

What is Rest API ?

Application Programming Interface(API) is a software interface that allows two applications or softwares to interact ,communicate and exchange data with each other without any user intervention.

API is a collection of software Functions and procedures.

REST(Representational State Transfer)

- REST protocols overcome SOAP's dependency on XML by supporting data transmission in multiple formats such as JSON (most prominent), HTML, Python, plain text as well as media files
- REST APIs follow a client-server architecture and must be stateless.
- Stateless communication implies that no client data is stored between GET requests.

HTTP:-

HTTP stands for Hypertext Transfer Protocol.

Hypertext Transfer Protocol is a set of rule which Is used for transferring the files like,audio,video,image,text and other multimedia files on the www.

HTTP methods

Following four HTTP methods are commonly used in REST based architecture.

- GET – Provides a read only access to a resource.
- POST – Used to create a new resource.
- DELETE – Used to remove a resource.
- PUT – Used to update a existing resource or create a new resource.

JPA – Java Persistent API

- The Java Persistence API (JPA) is a specification of Java. It is used to persist data between Java object and relational database.
- JPA acts as a bridge between object-oriented domain models and relational database systems.

RestDemo

```
package com.example.demo;  
  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.RequestParam;  
import org.springframework.web.bind.annotation.RestController;
```

```
@RestController
```

```
public class RestapiApplication {
```

```
    private int wins;
```

```
    private int losses;
```

```
    // Use @RequestParam to bind query parameters
```

```
    @RequestMapping(value = "/score/updates", method =  
RequestMethod.PUT)
```

```
    public String updateScore(@RequestParam int wins, @RequestParam int  
losses) {
```

```
        this.wins = wins;
```

```
        this.losses = losses;
```

```
        return "wins=" + wins + ", losses=" + losses;
```

```
    }
```

```
    @RequestMapping(value = "/score/delete", method =  
RequestMethod.DELETE)
```

```
    public String deleteScore() {
```

```
        this.wins = 10;
```

```
        this.losses = 10;
```

