

Anyframe I18N Plugin



Version 1.1.0

저작권 © 2007-2014 삼성SDS

본 문서의 저작권은 삼성SDS에 있으며 Anyframe 오픈소스 커뮤니티 활동의 목적하에서 자유로운 이용이 가능합니다. 본 문서를 복제, 배포할 경우에는 저작권자를 명시하여 주시기 바라며 본 문서를 변경하실 경우에는 원문과 변경된 내용을 표시하여 주시기 바랍니다. 원문과 변경된 문서에 대한 상업적 용도의 활용은 허용되지 않습니다. 본 문서에 오류가 있다고 판단될 경우 이슈로 등록해 주시면 적절한 조치를 취하도록 하겠습니다.

I. Introduction	1
II. 국제화 (Internationalization)	2
1. MessageSource	3
1.1. ReloadableResourceBundleMessageSource	3
1.2. DatabaseMessageSource	4
1.3. AggregatingMessageSource	5
2. 사용자 Locale 변경	6
2.1. LocaleResolver	6
2.2. LocaleChangeInterceptor	6

I.Introduction

I18N Plugin은 사용자의 Locale에 따라 국제화를 지원하는 예제를 포함하고 있다. 국제화 지원을 위해 LocaleChangeResolver, LocaleChangeInterceptor 그리고 MessageSource를 사용한다.

Installation

Command 창에서 다음과 같이 명령어를 입력하여 I18N plugin을 설치한다.

```
mvn anyframe:install -Dname=i18n
```

installed(mvn anyframe:installed) 혹은 jetty:run(mvn clean jetty:run) command를 이용하여 설치 결과를 확인해볼 수 있다.

Plugin Name	Version Range
cache [http://dev.anyframejava.org/docs/anyframe/plugin/optional/cache/1.1.0/reference/htmlsingle/cache.html]	2.0.0 > * > 1.0.3

II.국제화 (Internationalization)

국제화란 대상 어플리케이션이 별도 코드의 수정없이 다양한 지역/언어를 수용할 수 있게 하는 기능이다. Spring MVC에서는 Presentation Layer에서 사용자의 Locale에 맞는 Message를 출력할 수 있도록 지원한다.

국제화에 대한 자세한 설명은 국제화(Internationalization) 매뉴얼 [http://dev.anyframejava.org/docs/anyframe/plugin/essential/core/1.6.0/reference/htmlsingle/core.html#core_springmvc_internationalization]을 참조한다.

I18N Plugin에서는 Locale변경을 위해 LocaleResolver, LocaleChangeInterceptor를 사용하며 Locale별 Message를 관리하기 위해 MessageSource를 사용한다.

- MessageSource : 다양한 Resource(파일 또는 DB)로 부터 Locale별 Message를 읽어온다.
- LocaleResolver : 사용자의 Locale을 읽고 저장한다. .
- LocaleChangeInterceptor : Presentation Layer에서 Locale을 변경해 준다. 내부에 LocaleResolver를 사용한다.

1.MessageSource

Spring의 Message 태그는 정의된 Message Bundle을 기반으로 code(message key)값을 이용해 Locale에 해당하는 Message Value를 사용자에게 보여준다.

Anyframe에서는 파일 기반의 ReloadableResourceBundleMessageSource와 DB 기반의 DatabaseMessageSource를 지원하며, 사용 목적에 따라 정의된 MessageSource가 다수일 경우, 이를 관리하기 위한 AggregatingMessageSource를 제공한다. MessageSource에 대한 설명은 MessageSource 매뉴얼 [http://dev.anyframejava.org/docs/anyframe/plugin/essential/core/1.6.0/reference/htmlsingle/core.html#core_messagesource_part]을 참조한다.

I18N Plugin에서는 파일과 DB로 부터 Locale별 Message를 읽어오며, AggregatingMessageSource Bean을 통해 MessageSource들을 관리한다.

1.1.ReloadableResourceBundleMessageSource

I18N Plugin은 영화 조회 및 관리를 제공하는 화면에서 파일 기반의 국제화 지원을 위해 ReloadableResourceBundleMessageSource를 사용한다.

