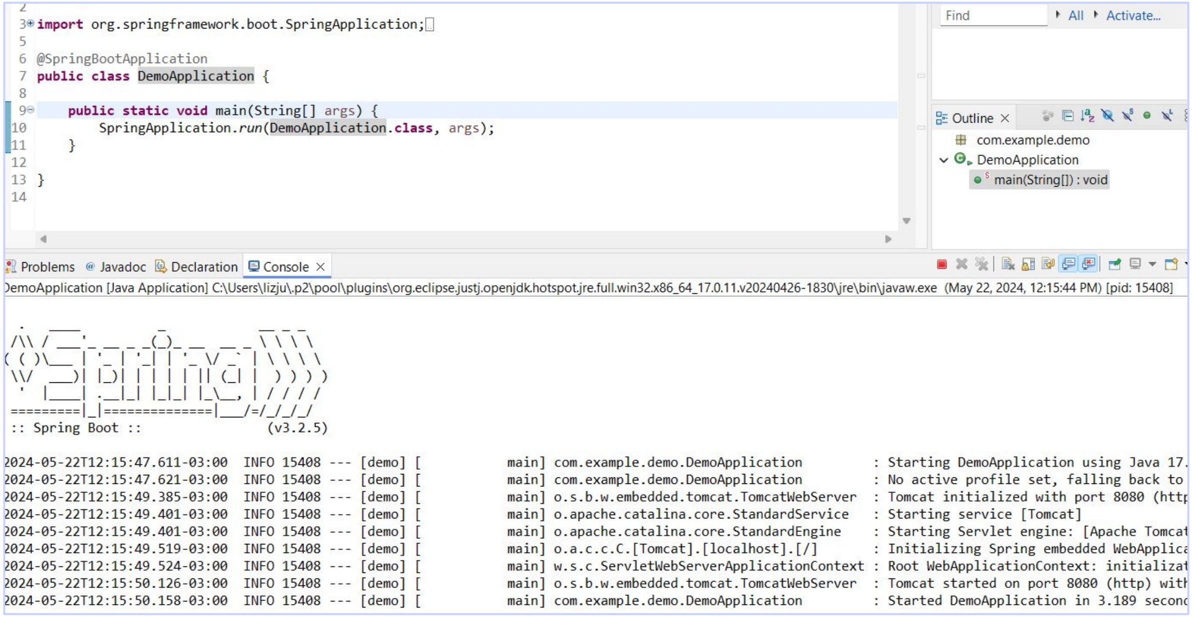
Probar las *APIs*

Ejecutar Spring Boot

Levantar Spring Boot,
en el IDE.



```java
import org.springframework.boot.SpringApplication;

@SpringBootApplication
public class DemoApplication {

    public static void main(String[] args) {
        SpringApplication.run(DemoApplication.class, args);
    }

}
```

```
DemoApplication [Java Application] C:\Users\lizju\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.11.v20240426-1830\jre\bin\javaw.exe  (May 22, 2024, 12:15:44 PM) [pid: 15408]

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v3.2.5)

2024-05-22T12:15:47.611-03:00  INFO 15408 --- [demo] [           main] com.example.demo.DemoApplication         : Starting DemoApplication using Java 17.
2024-05-22T12:15:47.621-03:00  INFO 15408 --- [demo] [           main] com.example.demo.DemoApplication         : No active profile set, falling back to
2024-05-22T12:15:49.385-03:00  INFO 15408 --- [demo] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat initialized with port 8080 (http
2024-05-22T12:15:49.401-03:00  INFO 15408 --- [demo] [           main] o.apache.catalina.core.StandardService   : Starting service [Tomcat]
2024-05-22T12:15:49.401-03:00  INFO 15408 --- [demo] [           main] o.apache.catalina.core.StandardEngine    : Starting Servlet engine: [Apache Tomcat
2024-05-22T12:15:49.519-03:00  INFO 15408 --- [demo] [           main] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring embedded WebApplica
2024-05-22T12:15:49.524-03:00  INFO 15408 --- [demo] [           main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initializat
2024-05-22T12:15:50.126-03:00  INFO 15408 --- [demo] [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 8080 (http) with
2024-05-22T12:15:50.158-03:00  INFO 15408 --- [demo] [           main] com.example.demo.DemoApplication         : Started DemoApplication in 3.189 second
```

