
marp: true

REST API Project Presentation

By Aaron Hesse

ASE 230 – Project 1

This presentation covers the 10 REST APIs implemented, how they work, and examples of requests and responses.

Overview

I implemented 10 REST APIs using PHP/MySQL and JSON.

Some APIs use Bearer Token authentication for secure data access.

The APIs included are: Countries, Weather, Pets, Trivia, Users, Dropbox, GitHub, Indeed, NASA, and Spotify.

Countries API

Purpose: Provides country information such as capital, population, and region. Useful for applications involving travel, mapping, or geography.

Endpoints: GET, POST, PUT, DELETE /countries.php

Data: Stored in MySQL (CRUD operations demonstrated).

Countries API – Example

Request (GET):

```
curl -X GET http://localhost:8000/countries.php
```

Response (JSON):

```
[  
  {"id":1,"name":"United States","capital":"Washington D.C.,"population":331000000},  
  {"id":2,"name":"Canada","capital":"Ottawa","population":38000000}  
]
```

Weather API

Purpose: Retrieves current weather conditions for cities. Demonstrates dynamic external data retrieval.

Endpoints: GET, POST, PUT, DELETE /weather.php

Data: Demonstrates integration with an external data source.

Weather API – Example

Request (GET):

```
curl -X GET http://localhost:8000/weather.php
```

Response (JSON):

```
[  
  {"id":1,"city":"New York","temperature":22,"condition":"Cloudy"},  
  {"id":2,"city":"Miami","temperature":30,"condition":"Sunny"}  
]
```

Pets API

Purpose: Manages pet information including name, type, and owner. Demonstrates CRUD operations with MySQL storage.

Endpoints: GET, POST, PUT, DELETE /pets.php

Data: Stored in MySQL.

Pets API – Example

Request (GET):

```
curl -X GET http://localhost:8000/pets.php
```

Response (JSON):

```
[  
  {"id":1,"name":"Bella","kind":"Dog","owner":"Alice Johnson"},  
  {"id":2,"name":"Luna","kind":"Cat","owner":"Bob Smith"}  
]
```

Trivia API

Purpose: Provides trivia questions and answers. Useful for quiz or educational apps.

Endpoints: GET, POST, PUT, DELETE /trivia.php

Data: Stored in JSON (simple data retrieval demonstrated).

Trivia API – Example

Request (GET):

```
curl -X GET http://localhost:8000/trivia.php
```

Response (JSON):

```
[  
  {"id":1,"question":"What is the capital of France?","answer":"Paris"},  
  {"id":2,"question":"Which planet is known as the Red Planet?","answer":"Mars"}  
]
```

Users API

Purpose: Manages user information such as name, email, and major. Demonstrates CRUD operations for user management.

Endpoints: GET, POST, PUT, DELETE /users.php

Data: Stored in MySQL.

Users API – Example

Request (GET):

```
curl -X GET http://localhost:8000/users.php
```

Response (JSON):

```
[  
  {"id":1,"name":"Alice Johnson","email":"alice.johnson@university.edu"},  
  {"id":2,"name":"Bob Smith","email":"bob.smith@university.edu"}  
]
```

Dropbox API (Bearer Token)

Purpose: Manages files and folders in Dropbox securely. Demonstrates secure API access using Bearer Tokens.

Endpoints: GET, POST, PUT, DELETE /dropbox.php

Security: Requires Authorization: Bearer <TOKEN> header.

Dropbox API – Example

Request (GET):

```
curl -X GET -H "Authorization: Bearer DROPBOXTOKEN98765"  
http://localhost:8000/dropbox.php
```

Response (JSON):

```
[  
  {"id":1,"filename":"notes.txt","size":"12KB"},  
  {"id":2,"filename":"project.zip","size":"120MB"}  
]
```

GitHub API

Purpose: Retrieves repository information securely. Demonstrates secure access to external services using Bearer Tokens.

Endpoints: GET, POST, PUT, DELETE /github.php

Security: Requires Authorization: Bearer <TOKEN> header.

GitHub API – Example

Request (GET):

```
curl -X GET -H "Authorization: Bearer GITHUBTOKEN12345"  
http://localhost:8000/github.php
```

Response (JSON):

```
[  
  {"id":1,"repo":"api_project","language":"PHP"},  
  {"id":2,"repo":"portfolio-site","language":"JavaScript"}  
]
```

Indeed API

Purpose: Provides job listings and company information. Useful for displaying real-time job postings.

Endpoints: GET, POST, PUT, DELETE /indeed.php

Data: Demonstrates integration with a job listing service.

Indeed API – Example

Request (GET):

```
curl -X GET http://localhost:8000/indeed.php
```

Response (JSON):

```
[  
  {"id":1,"position":"Software Engineer","company":"TechCorp","location":"New York"},  
  {"id":2,"position":"Network Security Analyst","company":"SecureNet","location":"Chicago"}  
]
```

NASA API (Bearer Token)

Purpose: Provides astronomy data (e.g., Astronomy Picture of the Day). Demonstrates integration with scientific APIs using Bearer Tokens.

Endpoints: GET, POST, PUT, DELETE /nasa.php

Security: Requires Authorization: Bearer <TOKEN> header.

NASA API – Example

Request (GET):

```
curl -X GET -H "Authorization: Bearer NASA123TOKEN" http://localhost:8000/nasa.php
```

Response (JSON):

```
[  
  {"id":1,"title":"Mars Rover Mission","status":"Active"},  
  {"id":2,"title":"James Webb Telescope","status":"Operational"}  
]
```

Spotify API (Bearer Token)

Purpose: Provides music and playlist data securely via Spotify API. Shows secure access to entertainment APIs using Bearer Tokens.

Endpoints: GET, POST, PUT, DELETE /spotify.php

Security: Requires Authorization: Bearer <TOKEN> header.

Spotify API – Example

Request (GET):

```
curl -X GET -H "Authorization: Bearer SPOTIFYTOKEN12345"  
http://localhost:8000/spotify.php
```

Response (JSON):

```
[  
  {"id":1,"track":"Blinding Lights","artist":"The Weeknd"},  
  ...  
]
```

```
{"id":2,"track":"Levitating","artist":"Dua Lipa"}  
]
```

Project Summary

This project included 10 REST APIs, 2 of which required Bearer Token authentication.
All CRUD operations were implemented and tested with .sh scripts and test.html.
Data was stored in both JSON and MySQL, demonstrating full database integration.