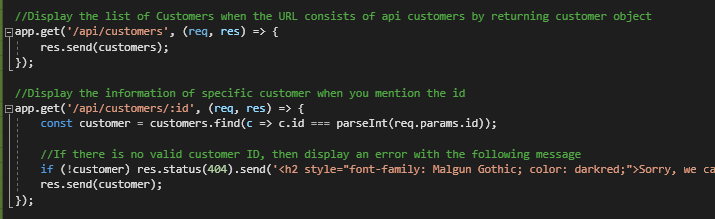


The application first creates an object “customers” which gives data to the server. When a client needs information from this object, the API will request it from the server, and get the current state of the customers object.

Using the URLs chosen, the following code will show information associated with said URL, and encapsulates the “Read” functionality. When the URL is visited, the API will automatically request the information to display to the user, in this case, the list of customers.

Within the second URL, the “:id” indicates that any id within the customers object could be referenced, and the information returned. For example “localhost/api/customers/1” would return “George”.

Additionally, this URL has a special error message if the id is not found, which would happen if the client attempted to enter an id which does not exist, such as id 6. This is important as it ties into the program being considered RESTful, and providing adequate information to the client.



* The functionality of Create, Update, and Delete work similarly to what is shown above, but instead use the “Post”, “Put”, and “Delete” HTTP methods, as opposed to “Get”.