Linguae

Dealing with languages in Spring applications

Vladimir Ritz Bossicard (vbossica@gmail.com)

In this day and age it is mandatory to write applications that seamlessly support multiple languages. This project explores possible solutions to some internationalization (i18n) and localization (l10n) problems in the context of a Spring based application.

Are you translated, yet?

When developing a web application in Java (and certainly in other technologies), it has been the practice to place the translated messages and labels inside the `WEB-INF` folder. This practice greatly facilitates the deployment (everything is inside the `WAR` file) but has the main disadvantage of forcing a new release for every change made to the translation assets (new supported language, typo). Furthermore the exhaustive list of messages to be translated is only known at the end of a development cycle, which usually leaves little or no time to the translators to finish their job before the product is being shipped.

The idea that has been explored was to leverage the Spring framework to offer the possibility to easily add and reload a language into the web application.

File System Provider

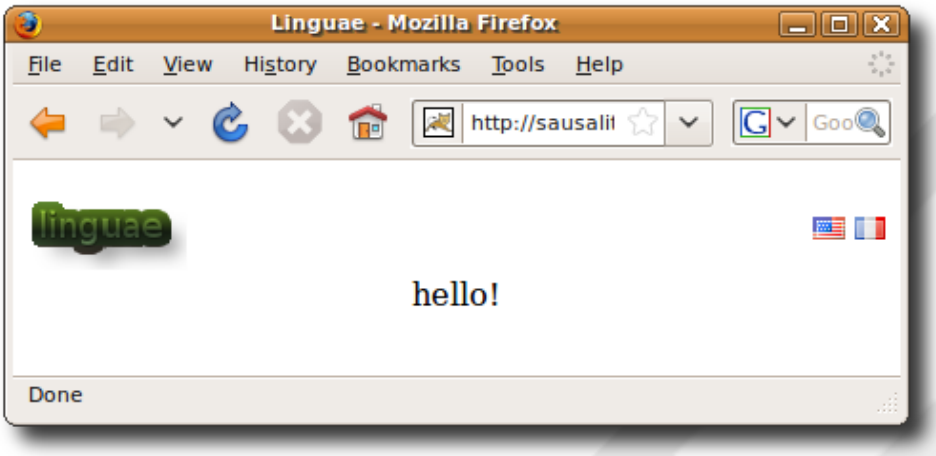
One obvious location for placing the properties file is of course directly on the file system and to auto-discover them with the `FilesystemDiscoveryMessageSource` class:

Figure 1. FilesystemDiscoveryMessageSource configuration

```
<bean name="messageSource"
        class="org.workingonit.linguae.FilesystemDiscoveryMessageSource">
    <property name="checkMessage" value="org.wkg.linguae.locale"/>
    <property name="basename" value="file:/opt/acme-lingua/i18n/labels"/>
</bean>
```

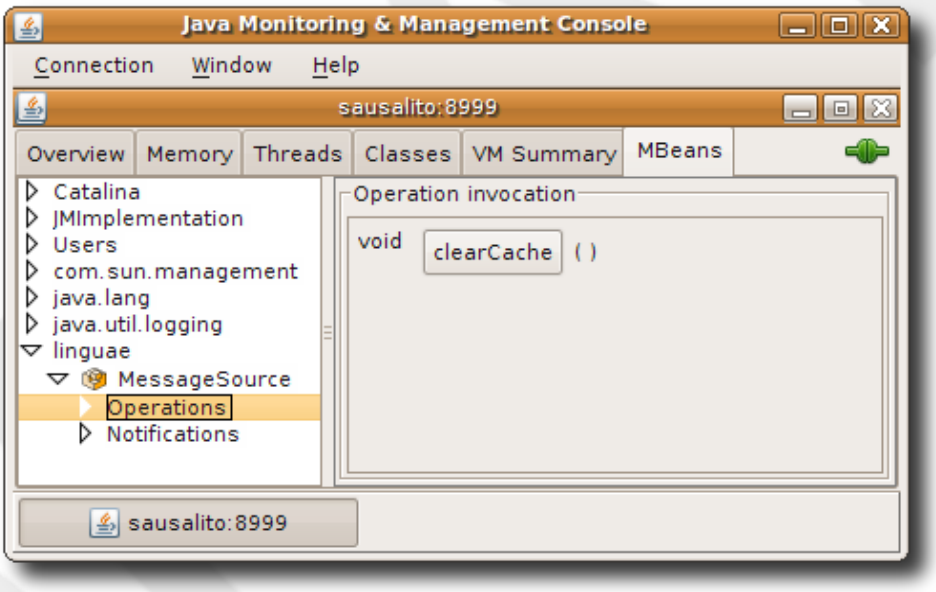
At startup, each language is tested by looking for the `org.wkg.linguae.locale` label. Since the list of available languages is automatically updated, it is trivial to let the user only select available (or delivered) languages:

Figure 2. Sample web application



So what happens when some new language is added? The application could of course be restarted but a more elegant solution would be to use JMX to manage the `FilesystemDiscoveryMessageSource`:

Figure 3. Reloading with a JMX client



Database Provider

Some people have suggested to store *translation data into a database* [http://forum.springsource.org/showthread.php?t=18194] to ease the maintenance. Since it is quite unlikely that these values will be changed in a production environment and in order to take advantage of the caching mechanism of the Spring `ReloadableResourceBundleMessageSource`, the `DatabaseDiscoveryMessageSource` class will simply be a specialization of the `FilesystemDiscoveryMessageSource`

Figure 4. DatabaseDiscoveryMessageSource configuration

```
<bean name="dbMessageSource"
        class="org.workingonit.linguae.DatabaseDiscoveryMessageSource">
    <property name="checkMessage" value="org.wkg.linguae.locale"/>
    <property name="basename" value="file:/opt/acme-lingua/i18n/labels"/>
    <property name="messageDao">
        <bean class="org.workingonit.linguae.dao.MessageDao">
            <property name="dataSource">
                <bean class="org.springframework.jdbc.datasource.DriverManagerDataSource">
                    <property name="driverClassName" value="oracle.jdbc.OracleDriver"/>
                    <property name="url" value="jdbc:oracle:thin:@sausalito:1521:XE"/>
                    <property name="username" value="scott"/>
                    <property name="password" value="tiger"/>
                </bean>
            </property>
        </bean>
    </property>
</bean>
```

And will first generate the properties files from the database before loading them as seen in the previous example:

Figure 5. Database autodiscovery

```
DatabaseDiscoveryMessageSource - loading messages for locale 'fr' into
    /opt/acme-lingua/i18n/labels_fr.properties
AbstractDiscoveryMessageSource - locale 'fr' was found
DatabaseDiscoveryMessageSource - loading messages for locale 'en' into
    /opt/acme-lingua/i18n/labels_en.properties
AbstractDiscoveryMessageSource - locale 'en' was found
```