Postman

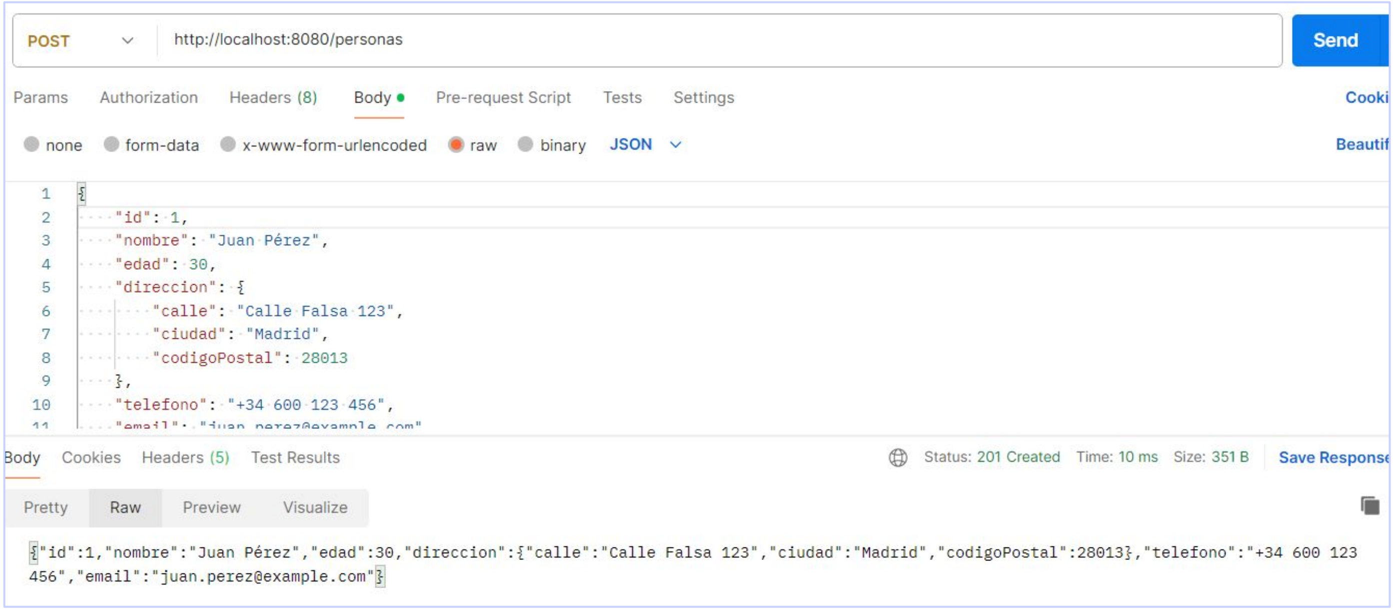
Es una herramienta que, **entre otras cosas, permite probar APIs.**

Fue creada para facilitar el trabajo de desarrolladores. Permite **enviar solicitudes HTTP y ver las respuestas de forma rápida y sencilla.** Algunas de sus funciones principales incluyen: Manejo de colecciones, entornos y variables, pruebas automatizadas, documentación de API, monitoreo de APIs.

Se puede descargar de Postman.com.

POST

En Postman, seleccionar ***JSON*** y agregar a una persona.