ReloadableResourceBundleMessageSource를 사용하기 위해서 'Key=Value' 형식으로 Message를 파일에 정의한다. 다음은 I18N Plugin에서 사용하는 'message-moviefinder.properties' 파일의 일부이다.

```
movie.title=Title
movie.director=Director
movie.genre=Genre
movie.actors=Actors
```

다음은 'message-moviefinder_ko.properties' 파일의 일부이다. 사용자의 Locale이 한국일 경우, ReloadableResourceBundleMessageSource는 message-moviefinder_ko.properties로 부터 Message를 가져오게 된다.

```
movie.title=타이틀
movie.director=감독
movie.genre=장르
movie.actors=배우
```

위와 같이 정의된 Message Bundle을 기반으로 Locale에 맞게 Message를 가져오려면 다음과 같이 등록해주어야 한다.

```
<bean id="fileMessageSource"
class="org.springframework.context.support.ReloadableResourceBundleMessageSource">
  <property name="basenames">
    <list>
      <value>classpath:message/message-generation</value>
      <value>classpath:message/message-moviefinder</value>
      <value>classpath:message/message-converter</value>
    </list>
  </property>
  <property name="defaultEncoding">
    <value>UTF-8</value>
  </property>
```

```
</bean>
```

1.2.DatabaseMessageSource

I18N Plugin에서는 DatabaseMessageSource를 사용하는 예제를 제공하기 위해 Login과 관련된 Locale 별 Message를 DB에 저장한다.

DB에 Message를 등록하기 위해서 먼저 다음과 같이 MESSAGE_SOURCE 테이블을 생성해주어야 한다. 테이블 명과 컬럼 명은 변경이 가능하다.

Column Name	Data Type	Type Name	Column Size
ID	12	VARCHAR	50
LANGUAGE	12	VARCHAR	2
COUNTRY	12	VARCHAR	3
MESSAGE	12	VARCHAR	200

다음은 I18N Plugin을 통해 Message 관리 테이블에 입력한 Message이며, Message의 Locale을 language와 country 컬럼으로 구별하고 있다.

ID	LANGUAGE	COUNTRY	MESSAGE
i18n.error	en	US	Invalid id or password.
i18n.id	en	US	ID
i18n.password	en	US	Password
i18n.logout	en	US	Logout
i18n.login	en	US	Login
i18n.error	ko	KR	아이디 혹은 비밀번호가 맞지 않습니다.
i18n.id	ko	KR	아이디
i18n.password	ko	KR	비밀번호
i18n.logout	ko	KR	로그아웃
i18n.login	ko	KR	로그인

다음은 DatabaseMessageSource Bean의 속성 정의 내용이다. DatabaseMessageSource에 대한 보다 자세한 설명은 DatabaseMessageSource 매뉴얼 [http://dev.anyframejava.org/docs/anyframe/plugin/essential/core/1.6.0/reference/htmlsingle/core.html#core_db_messagesource]을 참조한다.

```
<bean id="databaseMessageSource"
class="org.anyframe.spring.message.DatabaseMessageSource">
  <property name="dataSource" ref="dataSource"/>
  <property name="messageTable">
    <props>
      <prop key="table">MESSAGE_SOURCE</prop>
      <prop key="key.column">ID</prop>
      <prop key="language.column">LANGUAGE</prop>
      <prop key="country.column">COUNTRY</prop>
      <prop key="text.column">MESSAGE</prop>
    </props>
  </property>
  <property name="cacheManager" ref="cacheManager"></property>
</bean>
```

1.3. AggregatingMessageSource

어플리케이션 개발시 국제화 지원을 위한 Message 정보를 각각 다른 유형의 Message Resource에 관리 할 수 있다. 이를 위해 Anyframe에서는 `org.anyframe.spring.message.AggregatingMessageSource`를 제공하고 있다. 다양한 타입의 Message Resource를 사용하여 어플리케이션의 국제화를 지원하고자 하는 경우 `AggregatingMessageSource`를 기본 'messageSource' bean으로 정의하고 `AggregatingMessageSource`를 통해 각 Message Resource에 접근하면 된다.

l18N Plugin에서는 `ReloadableResourceBundleMessageSource`와 `DatabaseMessageSource`를 사용하며, 이 MessageSource들은 `AggregatingMessageSource`의 Property로 정의하고 있다. `AggregatingMessageSource`는 정의된 MessageSource를 대상으로 사용자의 Locale에 대한 Message를 보여주게 된다.

```
<bean name="messageSource" class="org.anyframe.spring.message.AggregatingMessageSource">
  <property name="messageSources">
    <list>
      <ref bean="databaseMessageSource"/>
      <ref bean="fileMessageSource"/>
    </list>
  </property>
</bean>
```

`AggregatingMessageSource`는 입력된 Locale 정보(Language 정보, Country 정보)와 Message Key를 기반으로 먼저 'databaseMessageSource'를 대상으로 메시지를 찾는 작업을 수행한다. 적절한 메시지가 존재하지 않으면 'fileMessageSource'를 대상으로 적절한 메시지를 찾는 작업을 다시 시도 할 것이다.