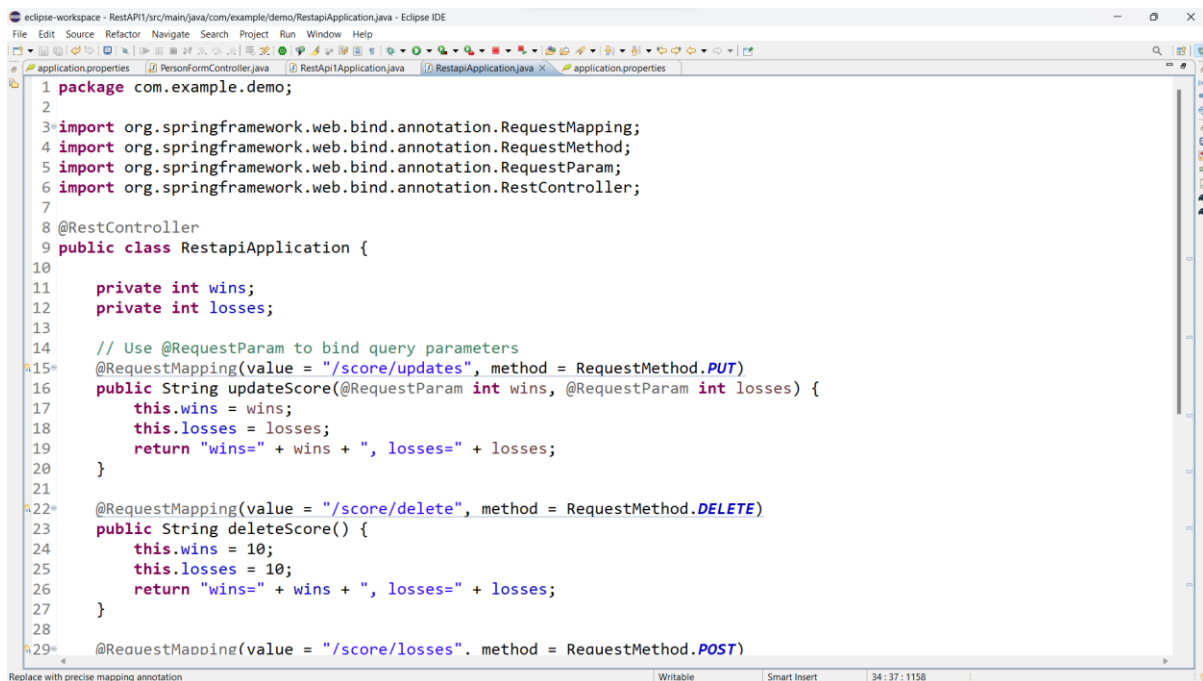
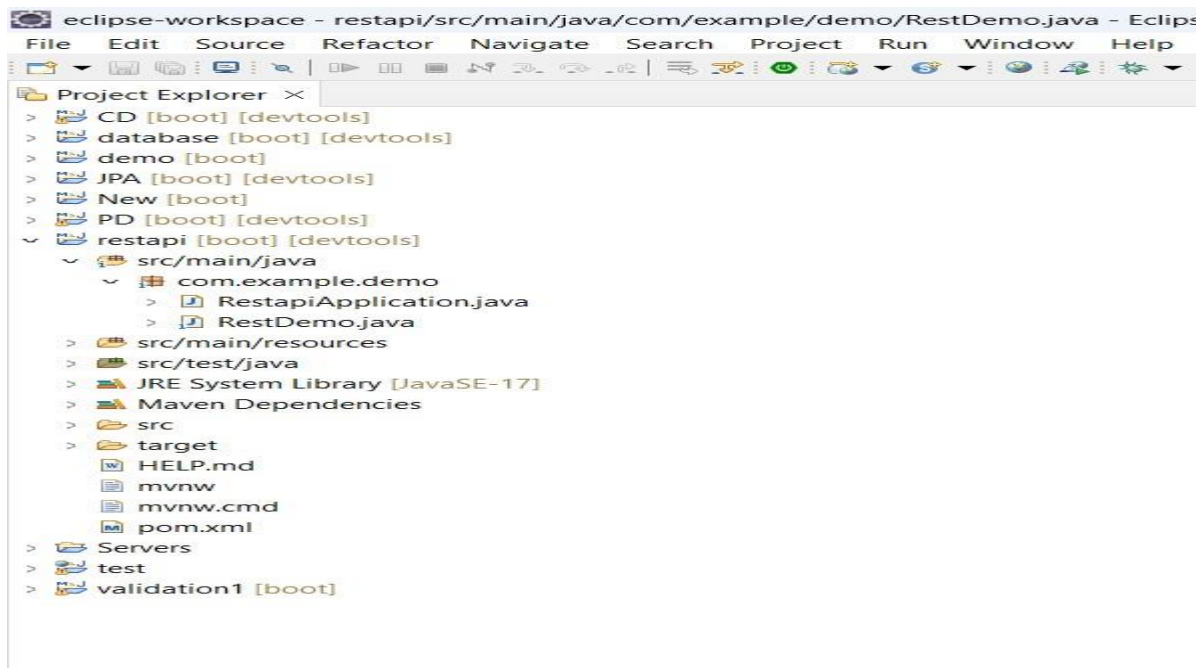
```
        return "wins=" + wins + ", losses=" + losses;
    }
```