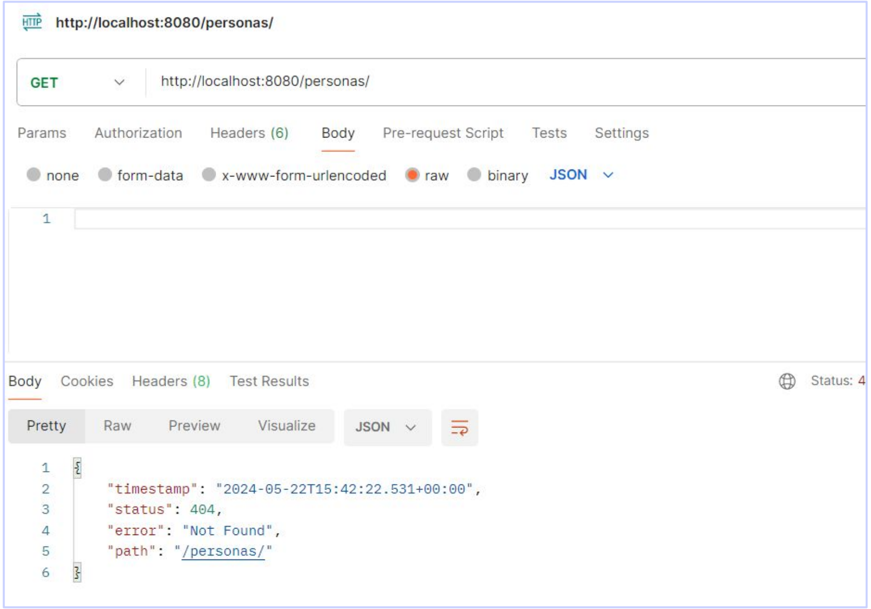


```
{
    "id": 1,
    "nombre": "Juan Pérez",
    "edad": 30,
    "direccion": {
        "calle": "Calle Falsa 123",
        "ciudad": "Madrid",
        "codigoPostal": 28013
    },
    "telefono": "+34 600 123 456",
    "email": "juan.perez@example.com"
```

```
{"id":1,"nombre":"Juan Pérez","edad":30,"direccion":{"calle":"Calle Falsa 123","ciudad":"Madrid","codigoPostal":28013},"telefono":"+34 600 123
456","email":"juan.perez@example.com"}
```

GET

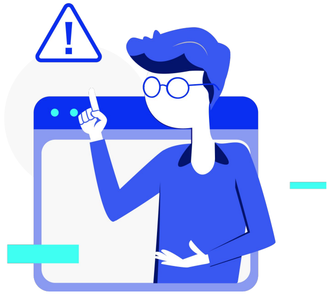
Cuidado con este **error:**


http://localhost:8080/personas/

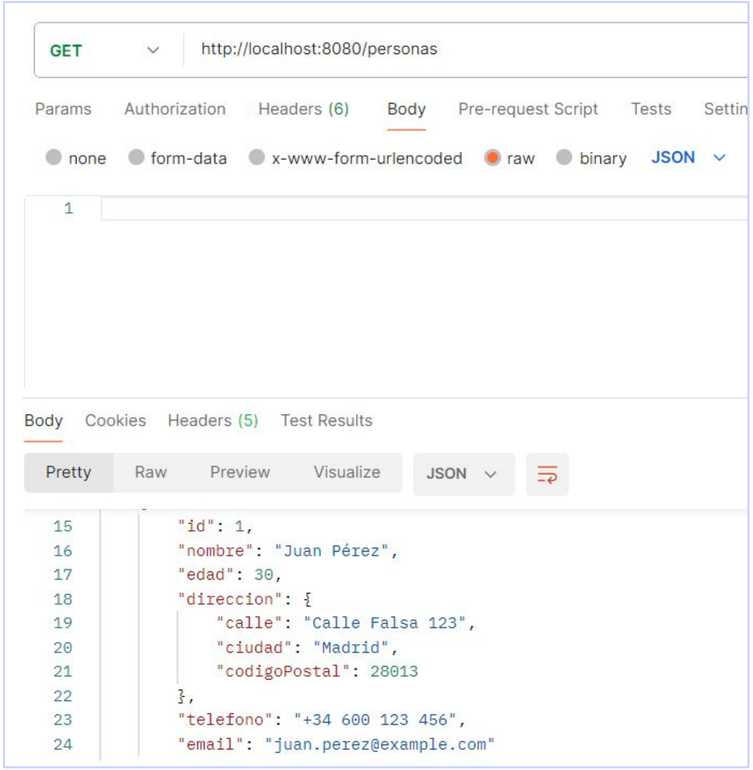
GET http://localhost:8080/personas/

```
{
    "timestamp": "2024-05-22T15:42:22.531+00:00",
    "status": 404,
    "error": "Not Found",
    "path": "/personas/"
}
```