MessageSource Bean 등록시 주의할 점

Spring Container는 Bean 이름이 'messageSource'인 MessageSource Bean을 내부적으로 참고하여 국제화를 수행한다. 여러개의 MessageSource를 사용하는 경우 `AggregatingMessageSource`의 Bean 이름을 'messageSource'로 정의하고 내부적으로 다른 MessageSource들에 접근하도록 할 수 있도록 정의해야 함에 유의 한다.

2.사용자 Locale 변경

I18N Plugin에서는 Locale 변경을 위해 LocaleResolver를 구현한 SessionLocaleResolver와 LocaleChangeInterceptor를 사용한다. LocaleResolver와 LocaleChangeInterceptor는 org.springframework.web.servlet.i18n 패키지에 정의되어 있으며, LocaleResolver와 관련된 보다 자세한 내용은 국제화(Internationalization) 매뉴얼 [http://dev.anyframejava.org/docs/anyframe/plugin/essential/core/1.6.0/reference/htmlsingle/core.html#core_springmvc_internationalization]을 참조한다.

2.1.LocaleResolver

사용자들이 사용하고자 하는 Locale을 직접 선택할 수 있도록 구현해야 한다면 CookieLocaleResolver 또는 SessionLocaleResolver를 이용할 수 있으며, I18N Plugin은 SessionLocaleResolver를 사용하고 있다.

다음은 SessionLocaleResolver Bean을 등록한 속성 정의 파일이다.




```
<bean id="localeResolver"
      class="org.springframework.web.servlet.i18n.SessionLocaleResolver">
  <property name="defaultLocale" value="en_US" />
</bean>
```

2.2.LocaleChangeInterceptor






LocaleResolver만 이용하여 Locale을 바꾸는 경우, Controller마다 정의한 LocaleResolver를 가져와서 Locale을 변경하는 코드가 필요하다. LocaleResolver를 이용한 Locale 변경 [http://dev.anyframejava.org/docs/anyframe/plugin/essential/core/1.6.0/reference/htmlsingle/core.html#core_springmvc_internationalization]을 참고한다.

I18N Plugin에서는 LocaleChangeInterceptor를 사용하여 Locale 변경 요청에 대해 자동으로 Locale을 변경하도록 한다. Controller에 매번 Locale 변경 코드를 넣는 것 보다 Interceptor를 이용하는 것이 더 간편하다. LocaleChangeInterceptor를 이용한 Locale 변경 [http://dev.anyframejava.org/docs/anyframe/plugin/essential/core/1.6.0/reference/htmlsingle/core.html#core_springmvc_internationalization_multilanguage_localechangeinterceptor]을 참고한다.

다음은 I18N Plugin을 통해 설치된 샘플 화면중의 하나이다.

Title: Now Playing:    Logout

Director	Actors	Ticket Price	Release Date
Tim Burton	Johnny Depp	8000.0	2011-03-04
James Cameron	Sigourney Weaver	7000.0	2011-02-16
Paul Greengrass	Yigal Naor	7000.0	2011-03-25

  1 2    Add

위 화면에서 보고 싶은 언어를 클릭하면 내부적으로 ChangeLocaleInterceptor가 호출되고, Request의 URL로 부터 Locale 값을 가져와서 LocaleResolver에 Locale값을 설정하게 될 것이다.

아래는 LocaleInterceptor Bean을 핸들러 매핑에 등록한 예이다. LocaleChangeInterceptor 내부에서 LocaleResolver를 사용하기 때문에 LocaleResolver가 먼저 속성 정의 파일에 정의되어야 한다.

```
<mvc:interceptors>
  <mvc:interceptor>
    <mvc:mapping path="/i18n*" />
    <bean id="localeChangeInterceptor"
      class="org.springframework.web.servlet.i18n.LocaleChangeInterceptor">
      <property name="paramName" value="locale" />
    </bean>
  </mvc:interceptor>
</mvc:interceptors>
```

앞 화면에서 태극기를 클릭한 경우 다음과 같이 다른 언어로 변경된 화면을 확인할 수 있다.

타이틀: 상영중:    로그아웃

감독	배우	티켓가격	개봉일
Tim Burton	Johnny Depp	8000.0	2011-03-04
James Cameron	Sigourney Weaver	7000.0	2011-02-16
Paul Greengrass	Yigal Naor	7000.0	2011-03-25



1
[2](#)



 추가