**Overview:**

The application must expose restful endpoints that will parse data (passed in the request body) and return the value back to the client. The API will have two versions and depending on the version endpoint, the parsing of the data will return a different value back to the client. Use TypeScript interfaces so the code assumes the design / object properties.

**Technologies Used:**

Node js, Express, Typescript

**Tools Used:**

Visual Studio Code IDE, Postman

**Libraries Used:**

nodemon, parser-json, jest, supertest

**Application has the following components:**

Created a localhost server (port = 3000) and hosted the application.

Created a Router with two POST methods with end points /api/v1/parse, /api/v2/parse, respectively.

**Model:**

Model has Data and Result set.

**Controller:**

Created Controllers for each end point where the data is validated. If the validation is successful, the control is passed to Handler else it returns an error with status code. Controller calls Handle() of respective handlers.

Each end point has their respective handlers. Base hander is an abstract class which is implemented in both V1 and V2 handlers. It has been created for reusability.

Business logic has been implemented in Deserializers which returns response. Deserialize() of V1Deserializer and V2Deserializer is called from handlers.

**Routes:**

Routes will route the request to appropriate controller(V1controller and V2Controller).

**Process Flow:**



**Unit Testing:**

Unit testing has been performed using jest and supertest. It can be found in Tests folder in Route.test.ts file.

**Code Coverage Report:**

Code coverage report can be found in coverage folder of the project.

**Postman Test:**

Used POST() in Postman to test the request and response data.

Request

{

data: “JOHN0000MICHAEL0009994567”

}

Response for /api/v1/parse

{

statusCode: 200,

data: {

firstName: “JOHN0000”,

lastName: “MICHAEL000”,

clientId: “9994567”

}

}

Response Body for /api/v2/parse

{

statusCode: 200,

data: {

firstName: “JOHN”,

lastName: “MICHAEL”,

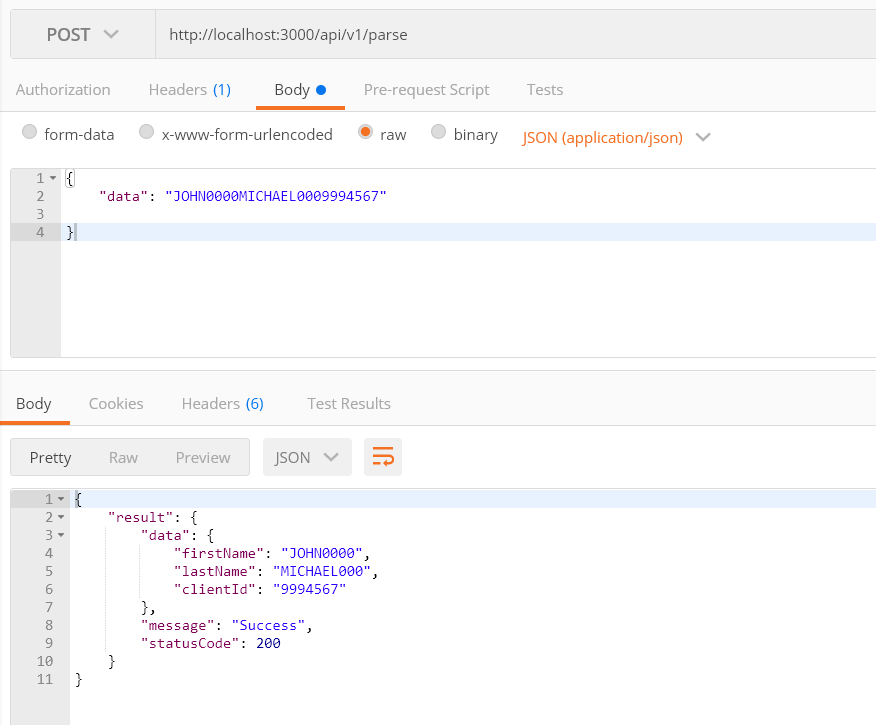
clientId: “999-4567”

