EXP 2 **Working with APIs**
8/7/24

AIM:

To complete the,

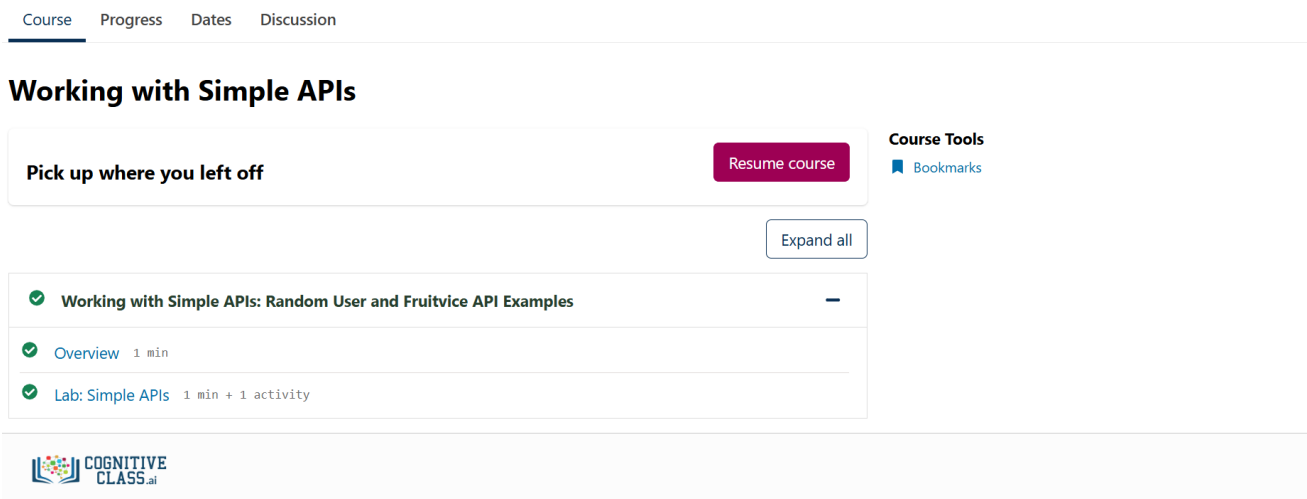
a) Guided projects in IBM skills network lab:
- Building RESTful APIs with Express
- Working with simple APIs

b) Building of a RESTful API and testing it with the Postman tool.
c) To consume a third party API in a webpage.

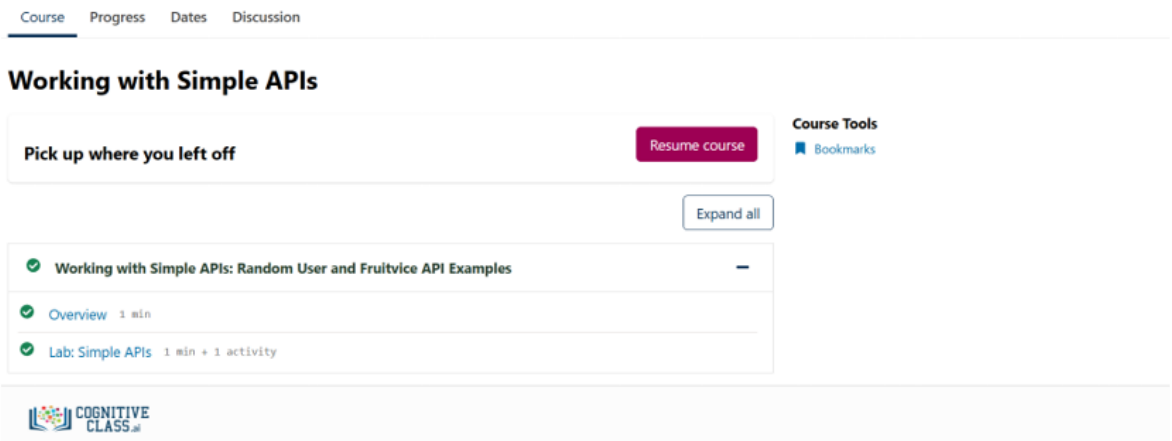
PROCEDURE:

