

Laboratorio 4

En este laboratorio, usará la base de un Proyecto Spring MVC ya creado para ver su estructura, y hacer una pequeña modificación

1. Descargue el archivo de files.zip del laboratorio. Descomprímalo.
2. Usando su explorador del sistema o la consola, localice el directorio .m2. Debería estar en el directorio de su usuario (en Windows, C:\Users\[SU-USUARIO], en MacOS o Linux en algo como /usr/[usuario])... téngalo abierto.
3. Descomprímalo. Luego abra STS.
4. En STS, seleccione File → Import → MAVEN → Existing Maven Project. Apunte al directorio raíz del archivo descomprimido. Debe salirle al seleccionarlo un checkbox indicando además el pom.xml
5. Proceda. Al iniciar el import, observe si hay cambios en .m2.
6. Responda las siguientes preguntas:
 - a. Qué es y para que sirve el archivo web.xml
 - b. Qué es un listener en java web development.
 - c. Qué es un servlet? Qué especifican las etiquetas servlet, servlet-name, servlet-class en web.xml?
 - d. Para qué sirve el archivo root-context.xml?
 - e. Para qué sirve el archivo servlet-context.xml?
 - f. En servlet-context:
 - i. Para qué sirve la etiqueta annotation-driven?
 - ii. Para qué se configura el bean de la clase InternalResourceViewResolver? Qué archivos están involucrados y por qué?
 - iii. Para qué sirve la etiqueta context:component-scan? Qué configura?
 - g. En el archivo root-context.xml:
 - i. Qué relación tiene los xsi:schemaLocation con las etiquetas context:component-scan y mvc:annotation-driven ?
 - ii. Las características y atributos de cada etiqueta de las mencionadas
 - iii. Qué se define en bean? Y qué relación tiene con el archivo home.jsp?
 - h. En la clase HomeController.java,
 - i. para qué sirve la anotación @Controller?
 - ii. Para qué sirve la anotación @Autowired
 - iii. Para qué sirve la anotación @RequestMapping y sus atributos?
 - iv. Qué es la clase Model referida? Y Locale?
7. Ponga a correr el Tomcat externo.
8. Usando el Project Explorer, revise el directorio /target del proyecto. Ahí haga click derecho y seleccione Show in → System Explorer. TOME EL ARCHIVO .WAR de ahí. Y cópielo al directorio webapps donde copió el war del laboratorio anterior. Observe los cambios en la consola.
9. Ahora vaya a <http://localhost:8080/componentes-1.0.0-BUILD-SNAPSHOT/>
10. Observe el resultado y compárelo con el archivo /src/webapp/WEB-INF/home.jsp
11. Retos:

- a. Agregue un nuevo valor en el model definido en el controller de Home y modifique home.jsp para desplegarlo. Tendrá que hacer un Run as → Maven Build y agregar la opción package como en el laboratorio anterior.
- b. Agregue una nueva página. Llámela como quiera y logre que se despliegue. Tendrá que crear un nuevo controller y mapping.