Ocurre porque la URL correcta es `http://localhost:8080/personas` **sin barra "/" al final.**



Si se coloca **`/personas`** con GET, traerá una lista de personas.

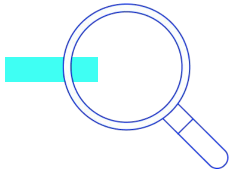


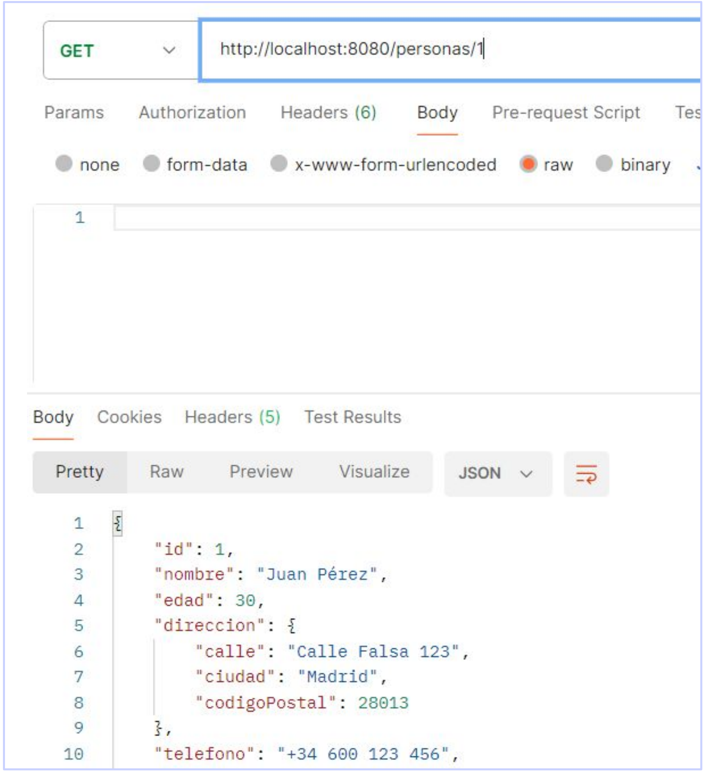
http://localhost:8080/personas

```
        "id": 1,
        "nombre": "Juan Pérez",
        "edad": 30,
        "direccion": {
            "calle": "Calle Falsa 123",
            "ciudad": "Madrid",
            "codigoPostal": 28013
        },
        "telefono": "+34 600 123 456",
        "email": "juan.perez@example.com"
```

GET by id

Si se coloca **`/personas/{id}`** con GET, traerá una persona por su `id`.

