Spring Framework

김재진(nije1983@gmail.com)
http://springstudyclub.tistory.com

Resources

일시 : 2008년 9월 21일

성명 : 김재진

Spring Framework

김재진(nije1983@gmail.com)
http://springstudyclub.tistory.com

1.Introduction

- java.net.URL 인터페이스의 URL 접두사를 위한 표준 핸들러에 접근성 강화 >> 하위자원에 접근가능
  - Ex) classpath 또는 Servlet Context 로 부터 상대적인 위치를 얻을 수 있다

- 자원의 존재 여부를 체크하는 등의 필요 함수가 추가

김재진(nije1983@gmail.com)
http://springstudyclub.tistory.com

2. The Resource Interface

Method Summary	
InputStream	**getInputStream**() Return an InputStream.

Method Summary	
Resource	**createRelative**(String relativePath) Create a resource relative to this resource.
boolean	**exists**() Return whether this resource actually exists in physical form.
String	**getDescription**() Return a description for this resource, to be used for error output when working with the resource.
File	**getFile**() Return a File handle for this resource.
String	**getFilename**() Return a filename for this resource, i.e. typically the last part of the path: for example, "myfile.txt".
URI	**getURI**() Return a URI handle for this resource.
URL	**getURL**() Return a URL handle for this resource.
boolean	**isOpen**() Return whether this resource represents a handle with an open stream.
boolean	**isReadable**() Return whether the contents of this resource can be read, e.g. via InputStreamSource.getInputStream() or getFile().
long	**lastModified**() Determine the last-modified timestamp for this resource.

김재진(nije1983@gmail.com)
http://springstudyclub.tistory.com

2. The Resource Interface

- getInputStream()
  - 자원의 위치를 정하고 연다
  - 자원을 읽기 위한 InputStream을 반환
  - 각각의 호출은 refresh된 InputStream을 반환
  - 스트림을 닫기 위한 호출자의 책임을 갖는다
- exists()
  - resource 가 물리적으로 존재 하는지 여부를 boolean값으로 반환

김재진(nije1983@gmail.com)
http://springstudyclub.tistory.com

2. The Resource Interface

- isOpen()
  - resource가 열린 스트림을 다루는지 여부를 boolean 값으로 반환
  - InputStreamResource 를 제외한 모든 resource는 항상 false를 반환

- getDescription()
  - resource의 상세 설명을 반환
  - resource를 작업할 때 에러처리에 출력 문으로 사용가능
  - 파일의 전체 경로나 실직적인 URL이 출력

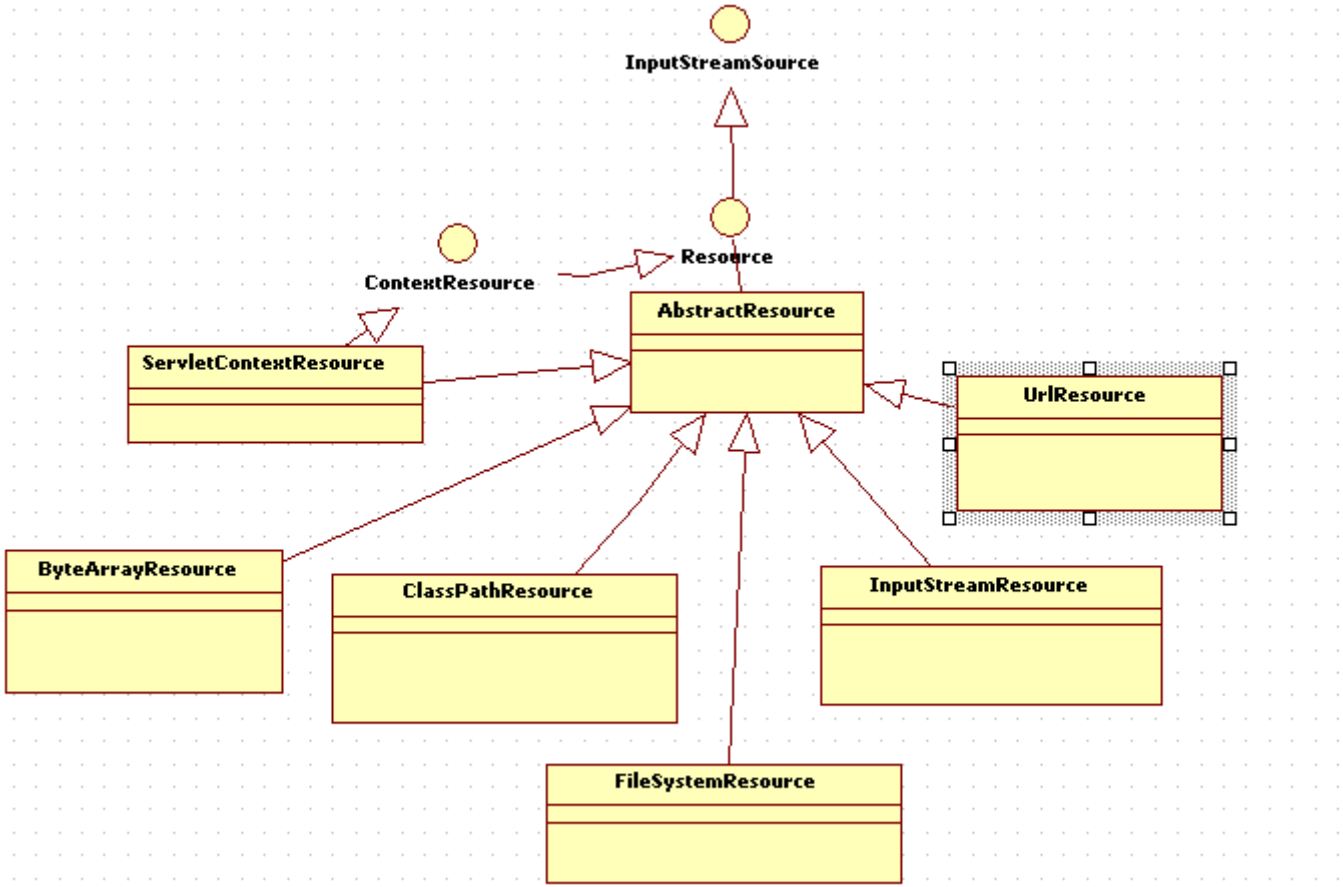
김재진(nije1983@gmail.com)
http://springstudyclub.tistory.com