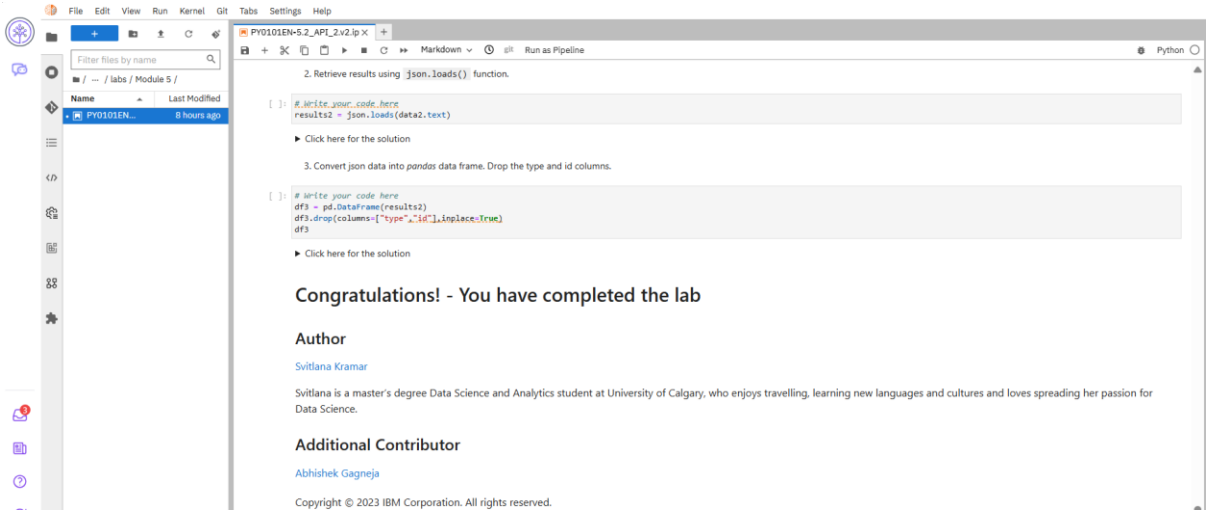
a) Guided projects in IBM skills network lab:

- Building RESTful APIs with Express.