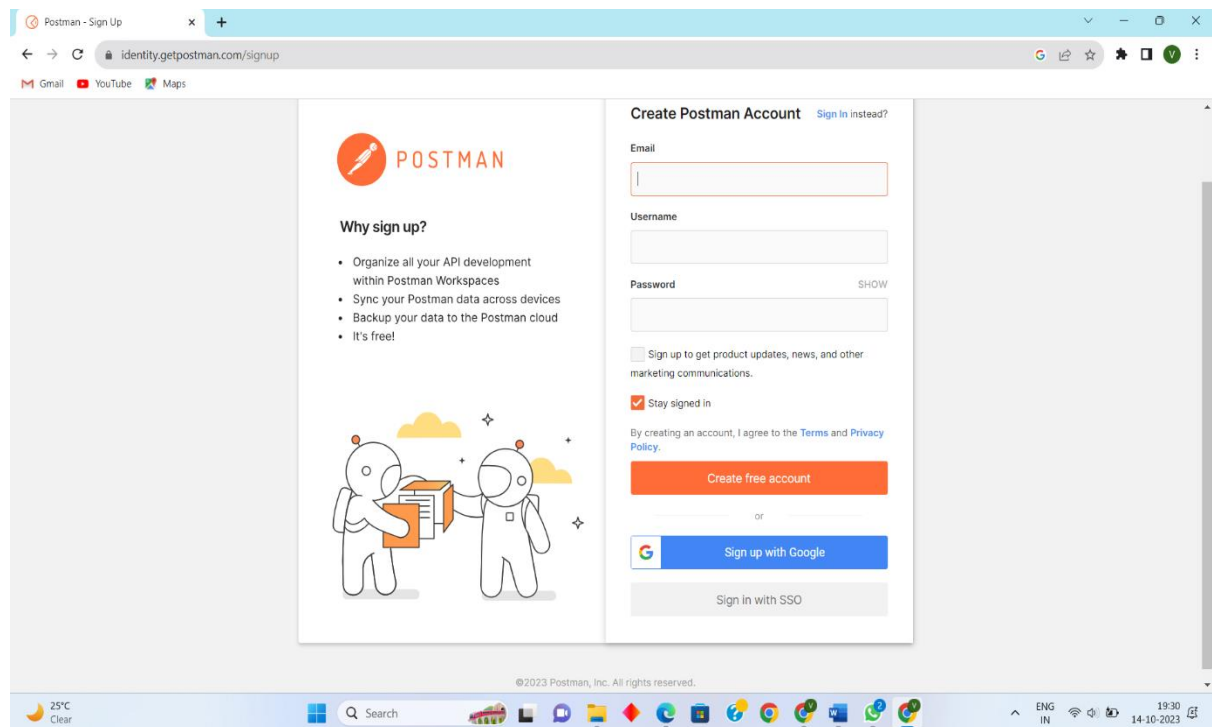
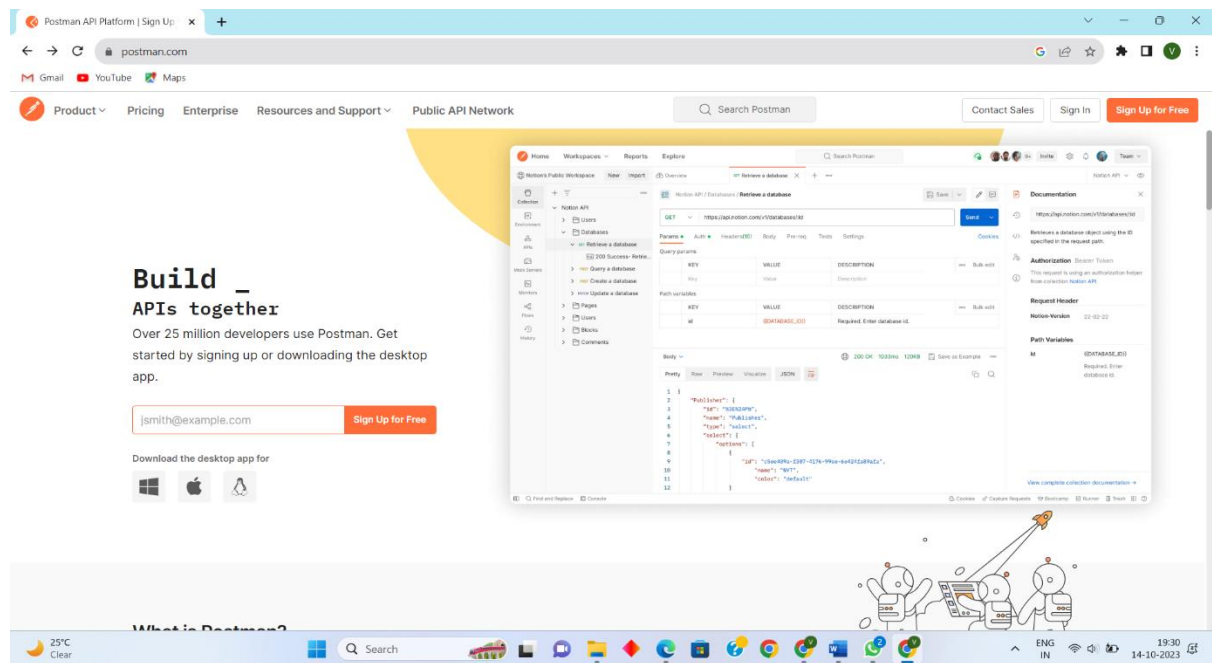
```
@RequestMapping(value = "/score/losses", method = RequestMethod.POST)
public int increaseLosses() {
    return ++losses; // Increment before returning
}
```

```
@RequestMapping(value = "/score/wins", method = RequestMethod.POST)
public int increaseWins() {
    return ++wins; // Increment before returning
}
```

```
@RequestMapping(value = "/score/wins", method = RequestMethod.GET)
public int getWins() {
    return wins;
}
```

```
@RequestMapping(value = "/score/losses", method = RequestMethod.GET)
public int getLosses() {
    return losses;
}
}
```





