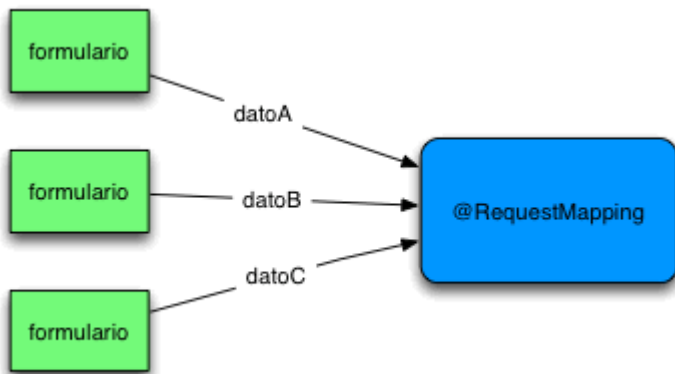


Spring MVC `@RequestMapping` es una de las anotaciones más usada en Spring MVC. Sin embargo en muchas casuísticas nos olvidamos de las opciones que soporta. En este caso vamos a revisar un poco como se pueden gestionar los diversos parámetros.



Partimos de un Controlador que nos redirecciona a un conjunto de formularios .

```
@Controller
@RequestMapping("/ControladorPetición")
public class ControladorPetición {

    @RequestMapping("/formularioPeticiones")
    public String formulario(Model modelo) {

        return "formularioPeticiones";

    }
}
```

En este caso disponemos de dos formularios que realizan diferentes peticiones dependiendo de lo que se introduzca en ellos (hemos omitido Spring Tags por simplicidad).

---

Nombre:

Nombre:

Lenguaje:

Nivel:

Vamos a ver su código :

```
&lt;!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd"&gt;
&lt;html&gt;
&lt;head&gt;
&lt;meta http-equiv="Content-Type" content="text/html;
charset=UTF-8"&gt;
&lt;title&gt;Insert title here&lt;/title&gt;
&lt;/head&gt;
&lt;body&gt;

&lt;form action="resultadoPeticiones"&gt;
  Nombre:&lt;input type="text" name="nombre"&gt;

  &lt;input type="submit" /&gt;
&lt;/form&gt;
&lt;br/&gt;
&lt;form action="resultadoPeticiones" method="post"&gt;
Nombre:&lt;input type="text" name="nombre"&gt;
&lt;br/&gt;
Lenguaje:&lt;select name="lenguaje"&gt;
```

```

&lt;option&gt;selecciona&lt;/option&gt;
&lt;option&gt;Java&lt;/option&gt;
&lt;option&gt;NET&lt;/option&gt;
&lt;/select&gt;
&lt;br/&gt;
Nivel:&lt;select name="nivel"&gt;
&lt;option&gt;selecciona&lt;/option&gt;
&lt;option&gt;1&lt;/option&gt;
&lt;option&gt;2&lt;/option&gt;
&lt;/select&gt;
&lt;br/&gt;
&lt;input type="submit" /&gt;
&lt;/form&gt;
&lt;/body&gt;
&lt;/html&gt;

```

Rellenamos el primer formulario y enviamos los datos para que se procesen. El controlador deberá añadir un nuevo método para gestionar la petición GET que acabamos de realizar:

```

@RequestMapping("/resultadoPeticiones")
public String formularioPeticiones(Model modelo) {

    modelo.addAttribute("mensaje","peticion GET");
    return "resultadoPeticiones";

}

```

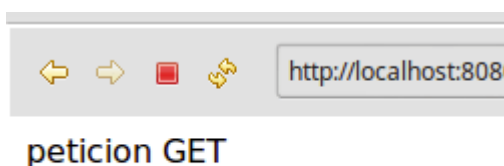
La petición se procesará y nos enviará a la pagina “resultadoPeticiones”

```

&lt;%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
&lt;!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
&lt;html&gt;
&lt;head&gt;
&lt;meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
&lt;title&gt;Insert title here&lt;/title&gt;
&lt;/head&gt;
&lt;body&gt;
    ${mensaje}
&lt;/body&gt;
&lt;/html&gt;