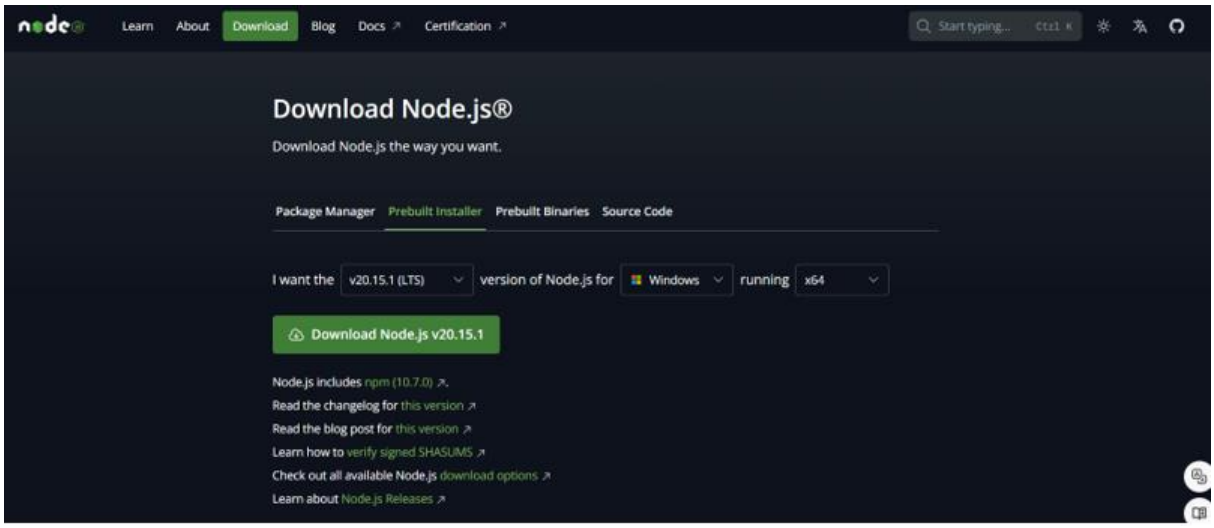
- Working with simple APIs

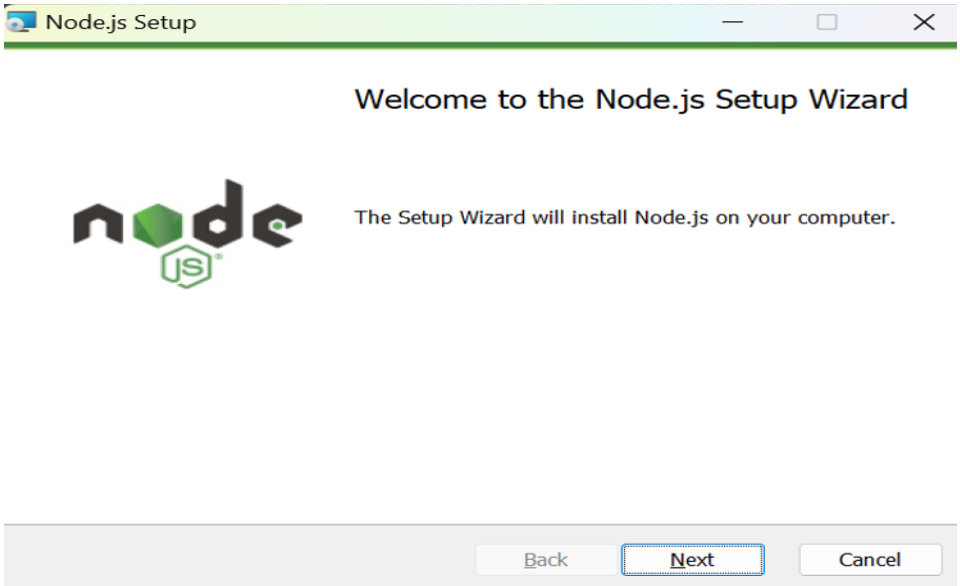




b) Building of a RESTful API and testing it with the Postman tool.

- Download Nodejs Installer: https://nodejs.org/en/download/prebuilt-installer and complete the setup.





- Check node installation by using following commands:

```
C:\Cloud>node -v
v20.15.1

C:\Cloud>npm -v
10.7.0
```

- Create a folder for your Project – REST and make it your current working directory.

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS C:\Cloud\nodejs\REST>
```

- Initialize your project using the command : **npm init -y**

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS C:\Cloud\nodejs\REST> npm init -y
Wrote to C:\Cloud\nodejs\REST\package.json:

{
  "name": "rest",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "license": "ISC",
  "dependencies": {
    "express": "^4.19.2",
    "nodemon": "^3.1.4"
  },
  "devDependencies": {},
  "description": ""
}
```

- This creates a package.json file in the project folder.

```
{} package.json ×
C: > Program Files > nodejs > REST > {} package.json > ...
1  {
2    "name": "rest",
3    "version": "1.0.0",
4    "main": "index.js",
       ▷ Debug
5    "scripts": {
6      "test": "echo \"Error: no test specified\" && exit 1"
7    },
8    "keywords": [],
9    "author": "",
10   "license": "ISC",
11   "description": ""
12 }
13
```

- Install Express using the command : **npm install express**

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS C:\Cloud\nodejs\REST> npm i express

