

Mockup API Server – Spring Boot

Objetivos:

- Utilizar las funcionalidades básicas del framework Spring Boot.
- Utilizar Spring MVC y Spring REST en nuestras aplicaciones.
- Utilizar JWT para el acceso a las API con Spring Boot.
- Utilizar la librería de Spring Security para controlar la seguridad.

Desarrollo de la práctica.

El estudiante realizará cada uno de los ejercicios indicando en dicho documento interactuando con el servidor de versiones. Las practicas serán utilizando el IDE de su preferencia. Los fuentes serán gestionados en el control de versiones Github.

Forma de Entrega:

La entrega de la práctica se debe realizar un reporte con lo trabajado, conjuntamente con la presentación del mismo en el laboratorio y enviar al servidor de control de versiones. La fecha de entrega está indicada en el PVA. La aplicación debe estar lista para ejecutar con la tarea **run** del plugin **application** en Gradle.

Desarrollo

Fuimos contratados por una empresa de desarrollo de software para mejorar significativamente los procesos y construir software de calidad. Una vez que usted conoce los procesos que tiene implementado la empresa, identifica que los nuevos proyectos para ser desarrollados son manejados por diferentes equipos de desarrolladores y la coordinación y el tiempo de espera para que las funcionalidades que debe realizar un equipo provoca la espera del otro equipo, perjudicando la empresa. Para la solución del problema, usted propone la creación de una herramienta para el bosquejo y consumo de las API (Mockup Server) previamente diseñadas y de esa forma reducir de forma significativa el tiempo muerto entre proyecto, herramienta que cobrará mucha relevancia por el cambio de arquitectura que estará proponiendo basado en Microservicios y Serverless.

Usted habiendo trabajado con los servicios [Mocky](#) y [Mockable.io](#), puede implementar en poco tiempo su propio servicio por las políticas internas que prohíben utilizar servicios de tercero. Dado el escenario planteado realice:

1. El sistema a desarrollar debe utilizar Spring Boot, una aplicación web con capacidad de web responsiva.
2. El sistema utilizará como base de datos H2, utilizando ORM bajo JPA. Una vez inicializado el sistema se debe crear un usuario administrador, el cual será el único que podrá crear usuarios nuevos y asignar roles de permisos a los usuario.
3. Para gestionar (CRUD) los mockups de nuestra API es necesario tener un usuario registrado y los elementos creados serán asociados a ellos, un usuario no debe ver los registro de otro, el administrador si.
4. Para crear un REST Mock, el formulario será muy simple pero validando la información, pidiendo los siguientes valores:
 - (a) Indicar la ruta del endpoint.
 - (b) Método de acceso (Get, Post, Put, Patch, Delete y Options).
 - (c) Headers para la trama de respuesta, indicando key y valor. Puede ninguno, uno o más de uno.
 - (d) Código de respuesta de la petición. Ver tabla código de respuesta HTTP. [Wikipedia](#) o [Mozilla](#).
 - (e) Indicar el Content-Type para trama de respuesta.
 - (f) Indicar el cuerpo del mensaje en la respuesta.
 - (g) Indicar el nombre y la descripción del endpoint.
 - (h) Indicar el tiempo de expiración del endpoint, por defecto un año. Los posibles valores son: 1 mes, 1 semana, 1 día, 1 hora.
 - (i) Indicar el tiempo para la demora de la respuesta, si es marcada, el valor es en segundos.
 - (j) Habilitar la opción para validar vía JWT el acceso al endpoint. El JWT será generado con la fecha de vigencia de la expiración del endpoint. Ver siguiente proyecto https://github.com/vacax/spring_boot_jwt.
5. Los endpoints deben estar asociados aun proyecto.
6. El formulario de creación de los Mock debe implementar i18n para dos idiomas.
7. La seguridad de la gestión de los Mock y creación de los usuarios será controlada vía Spring Security.