

## Resource Service

### Summary

We will examine the message supply services that are most frequently used, utilizing resources. The message supply service provides the function to read the message from the already defined file, and then get the message corresponding to key value to provide information message or when error occurs.

### Description

#### Message Basic

Explain the basic configuration and utilization for utilizing message, focusing on example.

#### Configuration

```
<bean name="messageSource"
class="org.springframework.context.support.ResourceBundleMessageSource">
<property name="useCodeAsDefaultMessage">
<value>true</value>
</property>
<property name="basenames">
<list>
<value>egovframework-message</value>
</list>
</property>
</bean>
```

In above setting, the file designated as "egovframework-message" is defined as egovframework-message.properties actually. Several methods are available for designating the location of file. See [4. Reference](#) for the setting.

#### Sample Source

Contents of message defined in //egovframework-message.properties.  
resource.basic.msg1=message1

```
@Resource(name="messageSource")
MessageSource messageSource ;
```

```
String getMsg = messageSource.getMessage("resource.basic.msg1" , null , Locale.getDefault() );
assertEquals("Get Message Success!", getMsg , "message1");
```

In above source, it can be checked that message can be obtained using messageSource.getMessage.

#### Message Locale

Can provide the message in the language suitable for user by setting and managing separately for each language with the same message key.

#### Configuration

```
<bean name="messageSource"
class="org.springframework.context.support.ResourceBundleMessageSource">
<property name="useCodeAsDefaultMessage">
<value>true</value>
</property>
<property name="basenames">
<list>
<value>egovframework-message-locale</value>
</list>
```

```
</property>
</bean>
```

In above setting, define the file designated as "egovframework-message-locale" with egovframework-message-locale\_ko.properties,egovframework-message-locale\_en.properties and designate the message corresponding to the same message key differently.

### Properties File

```
//egovframework-message-locale_ko.properties file contents
resource.locale.msg1=message 1
```

```
//egovframework-message-locale_en.properties file contents
resource.locale.msg1=en_message1
```

From above, it can be checked that other message was set in the key of resource.locale.msg1. Message can be provided depending on locale information by setting as shown above.

### Sample Source

Contents of message defined in //egovframework-message.properties.  
resource.basic.msg1=message1

```
String getMsg = messageSource.getMessage("resource.locale.msg1" , null , Locale.KOREAN );
assertEquals("Get Message Success!", getMsg , "message1");
```

```
String getMsg = messageSource.getMessage("resource.locale.msg1" , null , Locale.ENGLISH );
assertEquals("Get Message Success!", getMsg , "en_message1");
```

From above, it can be checked that the contents of message extracted differ depending on the locale information.

### Message Parameter

It is able to add and supply messages that occurs during execution of program. How to use this is same as the above and will be set in Properties file as shown below.

### Properties File

```
resource.basic.msg3=message {0} {1}
```

The properties file can receive with input of the additional message in the area defined as {0},{1} above. The utilizing sample of the above setting is as follow:

### Sample Source

```
Object[] parameter = { new String("1") , new Integer(2) };
```

```
String getMsg = messageSource.getMessage("resource.basic.msg3" , parameter , Locale.getDefault() );
assertEquals("Get Message Success!", getMsg , "message 1 2");
```

From above, designate 1 and 2 to parameter, call after inserting the second factor of getMessage and "message 1 2" can be obtained as return message.

### Reference

- [Spring Resource](#)