up to date, audited 94 packages in 2s

16 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

- Install nodemon using the command : **npm install nodemon**

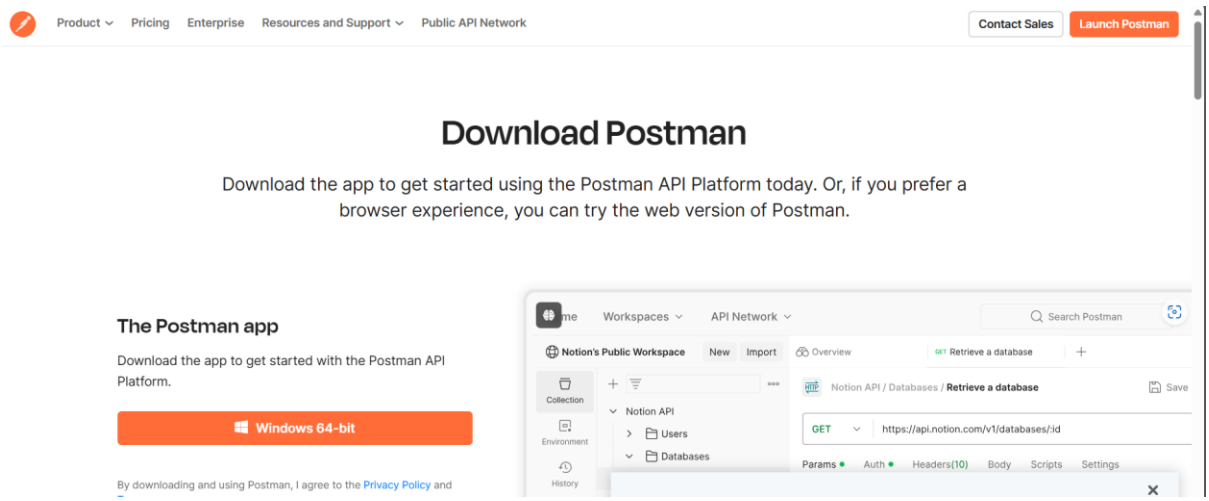
```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS C:\Cloud\nodejs\REST> npm install nodemon

up to date, audited 94 packages in 4s

16 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Cloud\nodejs\REST>
```

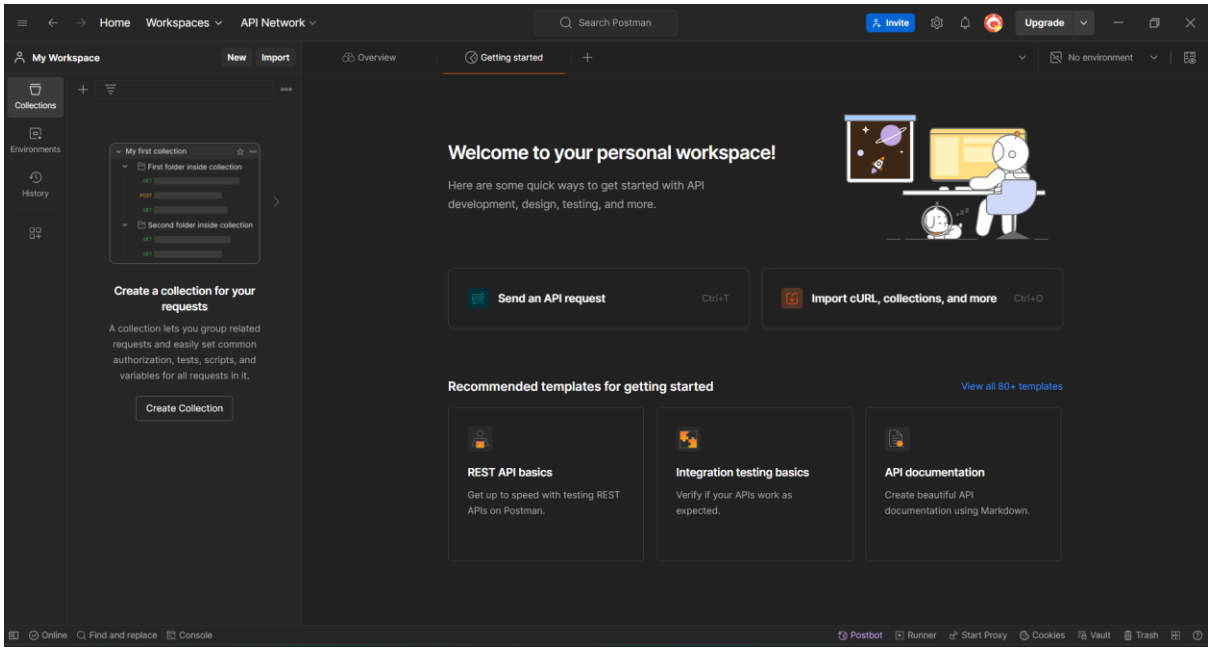
- Run a simple server.js code for testing.

```
JS server.js > ...
const http = require('http');
const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Have a good day');
});
const port = 3000;
server.listen(port, () => {
  console.log(`Server running at http://localhost:${port}/`);
});
```

