

Práctica 1. Testing con Spring

Convocatoria extraordinaria

Se desea implementar las pruebas necesarias para comprobar la correcta funcionalidad de una aplicación que gestiona una librería online. Se proporciona el código de dicha aplicación (publicado en el Aula Virtual: **P1-EXTRA-Enunciado**).

Se quieren realizar las siguientes pruebas de la aplicación:

Tests unitarios sobre BookRestController (4 puntos)

- Comprobar que se pueden recuperar todos los libros (como usuario sin loggear)
- Comprobar que podemos añadir un nuevo libro (como usuario logeado)
- Comprobar que podemos borrar un libro (como administrador)
- Comprobar que no podemos borrar un libro (como usuario normal)

Tests E2E de la API REST (RESTAssured) (4 puntos)

- Comprobar que se pueden recuperar todos los libros (como usuario sin loggear)
- Comprobar que podemos añadir un nuevo libro (como usuario logeado) y recuperarlo
- Comprobar que podemos borrar un libro (como administrador) y no podemos recuperarlo
- Comprobar que podemos añadir un nuevo libro, editarlo para que en su título termine en v2 y comprobar el resultado al solicitar el libro de nuevo

Para obtener la máxima calificación:

- **Se utiliza TestCointainers (2 punto)** para las pruebas con RESTAssured. Se obtendrá la máxima puntuación si se evita la repetición de código.

Consideraciones

- Los tests **deben ser independientes entre sí** (no depender de información que otros tests hayan creado o eliminado). Además de ser un anti-patrón, JUnit no garantiza el orden de ejecución por defecto.
- **No se pueden utilizar elementos pre-existentes en la base de datos**
- La aplicación utiliza HTTPS y Basic Auth.
- Se valorará la modularización de los tests en paquetes o clases diferentes, dado que son de diferente naturaleza.
- Todos los test deben poder ejecutarse correctamente utilizando el comando `"mvn clean test"`

Material de ayuda

Dado que no se ha explicado en clase cómo manejar los test de API cuando utilizamos autenticación, se proporciona a continuación código de ayuda para estos casos:

Autenticación en RestAssured

Es necesario añadir en el `given()` la autenticación necesaria. En este caso, es necesario poner credenciales existentes en la base de datos.

```
given()
    .auth()
        .basic("user", "pass")
```

Además, al trabajar con aplicaciones seguras que utilizan HTTPS, es necesaria una configuración adicional en RestAssured. Se proporciona a continuación código de ayuda para estos casos:

HTTPS en RestAssured

Es necesario sobrescribir la URL base a la que “atacarán” nuestros test, forzando HTTPS. Además, por defecto, no acepta certificados autofirmados, por lo que es necesario relajar esta restricción.

```
@BeforeEach
public void setUp() {
    RestAssured.port = port;
    RestAssured.useRelaxedHTTPSValidation();
    RestAssured.baseURI = "https://localhost:"+port;
}
```

Formato de entrega

La práctica se entregará teniendo en cuenta los siguientes aspectos:

- La práctica se entregará como un fichero .zip de la aplicación Java junto a los test desarrollados
- El nombre del fichero .zip será el correo URJC del alumno (sin @alumnos.urjc.es).
- El proyecto se puede crear con cualquier editor o IDE.

Las prácticas se podrán realizar de forma individual o por parejas. En caso de que la práctica se haga por parejas:

- Sólo será entregada por uno de los alumnos
- El nombre del fichero .zip contendrá el correo de ambos alumnos separado por guión. Por ejemplo p.perezf-z.gonzalez.zip