

Internacionalización basada en base de datos

Curso intermedio de Thymeleaf

Conceptos y configuración

- **Idénticos** a la i18n basada en ficheros
 - Locale
 - Mensajes
 - LocaleResolver
 - LocaleChangeInterceptor
- La utilización en la plantilla **Thymeleaf** es también **idéntica**
- **¿Qué cambia pues?**

¿Cómo obtiene Spring Boot + Thymeleaf el mensaje?

- Por defecto, Spring Boot utiliza una instancia de *ResourceBundleMessageSource*.
- Según la configuración de Spring Boot, se espera que los mensajes sean obtenidos por un bean llamado, literalmente, *messageSource*.
- Una forma de modificar el comportamiento es *solapar dicho bean con uno de nuestra cosecha*.
- En lugar de implementar la interfaz *MessageSource*, podemos extender la clase *AbstractMessageSource*.

Modelo

- Podemos crear **una entidad** que maneje los mensajes en diferentes idiomas.
- Añadimos su esquema al script sql
- Añadimos un repositorio con la consulta adecuada.

DatabaseDrivenMessageSource

- Implementamos el método *resolveCode* para que lo **busque en la base de datos usando el repositorio**.
- Debemos devolver una instancia de *MessageFormat*.
- ¡Importante! Para que este componente solape al MessageSource por defecto, lo *qualificaremos*.

Retos

- Completar los mensajes en la base de datos para toda la interfaz.
- Implementar un pequeño panel de control para manejar estos mensajes (formulario de creación, listado, edición, borrado, ...).
- Establecer el `DEFAULT_LOCALE` a través del fichero *application.properties*