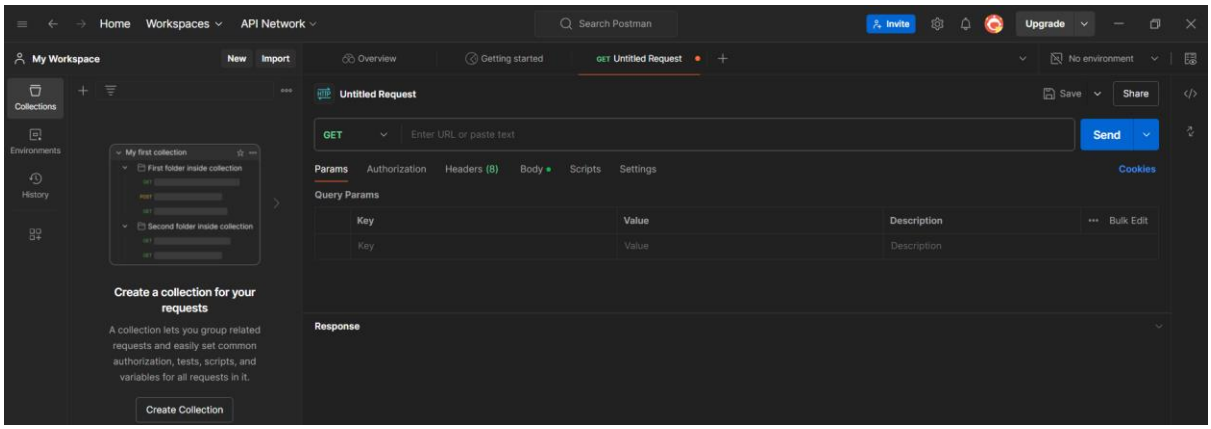
```
PS C:\Cloud\nodejs\REST> node server
Server running at http://localhost:3000/
```

- Install Postman tool: https://www.postman.com/downloads/ and open it.





- Click **Send an API request**.



- Do code for the Football players api and save it on your project directory.

CODE:

football.js

```
const express = require('express');
const bodyParser = require('body-parser');
const app = express();
const port = 5000;

let players = [
  { id: 1, name: 'Lionel Messi', nationality: 'Argentina', position: 'Attacker', goals: 700 },
  { id: 2, name: 'Cristiano Ronaldo', nationality: 'Portugal', position: 'Attacker', goals: 800 },
  { id: 3, name: 'Neymar jr', nationality: 'Brazil', position: 'Attacker', goals: 400 },
  { id: 4, name: 'David Beckham', nationality: 'England', position: 'Midfielder', goals: 200 },
  { id: 5, name: 'Sergio ramos', nationality: 'Spain', position: 'Defender', goals: 800 }
];
app.use(bodyParser.json());
```

