On Boarding Document
Created by Muhammad Shakir

What is API?
It is an engine under the hood and behind the scenes that we used to transfer data from one another. It is basically a messenger that takes requests and tells a system what you want to do and returns a response back to the client.

What is the HTTP method?
HTTP defines a set of request methods to indicate the desired action to be performed for a given resource. There are several Http Requests that can be used while creating a web service.

1. **GET**: The GET method requests a representation of the specified resource. Requests using GET should only retrieve data.
2. **POST**: The POST method is used to submit an entity to the specified resource, often causing a change in state or side effects on the server.
3. **PUT**: The PUT method replaces all current representations of the target resource with the request payload.
4. **PATCH**: The Patch method is used for making the Partial changes to an existing resource.
5. **DELETE**: The DELETE method deletes the specified resource.

Difference Between PUT and PATCH.
when updating a single field of the Resource, sending the complete Resource representation might be cumbersome and utilizes a lot of unnecessary bandwidth. In such cases, the semantics of PATCH make a lot more sense.

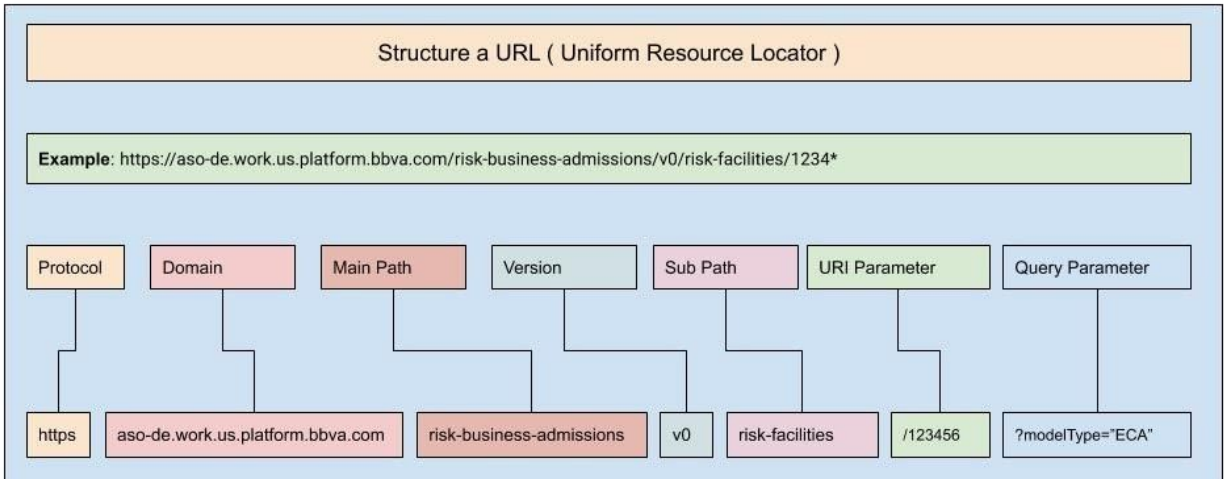
Difference Between URI Parameters and Query Parameters.
URI Parameter is basically used to identify the specific resource or resources whereas Query parameter is used to filter the resource.

For Instance, You want to identify the ARCE Team member based on XSID in that case we should use URI Parameter because each team member has a specific XSID. Now, consider another example where you want to filter all the employees based on the occupation, and in that case we should Query Parameters.

Example of URI Parameter: https://www.arce.com/teams/xs10000
Example of Query Parameter: https://www.arce-team.com/teams?occupation=javadeveloper

How to Structure a URL?

A uniform resource locator is a reference to a web resource that specifies it is location.



What are the Different Data Types?

A data type, in programming language, is a classification that specifies which type of value a variable has and what type of mathematical, relational, logical operations can be applied to it. A String, for example, is a data type that is used to classify text and integer is a data type used to classify whole numbers

In Java Data types are divided into two groups Primitive and Non-Primitive Data Type.

Primitive Data Type	Non-Primitive Data Type
A primitive data type specifies the size and type of variable values, and it has no additional methods.	Non-primitive data types are called reference types because they refer to objects.
1. Byte (size: 1Byte) 2. Short (size: 2Bytes) 3. Int (size: 4Bytes) 4. Long (size: 8Byts) 5. Float (size: 4Bytes) 6. Double (size: 8Bytes) 7. Boolean (size: 1Bit) 8. Char (size: 2Bytes)	1. Array 2. String 3. Classes
Primitive number types are divided into two groups. Integer types store whole numbers. Valid types are byte, short, int and long. Floating types represent numbers with a fractional part, containing one or more decimals. Valid types are float and double.	Non-primitive types are created by the programmer and are not defined by Java (except for String).

What are the HTTP Status Codes?

HTTP response status codes indicate whether a specific HTTP request has been successfully completed. HTTP responses are grouped in five different classes.

1. Informational responses (100-199)
2. Successful responses (200 - 299)
3. Redirects (300 -399)
4. Client errors (400 - 499)
5. Server Errors (500 - 599)

For more information, Please visit here

Attention: Prerequisites of JSON are Data Types.

What is JSON? (JavaScript Object Notation).

It is a syntax for storing and exchanging or transmitting data in web applications. It is easy to understand and user friendly. It is one of the most common formats used to transfer or exchange data.

JSON objects are surrounded by curly braces {}.

JSON objects are written in key/value pairs. Keys must be strings, and values must be a valid JSON data type (string, number, object, array, boolean or null). Keys and values are separated by a colon. Each key/value pair is separated by a comma.

Example of Json.

```
{
  "data": {
    "fullName": "Muhammad Shakir",
    "age": 28,
    "hasJob": true,
    "address": {
      "street": "924 Griffin Ave, Gadsden Alabama",
      "zipCode": 35903,
    }
    "experience": [
      {
        "profession": "delivery driver",
        "duration": "2 Years”
      },
      {
        "profession": "front-end developer",
        "duration": "1 Year”
      }
    ]
  }
}
```

What is the Json Request Body?

JSONRequest is proposed as a new browser service that allows for two-way data exchange with any JSON data server without exposing users or organization to harm. It exchanges data between scripts on pages with JSON servers in the web.

What is the Json Response?

JavaScript Object Notation (JSON) is a standard text-based format for representing structured data based on JavaScript object syntax. It is commonly used for transmitting data in web applications (e.g., sending some data from the server to the client, so it can be displayed on a web page, or vice versa).

What is Header?

HTTP headers let the client and the server pass additional information with an HTTP request or response. An HTTP header consists of its case-insensitive name followed by a colon (:), then by its value.

1. **Request Header**
    a. A request header is an HTTP header that can be used in an HTTP request, and that doesn't relate to the content of the message. Request headers, like Accept, Accept-*, or If-* allow to perform conditional requests
2. **Response Header**
    a. A response header is an HTTP header that can be used in an HTTP response and that doesn't relate to the content of the message. Response headers, like Age, Location or Server are used to give a more detailed context of the response.

Example of Headers

1. Content-Type: application/json
2. Content-agent: Mozilla Firefox
3. Response: 200 OK
4. Accept Language: en-us
5. Connection: keep-alive
6. Location: www.bbva.com/risk-business-admissions/risk-facilities/16958

Difference Between Restless and Restful API?

Useful Links

JSON Formatter:
http://www.bodurov.com/JsonFormatter/
API catalog Link:
https://catalogs.platform.bbva.com/apicatalog/business/apis/apis-risks-riskbusinessadmissions/versions/v/
My Velocity:
https://velocity.bbvacompass.com/?login-required=true&redirect-to=/myVelocity/myTime