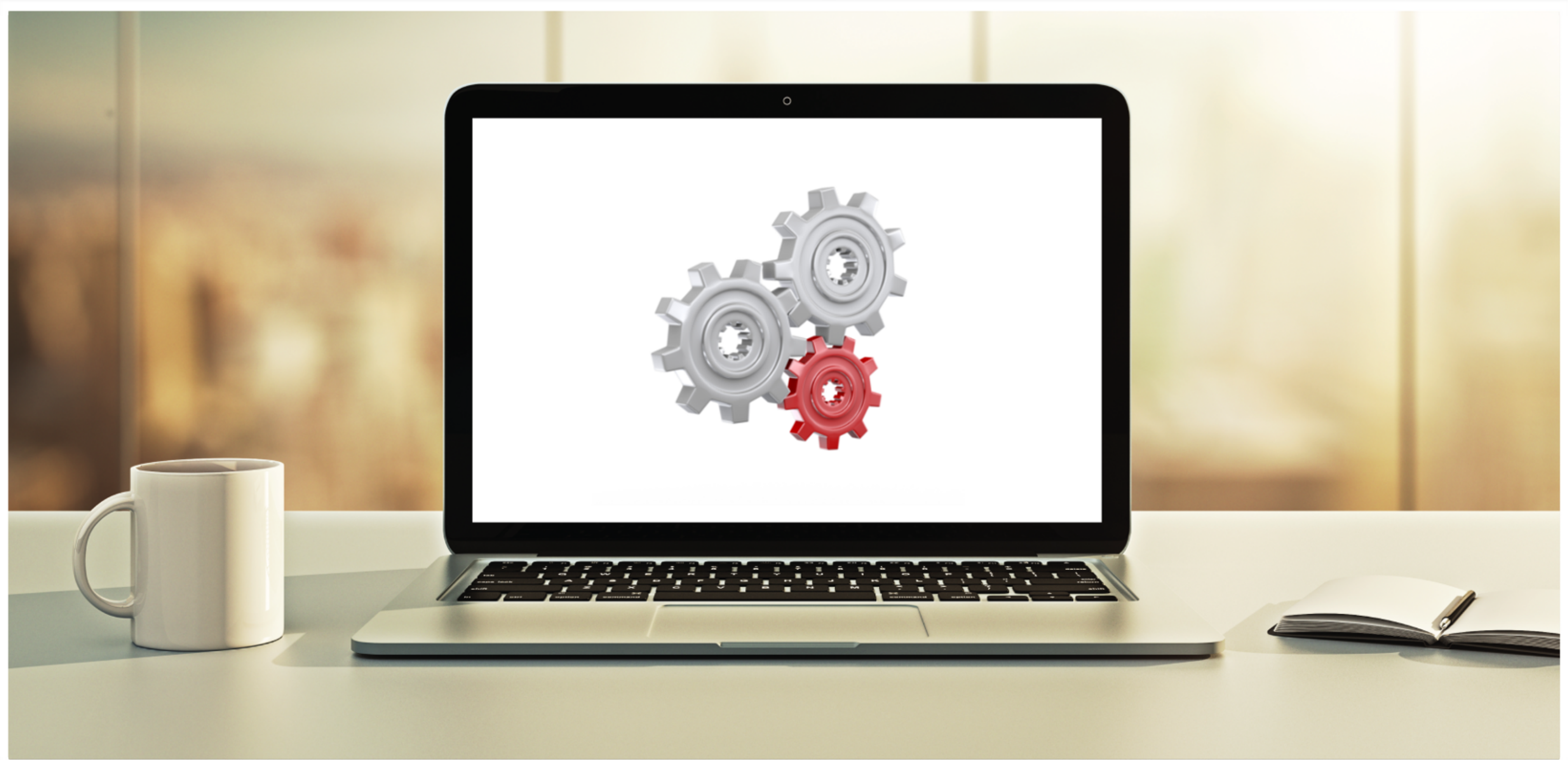# JSTL I18N/Formatting Tags

Apply Internationalization

Build a Multi-Lingual Application

luv2code

Multi-Lingual Application

English (US) | Spanish (ES) | German (DE)

Howdy

First Name *John*
Last Name *Doe*

Welcome to the training class.

