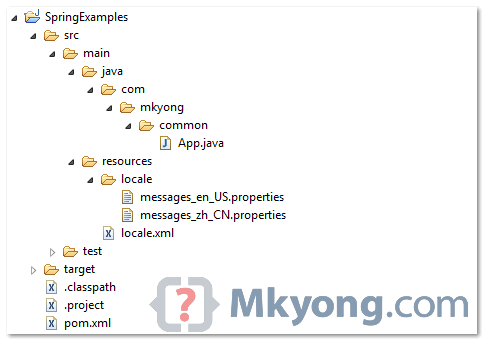
In Spring, you can use ResourceBundleMessageSource to resolve text messages from properties file, base on the selected locales. See following example :

1. Directory Structure

Review directory structure of this example.



2. Properties file

Create two properties files, one for English characters (messages\_en\_US.properties), other one for Chinese characters (messages\_zh\_CN.properties). Put it into the project class path (see figure above).

*File : messages\_en\_US.properties*

Bash

customer.name=Yong Mook Kim, age : {0}, URL : {1}

*File : messages\_zh\_CN.properties*

Bash

customer.name=\ufeff\u6768\u6728\u91d1, age : {0}, URL : {1}

The ‘**\ufeff\u6768\u6728\u91d1**‘ is Unicode characters in Chinese.

**Note**  
To display the Chinese characters correctly, you have to use “[native2ascii](http://www.mkyong.com/java/java-convert-chinese-character-to-unicode-with-native2ascii/)” tool to convert the Chinese characters into Unicode characters.

3. Bean configuration file

Include the properties file into the bean configuration file. Both “**messages\_en\_US.properties**” and “**messages\_zh\_CN.properties**” are consider one file in Spring, you just need to include the file name once, and Spring will find the correct locale automatically.

Markup

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans-2.5.xsd">

<bean id="messageSource" class="org.springframework.context.support.ResourceBundleMessageSource"> <property name="basename"> <value>locale\customer\messages</value>

</property>

</bean>

</beans>

*P.S Assume both files are located at “resources\locale\customer\” folder.*

4. Run it

Java

package com.mkyong.common;

import java.util.Locale;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class App {

public static void main(String[] args) {

ApplicationContext context = new ClassPathXmlApplicationContext("locale.xml");

String name = context.getMessage("customer.name", new Object[] { 28,"http://www.mkyong.com" }, Locale.US);

System.out.println("Customer name (English) : " + name);

String namechinese = context.getMessage("customer.name",new Object[] {28, "http://www.mkyong.com" },Locale.SIMPLIFIED\_CHINESE);

System.out.println("Customer name (Chinese) : " + namechinese);

}

}

*Output*