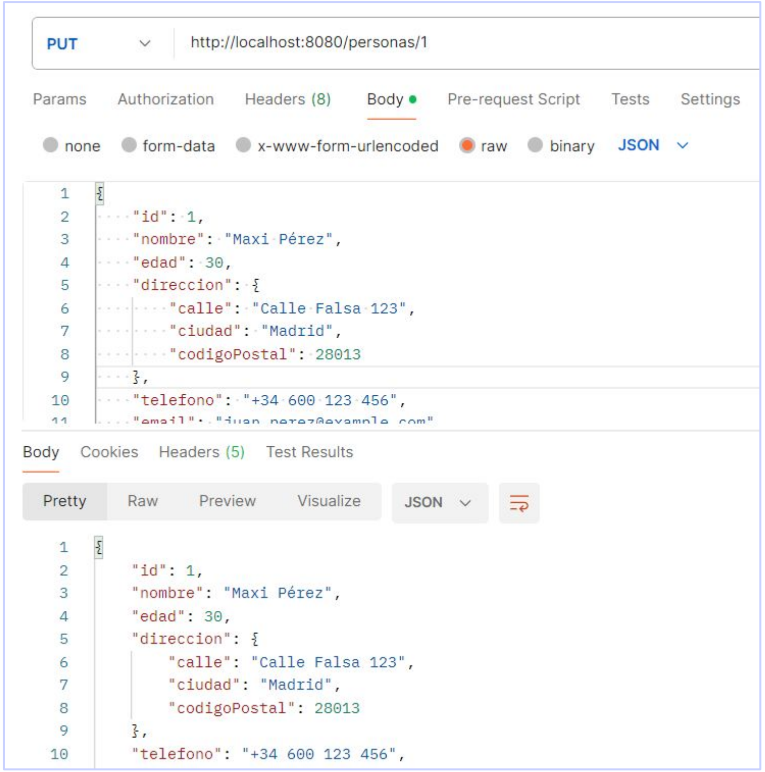




```
{
    "id": 1,
    "nombre": "Juan Pérez",
    "edad": 30,
    "direccion": {
        "calle": "Calle Falsa 123",
        "ciudad": "Madrid",
        "codigoPostal": 28013
    },
    "telefono": "+34 600 123 456",
```

PUT

Si se coloca **`/personas/{id}`** con PUT, actualizará una persona por su **`id`**.



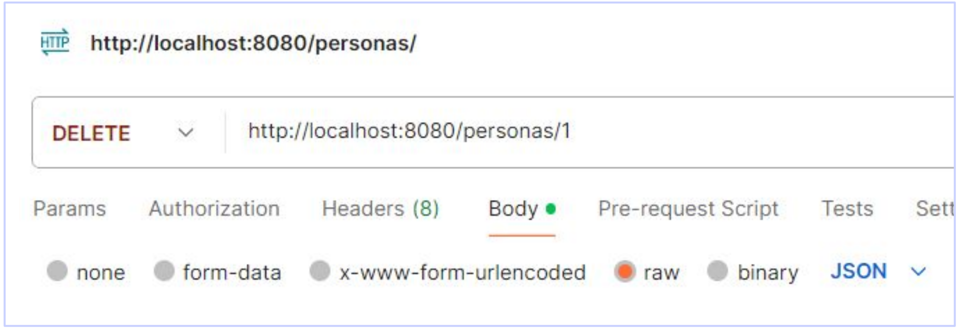
PUT http://localhost:8080/personas/1

```json
{
    "id": 1,
    "nombre": "Maxi Pérez",
    "edad": 30,
    "direccion": {
        "calle": "Calle Falsa 123",
        "ciudad": "Madrid",
        "codigoPostal": 28013
    },
    "telefono": "+34 600 123 456",
    "email": "juan.perez@example.com"
```

```json
{
    "id": 1,
    "nombre": "Maxi Pérez",
    "edad": 30,
    "direccion": {
        "calle": "Calle Falsa 123",
        "ciudad": "Madrid",
        "codigoPostal": 28013
    },
    "telefono": "+34 600 123 456",
```

DELETE

Si se coloca **`/personas/{id}`** con DELETE, borra una persona por su `id`.



http://localhost:8080/personas/

DELETE http://localhost:8080/personas/1

none form-data x-www-form-urlencoded [✓ raw] binary JSON