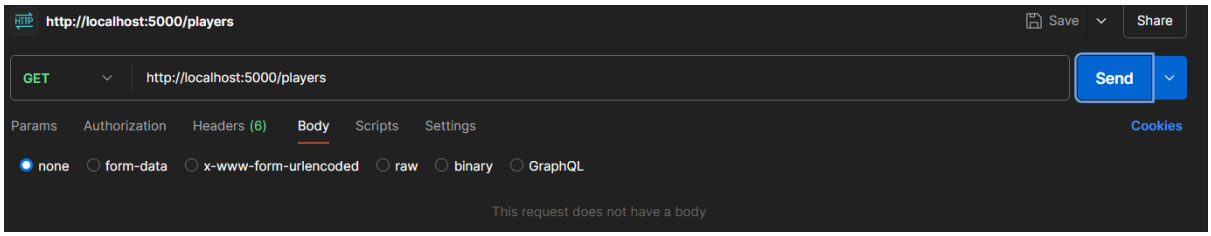
```
app.get('/players', (req, res) => {
  res.json(players);
});

app.get('/players/:id', (req, res) => {
  const playerId = parseInt(req.params.id);
  const player = players.find(player => player.id === playerId);
  if (player) {
    res.json(player);
  } else {
    res.status(404).send('Player not found');
  }
});

app.post('/players', (req, res) => {
  const newPlayer = req.body;
  if (!newPlayer || !newPlayer.name || !newPlayer.nationality || !newPlayer.position ||
!newPlayer.goals) {
    return res.status(400).send('Missing required fields');
  }

  newPlayer.id = Math.max(...players.map(player => player.id)) + 1;
  players.push(newPlayer);
  res.status(201).json(newPlayer);
});

app.put('/players/:id', (req, res) => {
  const playerId = parseInt(req.params.id);
  const updatedPlayer = req.body;
  const playerIndex = players.findIndex(player => player.id === playerId);

  if (playerIndex !== -1) {
    players[playerIndex] = { ...players[playerIndex], ...updatedPlayer };
    res.json(players[playerIndex]);
  } else {
    res.status(404).send('Player not found');
  }
});

app.delete('/players/:id', (req, res) => {
  const playerId = parseInt(req.params.id);
  const playerIndex = players.findIndex(player => player.id === playerId);

  if (playerIndex !== -1) {
    players.splice(playerIndex, 1);
    res.sendStatus(204);
  } else {
    res.status(404).send('Player not found');
```

```
  }
});

app.listen(port, () => {
  console.log(`Server listening on port ${port}`);
});
```

- Run the code football.js (server)

```
PS C:\Cloud\nodejs\REST> node football.js
Server listening on port 5000
```

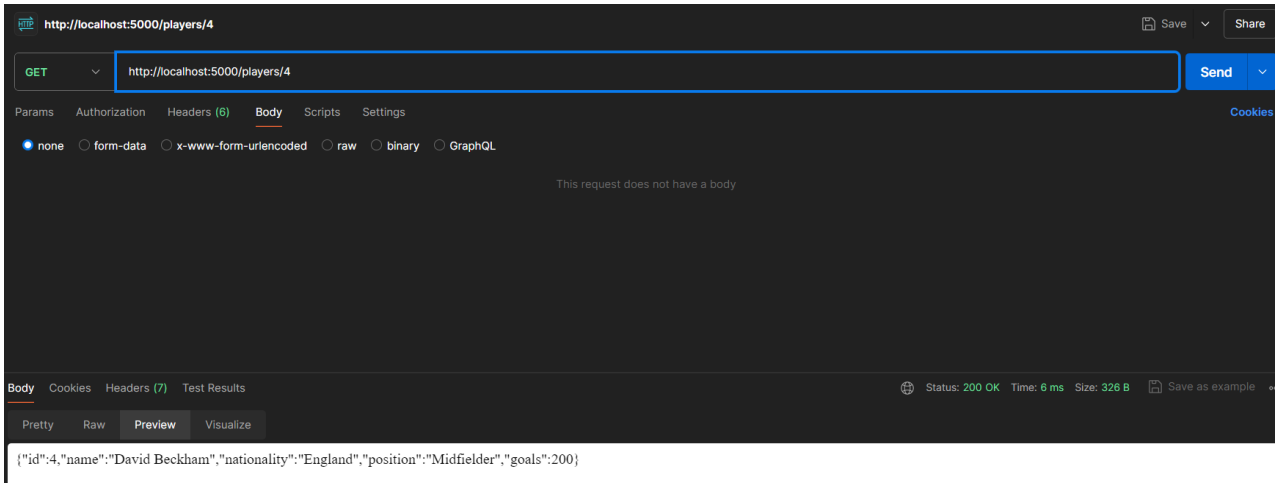
- Enter the URL of your server **http://localhost:5000** ,the port where your API code is running. And in the address bar and append **/players** to it.



