

## Testing II

# Testing en APIs

Teniendo en cuenta el contexto, es crucial probar la funcionalidad, la fiabilidad, el rendimiento y la seguridad de las API. Para ello, hay una serie de herramientas que apoyan estas pruebas, ya sean manuales o automatizadas. Postman, por ejemplo, es una de las herramientas que vimos en Testing I.

Se pueden realizar algunas pruebas para:

- Validar los datos de retorno.
- Validar los headers de respuesta.
- Validar que la respuesta es conforme.
- Validar el comportamiento después de cambiar el tipo de contenido.
- Validar si la estructura JSON o XML es correcta.
- Validar si el estado es acorde a los códigos de error.
- Validar el comportamiento de la solicitud con información incompleta.

## Rest Assured

Rest Assured es una de las herramientas de código abierto basadas en Java que se utilizan para simplificar las pruebas de los servicios REST. Permite crear llamadas HTTP y utilizar métodos y utilidades para validar la respuesta recibida del servidor, como el código de estado y otra información. Además, se puede utilizar junto con JUnit.

En el siguiente código tenemos la creación del script usando Rest Assured.

```
import io.restassured.RestAssured;
import io.restassured.http.Method;
import io.restassured.response.Response;
import io.restassured.specification.RequestSpecification;

public class RestAssuredAPITest {

    @Test
    public void GetBooksDetails() {
        // Specify the base URL to the RESTful web service
        RestAssured.baseURI = "https://demoqa.com/BookStore/v1/Books";
        // Get the RequestSpecification of the request to be sent to the server.
        RequestSpecification httpRequest = RestAssured.given();
        // specify the method type (GET) and the parameters if any.
        //In this case the request does not take any parameters
        Response response = httpRequest.request(Method.GET, "");
        // Print the status and message body of the response received from the server
        System.out.println("Status received => " + response.getStatusLine());
        System.out.println("Response=>" + response.prettyPrint());
    }
}
```

## Métodos HTTP

Los métodos identifican los diferentes tipos de operaciones que se pueden realizar para manipular los datos. En general, los métodos principales representan operaciones CRUD (crear, leer, actualizar, borrar).

Los códigos de estado HTTP se utilizan como respuesta a la solicitud, para indicar el estado/resultado de la misma. Estos códigos se agrupan en diferentes grupos: Informal, Éxito, Redirección, Error del Cliente y Error del Servidor.

Compruebe a continuación los principales métodos, sus significados y los códigos de respuesta:

Verbo HTTP	Descripción	Respuesta
<i>POST</i>	Se utiliza para crear un nuevo registro en la base de datos	Status code 201 - registro creado
<i>GET</i>	Se utiliza para leer registros de la base de datos	Status code 200 - devuelve los registros en el formato solicitado
<i>PUT</i>	Se utiliza para actualizar un registro en la base de datos	Status code 200 - registro actualizado
<i>PATCH</i>	Se utiliza para actualizar parte de un registro en la base de datos	Status code 200 - registro actualizado
<i>DELETE</i>	Se utiliza para eliminar un registro en la base de datos	Status code 204 - registro eliminado

¡Comprueben [aquí](#) los demás códigos de forma creativa!