메시지 처리

MessageSource를 이용한 메시지 국제화 처리

❖ 메시지의 국제화 지원

- org.springframework.context.MessageSource 인터페이스 제공
  - 지역 및 언어에 따라 알맞은 메시지를 구하는 메서드 정의

```
public interface MessageSource {
   String getMessage(String code, Object[] args,
                     String defaultMessage, Locale locale);

   String getMessage(String code, Object[] args, Locale locale)
             throws NoSuchMessageException;

   String getMessage(MessageSourceResolvable resolvable, Locale locale)
             throws NoSuchMessageException;
}
```

MessageSource를 이용한 메시지 국제화 처리

❖ 메시지의 국제화 지원

- ApplicationContext
  - MessageSource 인터페이스를 상속
  - 등록된 빈 객체 중에서 이름이 'messageSource'인 MessageSource 타입의 빈 객체를 이용하여 메시지를 가져옴
    - 스프링 설정 파일에 이름이 'messageSource'인 빈 객체를 정의해 주어야 함

```
<bean id="messageSource"
    class="org.springframework.context.support.ResourceBundleMessageSource">
    <property name="basename">
        <value>message.greeting</value>
    </property>
</bean>
```

MessageSource를 이용한 메시지 국제화 처리

❖ 메시지의 국제화 지원

- ApplicationContext
  - ResourceBundleMessageSource 클래스
    - MessageSource 인터페이스의 구현 클래스
    - java.util.ResourceBundle을 이용하여 메시지를 읽어오는 구현체
    - basename 프로퍼티 값 : 메시지를 로드할 때 사용할 ResourceBundle의 베이스 이름
      - > 패키지를 포함한 완전한 이름
      - > message.greeting : message 패키지에 있는 greeting 프로퍼티 파일

MessageSource를 이용한 메시지 국제화 처리

❖ 메시지의 국제화 지원

- 한 개 이상의 프로퍼티 파일로부터 메시지 로딩
  - basenames 프로퍼티에 <list> 태그를 이용하여 목록 전달

```
<bean id="messageSource"
    class="org.springframework.context.support.ResourceBundleMessageSource">
    <property name="basenames">
        <list>
            <value>message.greeting</value>
            <value>message.error</value>
        </list>
    </property>
</bean>
```

MessageSource를 이용한 메시지 국제화 처리

❖ 메시지의 국제화 지원

- ResourceBundle의 메시지 로딩
  - 프로퍼티 파일의 이름을 이용하여 언어 및 지역에 따라 파일 선택
    - message.properties - 기본 메시지. 시스템의 언어 및 지역에 맞는 프로퍼티 파일이 존재하지 않을 경우 사용
    - message_en.properties - 영어 메시지
    - message_ko.properties - 한글 메시지
    - message_en_UK.properties - 영국을 위한 영어 메시지

```
#message_ko.properties
greeting=\uc548\ub155\ud558\uc138\uc694!
```

안녕하세요!의 유니코드

MessageSource를 이용한 메시지 국제화 처리

❖ 메시지의 국제화 지원

- ㅇ 메시지 얻기
  - ▪ ApplicationContext.getMessage() 메서드 이용

```
// 디폴트 Locale
Locale locale = Locale.getDefault();
String greeting = context.getMessage( " greeting " , new Object[0], locale);

// 영어
Locale englishLocale = Locale.ENGLISH;
String englishGreeting = context.getMessge("greeting", new Object[0],
                                           englishLocale);
```

MessageSource를 이용한 메시지 국제화 처리

❖ 빈 객체에서 메시지 이용하기

- 빈 객체에서 스프링이 제공하는 MessageBource를 사용하는 방법
  - ApplicationContextAware 인터페이스 구현
    - setApplicationContext() 메서드를 통해 전달받은 ApplicationContext의 getMessage() 메서드 이용
  - MessageSourceAware 인터페이스 구현
    - setMessageSource() 메시지를 통해 전달받은 MessageSource의 getMessage() 메서드를 이용

MessageSource를 이용한 메시지 국제화 처리

❖ 빈 객체에서 메시지 이용하기

- MessageSourceAware 인터페이스

```
public interface MessageSourceAware {
    void setMessageSource(MessageSource messageSource);
}
```

MessageSource를 이용한 메시지 국제화 처리

❖ 빈 객체에서 메시지 이용하기

- 빈 클래스에서 MessageSourceAware 인터페이스 구현

```
public class LoginProcessor implements MessageSourceAware {
    private MessageSource messageSource;

    public void setMessageSource(MessageSource messageSource) {
        this.messageSource = messageSource;
    }

    public void login(String username, String password) {
        ...
            String failMessage = messageSource.getMessage("login.fail", args,
                                                          Locale.getDefault());
            throw new IllegalArgumentException(failMessage);
        ...
    }
}
```

예제 : 메시지 처리

❖ 메시지 준비

- src/main/resource 에서 message 패키지 추가
  - error.properties

```
login.fail=Member ID {0} is not matching password
```

  - greeting_ko.properties

```
#message_ko.properties
greeting=\uc548\ub155\ud558\uc138\uc694!
```

안녕하세요!의 유니코드

  - greeting_en.properties

```
greeting=Hello!
```

예제 : 메시지 처리

❖ 클래스 추가

- 패키지 : com.lecture.spring.message
- 클래스 : LoginProcessor

```
import org.springframework.context.MessageSource;
import org.springframework.context.MessageSourceAware;

public class LoginProcessor implements MessageSourceAware {
    private MessageSource messageSource;

    @Override
    public void setMessageSource(MessageSource messageSource) {
        this.messageSource = messageSource;
    }

    public void login(String username, String password) {
        if (!username.equals("madvirus")) {
            Object[] args = new String[] { username };
            String failMessage = messageSource.getMessage("login.fail", args,
                    Locale.getDefault());
            throw new IllegalArgumentException(failMessage);
        }
    }
}
```

예제 : 메시지 처리

❖ 스프링 설정 파일

- /src/main/resource에 applicationMessageContext.xml 추가

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

   <bean id="messageSource"
      class="org.springframework.context.support.ResourceBundleMessageSource">
      <property name="basenames">
         <list>
            <value>message.greeting</value>
            <value>message.error</value>
         </list>
      </property>
   </bean>

   <bean id="loginProcessor"
      class="com.lecture.spring.message.LoginProcessor" />
</beans>
```

예제 : 메시지 처리

❖ 테스트

- MessageTest.java

```
import java.util.Locale;

import org.springframework.context.support.AbstractApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.lecture.spring.message.LoginProcessor;

public class MessageTest {

   public static void main(String[] args) {
      String[] configLocations = new String[] {
                                    "applicationMessageContext.xml" };
      AbstractApplicationContext context = new ClassPathXmlApplicationContext(
                                       configLocations);

      Locale locale = Locale.getDefault();
      String greeting = context.getMessage("greeting", new Object[0], locale);
      System.out.println("Default Locale Greeting: " + greeting);
```

예제 : 메시지 처리

❖ 테스트

- MessageTest.java

```
        Locale englishLocale = Locale.ENGLISH;
        String englishGreeting = context.getMessage("greeting", new Object[0],
                englishLocale);
        System.out.println("English Locale Greeting: " + englishGreeting);


        LoginProcessor loginProcessor = context.getBean("loginProcessor",
                LoginProcessor.class);
        try {
            loginProcessor.login("hong", "1234");
        } catch (Throwable e) {
            System.out.println("예외 발생: " + e.getMessage());
        }
    }
}
```