GET:

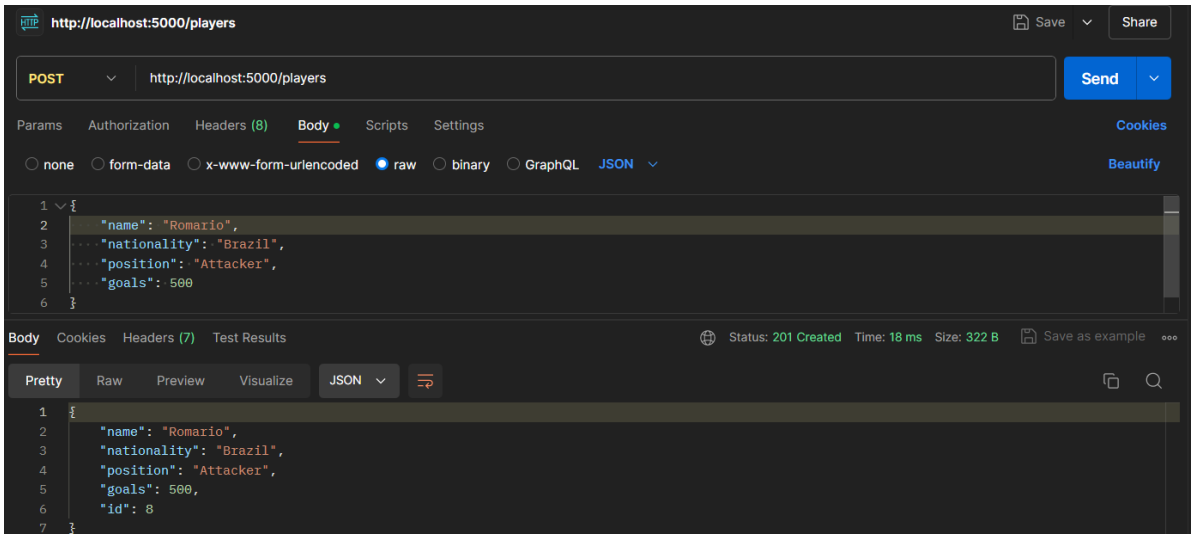
- Select request method as GET and click send. This gives details about all players.

- Append **/id** to the url to get details about the player with particular id.



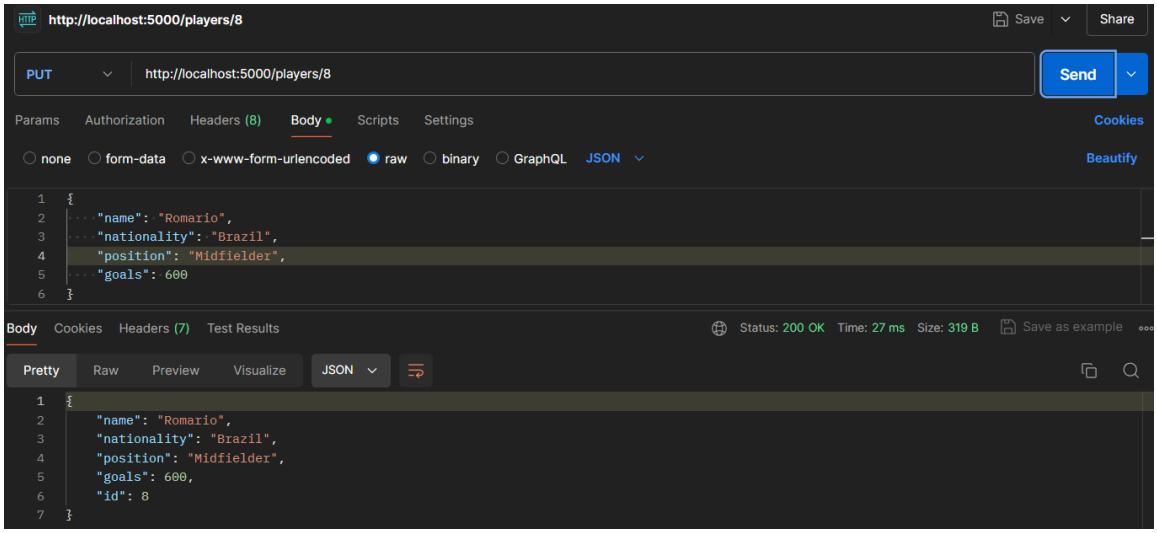
POST:

- Select request method as POST ,enter the details of new player in the body and click send. It adds the new user details to the server.