```

La cual imprime:



Podemos capturar una petición similar de tipo POST enviando el segundo formulario y añadiendo a nuestro controlador otro método con el mapeo correspondiente.

```

@RequestMapping(value="/resultadoPeticiones",
method=RequestMethod.POST)

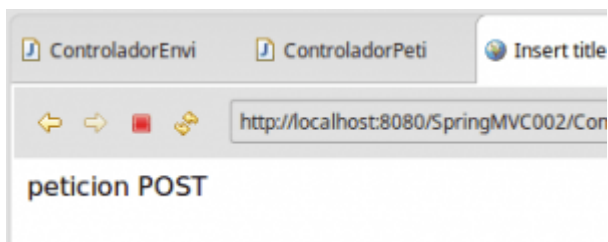
```

```
public String formularioPeticionesPOST(Model modelo) {

    modelo.addAttribute("mensaje","peticion POST");
    return "resultadoPeticiones";

}
```

Hemos usado la propiedad “method” para solo aceptar de tipo POST el resultado es:



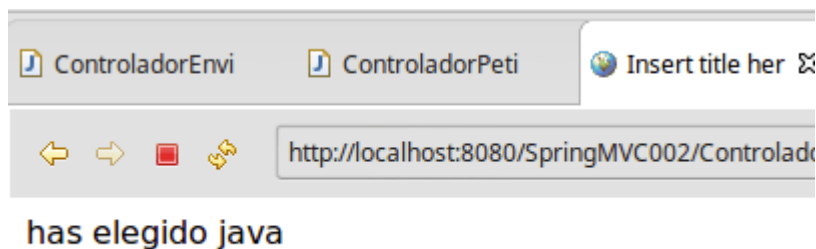
Existen más opciones ya que por ejemplo si seleccionamos de lenguaje de programación Java en el combo, podremos procesar la petición a través del siguiente método utilizando la propiedad de params de nuestra anotación:

```
@RequestMapping(value="/resultadoPeticiones",method=RequestMethod.POST
,params={"lenguaje=Java"})
public String formularioPeticionesLenguajeJava(Model modelo) {

    modelo.addAttribute("mensaje","has elegido java");
    return "resultadoPeticiones";

}
```

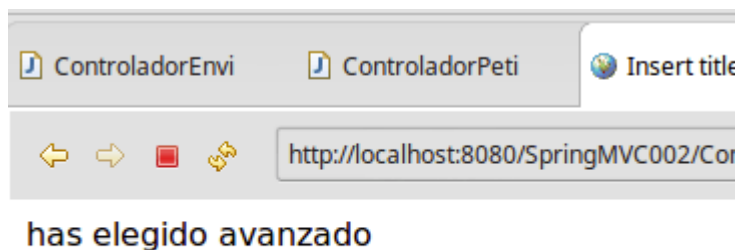
El resultado será:



Incluso podemos realizar una combinación de parámetros y gestionar una petición en la que se haya seleccionado Java como lenguaje y nivel 2.

```
@RequestMapping(value="/resultadoPeticiones",  
method=RequestMethod.POST,params={"nivel=2", "lenguaje=Java"})  
public String formularioPeticionesLenguaje(Model modelo) {  
  
    modelo.addAttribute("mensaje","has elegido avanzado");  
    return "resultadoPeticiones";  
  
}
```

El resultado será:



Como vemos Spring MVC @RequestMapping soporta muchas configuraciones posibles, aprendamos a usarlas y simplificaremos bastante nuestros controladores.

Acabamos de ver como usar Java equals y hashCode

Otros artículos relacionados :

[Spring MVC Configuración](#)

[Spring MVC Anotaciones](#)

[Spring MVC Model Attribute](#)