



Geeks México

BLOG DE PROGRAMACIÓN EN ESPAÑOL SOBRE JAVA,
FRAMEWORKS, BASES DE DATOS, CÓMPUTO EN LA NUBE, ETC.
EN ESPAÑOL Y EN INGLÉS.

[TUTORIALES EN ESPAÑOL](#)

[TUTORIALS IN ENGLISH](#)

[ABOUT](#)

[CONTACT](#)

Anuncios

Crea una aplicación

Web+REST (Spring boot + AngularJS) en minutos utilizando JHipst er

📅 SEPTIEMBRE 29, 2016 💬 1 COMENTARIO

JHipster es una herramienta de *Scaffolding* creada por Yeoman utilizada para crear proyectos utilizando Spring Boot + AngularJs.

Instalación

Antes de iniciar con la creación de las aplicaciones debemos asegurarnos que tenemos el ambiente de trabajo instalado y configurado correctamente. Para esto debemos seguir los siguientes pasos:

- Instalar y configurar la versión 8 de Java
- Instalar Node.js del [sitio web](#), esto instalará el comando **npm**.
- Instalar Yeoman utilizando el comando **npm install -g yo**
- Instalar Bower utilizando el comando **npm install -g bower**
- Instalar Gulp utilizando el comando **npm install -g gulp-cli**

- Instalar **JHipster** utilizando el comando **npm install -g generator-jhipster**

Con estos sencillos pasos JHipster ha sido instalado de forma exitosa.

Creando una aplicación en minutos utilizando JHipster

Una vez instalado **JHipster** el siguiente paso es crear una aplicación para esto seguiremos los siguientes pasos:

- Crear una carpeta vacía la cuál contendrá el proyecto que se genere:

```
1 | mkdir jhipster-example
```

- Ir al directorio creado utilizando:

```
1 | cd jhipster-example
```

- Crear la aplicación utilizando el comando:

```
1 | yo jhipster
```

JHipster preguntará por lo siguiente:

- Permiso para utilizar estadísticas de uso.
- El tipo de aplicación que se desea construir como se muestra en la imagen:
 - Monolithic application



- El nombre de la aplicación
 - raidentrance
- Nombre del paquete base
 - com.raidentrance.jhipster
- Tipo de autenticación que se desea utilizar,
 - HTTP Session Authentication (stateful, default Spring Security mechanism)
- Tipo de base de datos a utilizar
 - SQL
- Base de datos a utilizar en producción
 - MySQL
- Base de datos a utilizar en desarrollo
 - MySQL
- Si se utilizará el segundo nivel de cache de Hibernate
 - Yes, with ehcache (local cache, for a single node)
- Herramienta para construcción del proyecto
 - para nuestro ejemplo utilizaremos Maven.
- Herramientas adicionales
 - No se selecciona ninguna opción
- Utilizar **LibSass** para el procesamiento de CSS
 - Yes
- Soporte para internacionalización
 - No
- Framework de testing a utilizar
 - Gatling
- Sobre escribir el archivo .gitignore
 - Si

Con estos pasos JHipster generará la aplicación completa y mostrará el siguiente mensaje:

Server app generated successfully.

Client app generated successfully.

```
[09:09:03] Using gulpfile ~/Documents/Github/jhipster-example/gulpfile.js
[09:09:03] Starting 'install'...
[09:09:03] Starting 'inject:test'...
[09:09:03] Starting 'inject:vendor'...
[09:09:03] Starting 'ngconstant:dev'...
[09:09:03] Finished 'install' after 143 ms
[09:09:03] Finished 'ngconstant:dev' after 185 ms
[09:09:03] gulp-inject 1 files into vendor.scss.
[09:09:03] gulp-inject 17 files into karma.conf.js.
[09:09:03] gulp-inject 17 files into index.html.
[09:09:03] Finished 'inject:test' after 175 ms
[09:09:03] Finished 'inject:vendor' after 149 ms
[09:09:03] Starting 'inject:dep'...
[09:09:03] Finished 'inject:dep' after 3.72 µs
[09:09:03] Starting 'sass'...
[09:09:03] Tested 3 tests, 3 passes, 0 failures: PASS
[09:09:03] Finished 'sass' after 386 ms
[09:09:03] Starting 'inject:app'...
[09:09:03] gulp-inject 92 files into index.html.
[09:09:03] Finished 'inject:app' after 128 ms
[09:09:03] Starting 'inject:troubleshoot'...
[09:09:03] gulp-inject 1 files into index.html.
[09:09:03] Finished 'inject:troubleshoot' after 3.53 ms
```

Iniciar la aplicación

Una vez creado el proyecto el siguiente paso es iniciar la aplicación, para esto se deben seguir los siguientes pasos:

- Compilar el proyecto con maven utilizando el comando **mvn clean install**
- Crear una base de datos vacía para el proyecto
- Modificar configuraciones del proyecto, para esto abriremos el archivo **application-dev.yml** ubicado en **src/main/resources/config**. Para este ejemplo se modificará lo siguiente:
 - URL de la base de datos
 - Usuario de la base de datos
 - Password de la base de datos
- Para ejecutar el proyecto es posible hacer lo siguiente:
 - Ejecutar en el IDE la clase RaidentranceApp.java como Spring boot application
 - Utilizar el plugin de Spring boot con el comando **mvn spring-boot:run**
- Abrir en el navegador la dirección <http://127.0.0.1:8080>. Se mostrará lo siguiente:



Puntos a analizar:

- En la parte superior izquierda muestra que se está utilizando el perfil dev
- En la pestaña de Account se tiene la posibilidad de registrar a un usuario nuevo o autenticar uno existente.
- Al seleccionar Sign in se mostrará lo siguiente

Sign in ×

Sign in

Login

Password

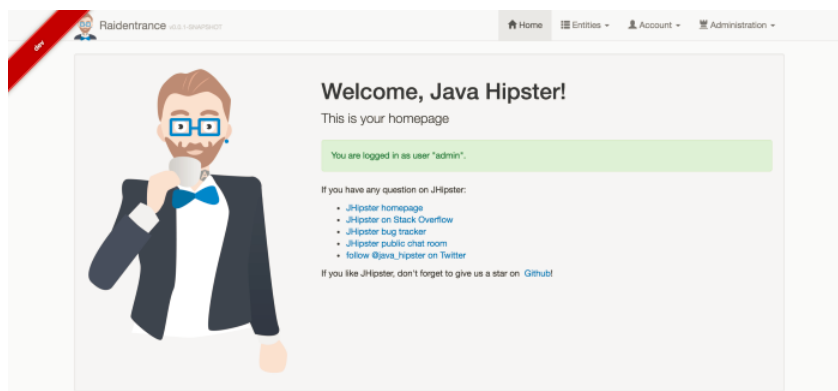
☒ **Remember me**

Sign in

Did you forget your password?

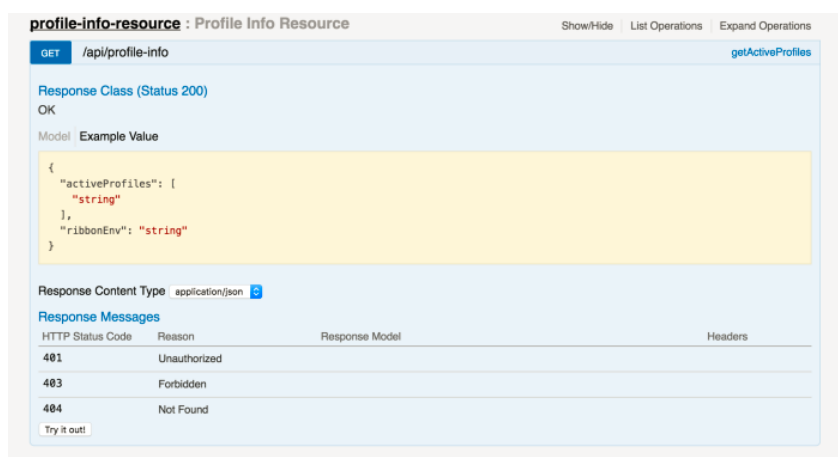
You don't have an account yet? **Register a new account**

- Por default en desarrollo JHipster crea un usuario llamado admin con el password admin, al autenticarte se mostrará la siguiente página de bienvenida:



- En la sección entities se mostrarán las entidades propias del proyecto, en este momento no ha creado ninguna por lo que aparecerá vacía.
- Account muestra lo siguiente:
 - Settings: Permite modificar nombre, apellido y correo electrónico
 - Password: Permite cambiar el password del usuario autenticado
 - Sessions: Muestra las sesiones activas del usuario
 - Sign out: Permite cerrar sesión en la aplicación
- Administración muestra lo siguiente:
 - User management: Permite llevar acabo la administración de usuarios de la aplicación
 - Metrics: Muestra métricas de la aplicación, por ejemplo (número de peticiones http, estadísticas de uso de los web services REST, estadísticas de uso de cache y estadísticas de uso del datasource)
- Health muestra lo siguiente:
 - Muestra los health checks definidos en la aplicación
- Configuration
 - Muestra todas las configuraciones de la aplicación
- Audits

- Muestra los registros para hacer auditoría en el sistema
- Logs
 - Muestra y permite cambiar los niveles de log de la aplicación
- API
 - Muestra los web services de la aplicación así como una explicación de como utilizarlos como se muestra en la siguiente imagen



Toda la funcionalidad mostrada fue creada siguiendo buenas prácticas de desarrollo solo ejecutando un solo comando y llenando un wizard sencillo.

En el siguiente POST se explicará cómo crear nuevas páginas en minutos, la idea será crear una aplicación completa en minutos.

Subiré el código completo en el siguiente post con el fin de que esté completo.

Autor: Alejandro Agapito Bautista

Twitter: [@raidentrance](#)

(<https://geeksjavamexico.wordpress.com/mentions/raidentrance/>)

Contacto:raidentrance@gmail.com

Anuncios



1 comentario »



Pingback: [5 Comandos, una aplicación web completa con JHipster en español – Geeks Java México](#)