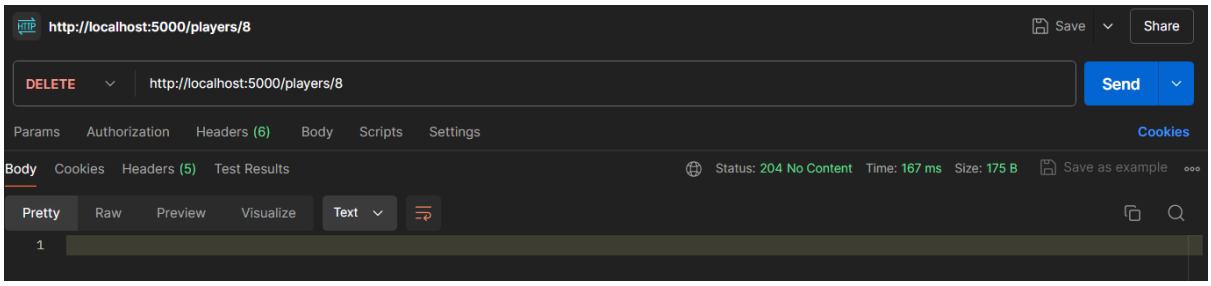
PUT:

- Select request method as PUT ,specify the id to update on the URL, enter the details to update in the body and click send. It updates the details.



DELETE:

- Select request method as DELETE ,specify the id to delete on the URL and click send. It deletes the respective id details.



c) To consume a third party API in a webpage.

CODE:
quotes.html

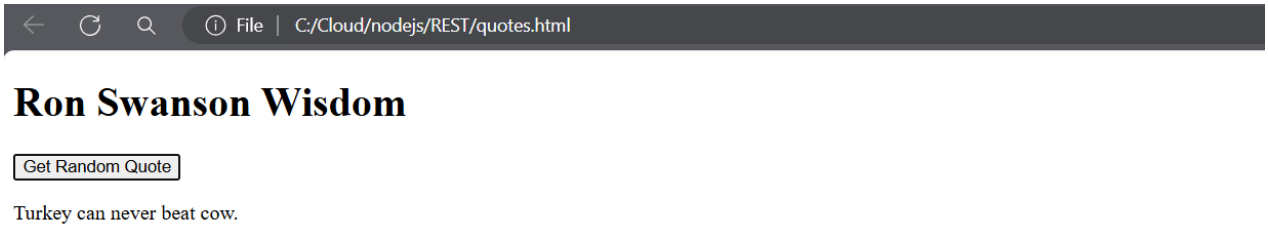
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Random Ron Swanson Quote</title>
</head>
<body>
  <h1>Ron Swanson Wisdom</h1>
  <button id="get-quote-button">Get Random Quote</button>
  <p id="quote-display"></p>
```

```
 <script src="quotes.js">
 </script>
</body>
</html>
```

quotes.js

```
const quoteButton = document.getElementById('get-quote-button');
const quoteDisplay = document.getElementById('quote-display');

quoteButton.addEventListener('click', getQuote);

function getQuote() {
 fetch('https://ron-swanson-quotes.herokuapp.com/v2/quotes')
  .then(response => response.json())
  .then(data => {
    const quote = data[0];
    quoteDisplay.textContent = quote;
  })

  .catch(error => {
    console.error('Error fetching quote:', error);
    quoteDisplay.textContent = 'Error: Could not retrieve quote.';
  });
}
```

OUTPUT:



RESULT:

Thus, the guided projects in IBM skills network lab: Building RESTful APIs with Express and Working with simple APIs, Building of a RESTful API and testing it with the Postman tool and consuming a third party API in a webpage have been completed successfully and output have been verified.