}

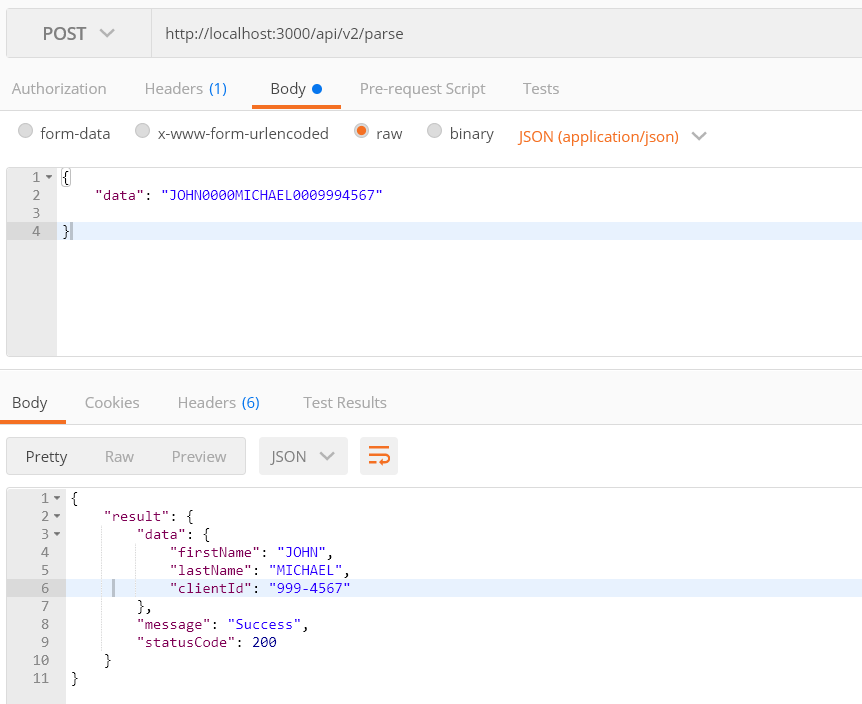
}

**Postman Request, Response Outputs:**

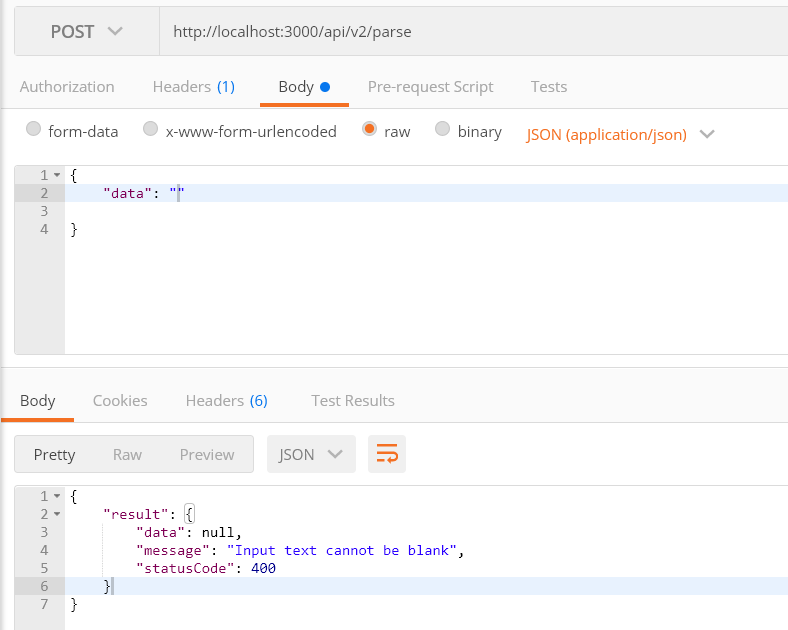
Endpoint: api/v1/parse



Endpoint: api/v2/parse



Negative Scenario:



Unit Test Results:

