Name: Varun Chakravarthy Kanthety
Email: vc.kanthety@hotmail.com

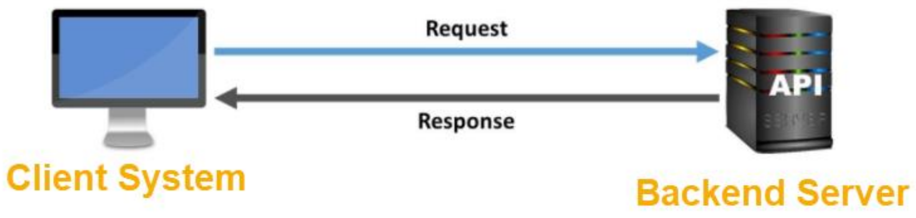
Sample API Document Using Postman and Swagger

API Overview

An application program interface (API) is a piece of code that acts as an interface between two applications. Using APIs, the applications can communicate with each other. It defines a set of protocols to request services from an application (server) and expose data within different contexts and across multiple channels. Information in different formats is shared using the API request/response process.

How API Works?

An API acts as a medium between a client and a server. When a customer wants to access the services from a server, he/she sends a request (API call) to that server. If the customer has sent a proper request, then the server responds with required services back to the customer. Here, the API passes customer request to the server and gives back the response from the server. The working functionality of API is shown in the figure below:



Example 1: Login API Demo Using Postman

Description

To access the services from the target application, users first need to authenticate themselves using their login credentials. This API call demonstrates the procedure to authenticate using "username" and "password" input parameters. Add the "login" endpoint after the base URL to send an API request.

URL

https://<ip-or-url>/pd/v1/login?username=<username>&password=<password>

HTTP Method

POST

Name: Varun Chakravarthy Kanthety
Email: vc.kanthety@hotmail.com

Request Parameters

Following parameters are required to login to the application:

1. **username**: string (Required) (Type the username)
2. **password**: string (Required) (Type the password)

JSON Response

The following response is obtained when the login request is processed successfully:

```
{
    "status": "Success",
    "code": 200,
    "data": {
        "status": "SUCCESS",
        "username": "demo",
        "tokens": {
            "ExpiresIn": 3600,
            "IdToken":"eyJraWQiOiJWVU52alRPd1B3cWRiTV
        J9kZl",
            "RefreshToken":"eyJjdHkiOiJKV1QbmMiOiJBMj
        U2wiY",
            "TokenType": "Bearer",
            "AccessToken":"eyJraWQiOiJVUHlodpSFlPSzNQ
        UFYK2"
            },
        "id": 23
        }
}
```

Response Parameters

Following response parameters are obtained from backend server:

1. **AccessToken** - A unique value provided by backend server to each authenticated user. This token value in required in further API calls to

Name: Varun Chakravarthy Kanthety
Email: vc.kanthety@hotmail.com

perform various actions. Provide this value in the **Authorization** header in 'Headers' section of API calls. This value expires after 3600 seconds (1 hr) from the time of generation.

2. **id** - A unique ID that is provided to each user. Use this ID in various API calls as required to perform user-specific actions.

Example 2: Login API Call in Swagger OpenAPI Format

```
swagger: "2.0"
info:
  description: "This is a demo API call in Swagger OpenAPI format."
  version: "1.0.0"
  title: "Demo API"
host: "ip-or-url.com"
basePath: "/pd/v1"
schemes:
- "https"
- "http"
paths:
  /login:
    post:
      tags:
      - "login"
      summary: "To login to the application."
      description: "Use this API call to login to the application to
access its services. This API call requires /login endpoint at the end
of the URL."
      consumes:
      - "application/json"
      produces:
      - "application/json"
      parameters:
      - in: "body"
        name: "username"
        type: "string"
        description: "Type the username in this field."
        required: true
        name: "password"
        type: "string"
        description: "Type the password in this field."
        required: true
        schema:
          $ref: "#/definitions/schemas/login"
```

Name: Varun Chakravarthy Kanthety
Email: vc.kanthety@hotmail.com

```
        responses:
          "400":
            description: "Bad Request"
            content:
              application/json:
                schema:
                  $ref: "#/definitions/schemas/400"
          "401":
            description: "Unauthorized"
            content:
              application/json:
                schema:
                  $ref: "#/definitions/schemas/401"
          "403":
            description: "Forbidden"
            content:
              application/json:
                schema:
                  $ref: "#/definitions/schemas/403"
          "404":
            description: "Not Found"
            content:
              application/json:
                schema:
                  $ref: "#/definitions/schemas/404"
  definitions:
    schemas:
      login:
        type: "object"
        properties:
          username:
            type: "string"
            example: "john"
          password:
            type: "string"
            example: "pass@999"
      400:
        type: "object"
        properties:
          message:
            type: "string"
            example: "Bad Request"
      401:
        type: "object"
        properties:
```

Name: Varun Chakravarthy Kanthety
Email: vc.kanthety@hotmail.com

```
        message:
          type: "string"
          example: "Unauthorized"
    403:
      type: "object"
      properties:
        message:
          type: "string"
          example: "Forbidden"
    404:
      type: "object"
      properties:
        message:
          type: "string"
          example: "Not Found"
```