Selected locale: en_US

English (US) | Spanish (ES) | German (DE)

Hola

Nombre de pila *John*
Apellido *Doe*

Bienvenidos a la clase de formación.

Selected locale: es_ES

English (US) | Spanish (ES) | German (DE)

Hallo

Vorname *John*
Nachname *Doe*

Willkommen in der Ausbildung Klasse.

Selected locale: de_DE

Formatting Messages

labels / placeholders

[greet]

[first_name]: John
[last_name]: Doe

[welcome_message]

English (US) | Spanish (ES) | German (DE)

Hola

Nombre de pila *John*
Apellido *Doe*

Bienvenidos a la clase de formación.

Selected locale: es_ES

To Do List

- Step 1: Create Resource Files

  - Translated versions of your labels

- Step 2: Create JSP Page with labels

- Step 3: Update JSP page to change locale based on user selection

Step 1: Create Resource Files

- Translated versions of your labels

  - You will need to use Google Translate or similar translator service

- Must name the file using the **locale**

What is a Locale?

Locale = Language Code + Country Code

- There are differences for a language depending on the country

- For example

  - English (US)

  - English (GB)

Step 1: Create Resource Files

- File name must follow specific format

  - **<your project file name>_LANGUAGECODE_COUNTRYCODE.properties**

- Examples:

  - mylabels_es_ES.properties

  - mylabels_de_DE.properties

  - mylabels_en_GB.properties

Step 1: Create Resource Files

- Language Codes based on ISO Standard 639-1 (2 letter code)

  - https://en.wikipedia.org/wiki/ISO_639-1

- Country Codes based on ISO Standard 3166-2 (2 letter code)

  - https://en.wikipedia.org/wiki/ISO_3166-2

Step 1: Create Resource Files

- Here's an example for the locale: **Spanish - Spain**

File: mylabels_es_ES.properties

```
label.greeting=Hola
label.firstname=Nombre de pila
label.lastname=Apellido
label.welcome=Bienvenidos a la clase de formación.
```

Step 2: Create JSP page with labels

File: i18n-messages-test.jsp

```
<fmt:message key="label.greeting" /> <br/><br/>

<fmt:message key="label.firstname" /> <i>John</i> <br/>

<fmt:message key="label.lastname" /> <i>Doe</i> <br/><br/>

<fmt:message key="label.welcome" /> <br/>
```

Howdy

First Name *John*
Last Name *Doe*

Welcome to the training class.

Hola

Nombre de pila *John*
Apellido *Doe*

Bienvenidos a la clase de formación.

Step 3: Update JSP page to change locale based on user selection

- Our JSP page will send links to itself

- Will pass the locale as a parameter

English (US) | Spanish (ES) | German (DE)

Howdy

First Name *John*

Step 3: Update JSP page to change locale based on user selection

```
<a href="i18n-messages-test.jsp?theLocale=en_US">English (US)</a>
 |
<a href="i18n-messages-test.jsp?theLocale=es_ES">Spanish (ES)</a>
 |
<a href="i18n-messages-test.jsp?theLocale=de_DE">German (DE)</a>
```

English (US) | Spanish (ES) | German (DE)
_____

Howdy

First Name *John*

~~Last Name Doe~~

Step 3: Update JSP page to change locale based on user selection

```jsp
<c:set var="theLocale"
    value="${not empty param.theLocale ? param.theLocale : pageContext.request.locale}"
    scope="session" />

<fmt:setLocale value="${theLocale}" />

<fmt:setBundle basename="com.luv2code.jsp.tagdemo.i18n.resources.mylabels" />
```

JSTL I18N/Formatting Tags