The top screenshot shows the Postman 'Overview' page for a workspace named 'My Workspace'. The URL bar displays 'web.postman.co/workspace/My-Workspace~82ad3154-b22b-4dd8-8727-c6e565ba8e4/overview'. The left sidebar contains 'Collections', 'Environments', and 'History'. The main area features a 'Use a template to quickly set up your workspace' section with options for 'API demos', 'API development', and 'API testing'. Below this is a 'Create a collection for your requests' section with a 'Create Collection' button. The bottom status bar shows '25°C Clear' and various system icons.

The bottom screenshot shows the Postman 'Request' page for a GET request to 'http://localhost:9090/score/wins'. The URL bar displays 'http://localhost:9090/score/wins' and the address bar shows 'red-space-987226.postman.co/workspace/My-Workspace~a76bf119-6c13-4d43-8563-ae874bf5b0e/request/create?requestId=85f8cd1c-a5ac-4113-ba35-5d5f83...'. The left sidebar is the same as the top screenshot. The main area shows the request details, including a 'Send' button, 'Params' tab, 'Query Params' table, 'Body' tab, and 'Test Results' section. The 'Query Params' table has the following structure:

Key	Value	Description
Key	Value	Description

The 'Body' tab shows a status of '200 OK', 'Time: 111 ms', and 'Size: 165 B'. The 'Test Results' section shows a 'Pretty' view of the response body.

http://localhost:9090/score/wins

GET http://localhost:9090/score/wins

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 111 ms Size: 165 B Save as Example

1 0

http://localhost:9090/score/update?wins=8&losses=5

PUT http://localhost:9090/score/update?wins=8&losses=5

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

wins 8

losses 5

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 15 ms Size: 178 B Save as Example

1 wins=8losses=5

The screenshot displays the Postman API client interface. The browser address bar shows the URL `http://localhost:9090/score/delete`. The Postman interface includes a sidebar with 'My Workspace' and 'Collections'. The main panel shows a DELETE request to `http://localhost:9090/score/delete`. The 'Query Params' section is empty. The 'Body' tab is selected, showing the JSON payload `{ "wins": 0, "losses": 0 }`. The status bar at the bottom indicates a successful response with status 200 OK, time 6 ms, and size 178 B.

Home Workspaces API Network Explore Search Postman

My Workspace New Import Overview DEL http://localhost:9090/s

http://localhost:9090/score/delete Save

DELETE http://localhost:9090/score/delete Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (5) Test Results Status: 200 OK Time: 6 ms Size: 178 B Save as Example

Pretty Raw Preview Visualize Text

```
1 { "wins": 0, "losses": 0 }
```

Online Console Postbot Runner Auto-select agent Cookies Trash

Search

15:49 10-10-2023