SPRING BOOT + SECURITY + BOOTSTRAP

# SEGURIDAD CON SPRING BOOT + THYMELEAF Y MYSQL. CAPÍTULO 7. VISTAS DE PRODUCTOS

Dentro de la carpeta templetes habíamos creado otra de nombre producto. En esta crearemos 4 vistas:

* detalle
* editar
* lista
* nuevo

**lista.html:**

Aquí mostramos una tabla con todos los productos. Si el usuario es administrador se verán los botones de nuevo, actualizar y eliminar:

|  |  |
| --- | --- |
|  | <!DOCTYPE html> |
|  | <html xmlns:th="http://www.thymeleaf.org" |
|  | xmlns:sec="http://www.thymeleaf.org/extras/spring-security"> |
|  | <head th:insert="fragments.html :: headerfiles"></head> |
|  | <body> |
|  | <header th:insert="fragments.html :: nav"></header> |
|  | <div class="container"> |
|  | <table class="table table-dark" th:if="${not #lists.isEmpty(productos)}"> |
|  | <thead> |
|  | <tr> |
|  | <th scope="col">ID</th> |
|  | <th scope="col">Nombre</th> |
|  | <th scope="col">Precio</th> |
|  | <th scope="col">Ver</th> |
|  | <th scope="col" sec:authorize="hasRole('ROLE\_ADMIN')">Editar</th> |
|  | <th scope="col" sec:authorize="hasRole('ROLE\_ADMIN')">Eliminar</th> |
|  | </tr> |
|  | </thead> |
|  | <tbody> |
|  | <tr th:each="producto: ${productos}"> |
|  | <td th:text="${producto.id}">id</td> |
|  | <td th:text="${producto.nombre}"></td> |
|  | <td th:text="${producto.precio} + ' €'"></td> |
|  | <td><a th:href="@{'/producto/detalle/' + ${producto.id}}"><button class="btn btn-primary"><i class="fa fa-eye"></i></button></a></td> |
|  | <td sec:authorize="hasRole('ROLE\_ADMIN')"><a th:href="@{'/producto/editar/' + ${producto.id}}"><button class="btn btn-warning"><i class="fa fa-edit"></i></button></a></td> |
|  | <td sec:authorize="hasRole('ROLE\_ADMIN')"><a th:href="@{'/producto/eliminar/' + ${producto.id}}"><button class="btn btn-danger"><i class="fa fa-trash"></i></button></a></td> |
|  | </tr> |
|  | </tbody> |
|  | </table> |
|  | <a sec:authorize="hasRole('ROLE\_ADMIN')" th:href="@{/producto/nuevo}"><button class="btn btn-success btn-lg"><i class="fas fa-plus-circle"></i> Nuevo Producto</button></a> |
|  | </div> |
|  | <footer th:insert="fragments.html::footer"></footer> |
|  | </body> |
|  | </html> |

[**view raw**](https://gist.github.com/cavanosa/902126e7574b15f15149778063d0133d/raw/adf231032f1a3342649c90f981da14824c40f3b9/lista.html)[**lista.html**](https://gist.github.com/cavanosa/902126e7574b15f15149778063d0133d#file-lista-html) hosted with  by [**GitHub**](https://github.com/)

Con thymeleaf security es muy sencillo comprobar si el usuario tiene privilegios de administrador:

|  |
| --- |
| sec:authorize="hasRole('ROLE\_ADMIN')" |

**detalle.html:**

En esta vista se muestra un producto al hacer click en el botón ver de la lista:

|  |  |
| --- | --- |
|  | <!DOCTYPE html> |
|  | <html xmlns:th="http://www.thymeleaf.org" xmlns:sec="http://www.thymeleaf.org/extras/spring-security"> |
|  | <head th:insert="fragments.html :: headerfiles"></head> |
|  | <body> |
|  | <header th:insert="fragments.html :: nav"></header> |
|  | <div class="container"> |
|  | <p><b>Nombre del producto</b></p><p th:text="${producto.nombre}"></p><br/> |
|  | <p><b>Precio del producto</b></p><p th:text="${producto.precio} + ' €'"></p><br/> |
|  | <td><a th:href="@{/producto}"><button class="btn btn-info" ><i class="far fa-arrow-alt-circle-left"></i> Volver</button></a></td> |
|  | </div> |
|  | <footer th:insert="fragments.html::footer"></footer> |
|  | </body> |
|  | </html> |

[**view raw**](https://gist.github.com/cavanosa/02b287ae01233f5a98ba1672297c8c7b/raw/4a9bcdacaf83612f4ca63c2ec7f6cc927da15d9f/detalle.html)[**detalle.html**](https://gist.github.com/cavanosa/02b287ae01233f5a98ba1672297c8c7b#file-detalle-html) hosted with  by [**GitHub**](https://github.com/)

