Spring MVC

Locales

- Spring MVC tiene soporte para internacionalización de aplicaciones
  - *DispatcherServlet* permite resolver automáticamente los mensajes usando el *locale* del cliente
  - Este trabajo se realiza con los objetos *LocaleResolver*
- Cuando se recibe un requerimiento *DispatcherServlet* busca un *LocaleResolver*
  - Si encuentra uno, lo intenta usar para asignar un *locale*
  - Se puede usar *RequestContext.getLocale()* para obtener el *locale* que fue resuelto

Locales

- Aparte de resolver automáticamente el *locale* se puede agregar un *Interceptor*
  - Útil para cambiar el *locale* bajo ciertas circunstancias, como algún valor especial de parámetro de request

```
<!-- Para la internacionalización del sitio se crea un interceptor que revisara el parámetro 'lang' -->
<mvc:interceptors>
    <bean class="org.springframework.web.servlet.i18n.LocaleChangeInterceptor">
        <property name="paramName" value="lang" />
    </bean>
</mvc:interceptors>

<!-- El default locale del sitio es configurado a 'es' y es almacenado en la sesión de usuario -->
<bean id="localeResolver" class="org.springframework.web.servlet.i18n.SessionLocaleResolver">
    <property name="defaultLocale" value="es" />
</bean>
```

Soporte para recibir archivos

- Spring provee soporte para recibir archivos
  - Se debe habilitar este soporte con un *MultipartResolver* en el contexto de la aplicación
  - Spring provee una implementación con *Commons FileUpload*

    http://commons.apache.org/proper/commons-fileupload/
  - Cada request será revisado por si contiene un "multipart"
    - Si no se encuentra, el request continua su camino
    - Si se encuentra, el resolver realizará su trabajo

MultipartResolver

- Ejemplo de como usar *CommonsMultipartResolver*:

```
<bean id="multipartResolver"
    class="org.springframework.web.multipart.commons.CommonsMultipartResolver">

    <!-- one of the properties available; the maximum file size in bytes -->
    <property name="maxUploadSize" value="100000"/>
</bean>
```

  - Es necesario agregar *commons-fileupload.jar* al classpath
  - El resolver, actúa convirtiendo el *HttpServletRequest* en un *MultipartHttpServletRequest*

Formulario de Upload

```
<html>
    <head>
        <title>Upload a file please</title>
    </head>
    <body>
        <h1>Please upload a file</h1>
        <form method="post" action="/form" enctype="multipart/form-data">
            <input type="text" name="name"/>
            <input type="file" name="file"/>
            <input type="submit"/>
        </form>
    </body>
</html>
```

Controlador para recibir archivo

```java
@Controller
public class FileUploadController {

    @RequestMapping(value = "/form", method = RequestMethod.POST)
    public String handleFormUpload(@RequestParam("name") String name,
        @RequestParam("file") MultipartFile file) {

        if (!file.isEmpty()) {
            byte[] bytes = file.getBytes();
            // store the bytes somewhere
           return "redirect:uploadSuccess";
       } else {
           return "redirect:uploadFailure";
       }
    }

}
```