**nuevo.html:**

|  |  |
| --- | --- |
|  | <!DOCTYPE html> |
|  | <html xmlns:th="http://www.thymeleaf.org" xmlns:sec="http://www.thymeleaf.org/extras/spring-security"> |
|  | <head th:insert="fragments.html :: headerfiles"></head> |
|  | <body> |
|  | <header th:insert="fragments.html :: nav"></header> |
|  | <div class="container"> |
|  | <div class="page-header"> |
|  | <h1>Nuevo Producto</h1> |
|  | </div> |
|  | <p th:if="${error}" class="text text-danger" th:text="${error}"></p> |
|  |  |
|  | <form th:action="@{/producto/crear}" method="post" class="form-horizontal"> |
|  | <!-- protección csrf --> |
|  | <input type="hidden" th:name="${\_csrf.parameterName}" th:value="${\_csrf.token}" /> |
|  | <!-- fin csrf --> |
|  | <div class="form-group"> |
|  | <label for="nombre" class="col-sm-2 control-label">Nombre del Producto</label> |
|  | <div class="col-sm-10"> |
|  | <input type="text" class="form-control" name="nombre" id="nombre" |
|  | placeholder="nombre..." required="required"/> |
|  | </div> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label for="precio" class="col-sm-2 control-label">Precio (Euros)</label> |
|  | <div class="col-sm-10"> |
|  | <input type="number" class="form-control" name="precio" id="precio" |
|  | required="required"/> |
|  | </div> |
|  | </div> |
|  | <div class="form-group"> |
|  | <div class="col-sm-offset-2 col-sm-10"> |
|  | <button type="submit" class="btn btn-outline-success btn-lg">Crear</button> |
|  | </div> |
|  | </div> |
|  | </form> |
|  |  |
|  | </div> |
|  | <footer th:insert="fragments.html::footer"></footer> |
|  | </body> |
|  | </html> |

[**view raw**](https://gist.github.com/cavanosa/1b162cb2b58e85a3198dc886c7c9c965/raw/001a10cf05b8898167724ef596e7bdc98f97bd36/nuevo.html)[**nuevo.html**](https://gist.github.com/cavanosa/1b162cb2b58e85a3198dc886c7c9c965#file-nuevo-html) hosted with  by [**GitHub**](https://github.com/)

Como al crear un nuevo producto se ejecuta un método POST, también debemos icluir aquí los csrf token.

En caso de error (valores erróneos en los campos) se mostrará el correspondiente mensaje:

|  |
| --- |
| th:if="${error}" class="text text-danger" th:text="${error}" |

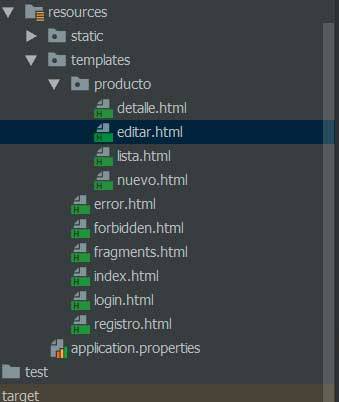
**editar.html:**

Y por último el formulario de edición, muy similar al de crear un producto nuevo:

|  |  |
| --- | --- |
|  | <!DOCTYPE html> |
|  | <html xmlns:th="http://www.thymeleaf.org" xmlns:sec="http://www.thymeleaf.org/extras/spring-security"> |
|  | <head th:insert="fragments.html :: headerfiles"></head> |
|  | <body> |
|  | <header th:insert="fragments.html :: nav"></header> |
|  | <div class="container"> |
|  | <div class="page-header"> |
|  | <h1>Editar Producto</h1> |
|  | </div> |
|  | <p th:if="${error}" class="text text-danger" th:text="${error}"></p> |
|  |  |
|  | <form th:action="@{/producto/actualizar}" method="post" class="form-horizontal"> |
|  | <div class="form-group"> |
|  | <!-- protección csrf --> |
|  | <input type="hidden" th:name="${\_csrf.parameterName}" th:value="${\_csrf.token}" /> |
|  | <!-- fin csrf --> |
|  | <label for="nombre" class="col-sm-2 control-label">Nombre del Producto</label> |
|  | <div class="col-sm-10"> |
|  | <input type="text" class="form-control" name="nombre" id="nombre" |
|  | required="required" th:value="${producto.nombre}" /> |
|  | </div> |
|  | </div> |
|  | <div class="form-group"> |
|  | <label for="precio" class="col-sm-2 control-label">Precio (Euros)</label> |
|  | <div class="col-sm-10"> |
|  | <input type="number" class="form-control" name="precio" id="precio" |
|  | required="required" th:value="${producto.precio}" /> |
|  | </div> |
|  | </div> |
|  | <input type="hidden" id="id" name="id" th:value="${producto.id}" /> |
|  | <div class="form-group"> |
|  | <div class="col-sm-offset-2 col-sm-10"> |
|  | <button type="submit" class="btn btn-outline-success btn-lg">Actualizar</button> |
|  | </div> |
|  | </div> |
|  | </form> |
|  |  |
|  | </div> |
|  | <footer th:insert="fragments.html::footer"></footer> |
|  | </body> |
|  | </html> |

[**view raw**](https://gist.github.com/cavanosa/cdc39bc96ab2263e9191fcc2e9b60251/raw/992c9a0f045ba3645ce9d251f2a774fadd5c6a8e/editar.html)[**editar.html**](https://gist.github.com/cavanosa/cdc39bc96ab2263e9191fcc2e9b60251#file-editar-html) hosted with  by [**GitHub**](https://github.com/)

Y con esto hemos terminado de escribir el código del proyecto. El árbol de archivos de templates lo deberíamos tener así:



En el siguiente post generaremos el archivo war.