3.Built-in Resource implementations

- UrlResource
- ClassPathResource
- FileSystemResource
- ServletContextResource
- InputStreamResource
- ByteArrayResource

김재진(nije1983@gmail.com)
http://springstudyclub.tistory.com

3.Built-in Resource implementations



김재진(nije1983@gmail.com)
http://springstudyclub.tistory.com

3.1 UrlResource

- java.net.URL 을 포장
- FILE, HTTP, FTP 타겟 과 같은 URL을 통해 객체에 접근하기 위해 사용
- 적절히 표준화된 접두사를 String 표현 (file:, http:, ftp:)

김재진(nije1983@gmail.com)
http://springstudyclub.tistory.com

3.2 ClassPathResource

- ClassPath로 부터 획득할 수 있는 Resource를 표현

- Resource에 대해 java.io.File로 해결
  - resource 가 jar내부에 있는 경우 제외
  - 서블릿 엔진 또는 환경에 의해 확장된 파일 시스템 제외

- 각각의 Resource의 주소는 항상 java.net.URL로 해결(file: 접두사)

김재진(nije1983@gmail.com)
http://springstudyclub.tistory.com

3.3 FileSystemResource

- java.io.File 을 다루기 위한 Class
- *java.io.File, java.net.URL로의 해결을 지원*
- '\\' = '/'

김재진(nije1983@gmail.com)
http://springstudyclub.tistory.com

3.4 ServletContextResource

- 웹 어플리케이션 root Directory 내의 상대적인 경로를 해석하는 Resource Class

- Stream 접근과 URL 접근을 항상 지원
  - 웹 어플리케이션의 archive가 확장되고 Resource가 파일시스템에 물리적으로 존재할 때만 java.io.File 의 접근을 허용(file:)
  - Jar 또는 DB에 직접 접근하는지는 서블릿 컨테이너에 의존(jar: 등)

김재진(nije1983@gmail.com)
http://springstudyclub.tistory.com

3.5 InputStreamResource

- InputStream을 위한 Class
- 특정한 Resource의 구현이 아닐 경우에만 사용
- 가능하다면 ByteArrayResource 또는 파일기반의 Resource Class를 사용

김재진(nije1983@gmail.com)
http://springstudyclub.tistory.com

3.6 ByteArrayResource

- byte[] 을 위한 Resource Class
- 한번 사용되는 InputStream을 재정렬하지 않고 byte[]로 부터 내용을 로드한다

김재진(nije1983@gmail.com)
http://springstudyclub.tistory.com

4. The ResourceLoader

- Resource instance를 반환해주는 Interface

Method Summary	
ClassLoader	**getClassLoader**() Expose the ClassLoader used by this ResourceLoader.
Resource	**getResource**(String location) Return a Resource handle for the specified resource.

- 모든 applicationContext 는 ResourceLoader 를 구현한다

- 따라서 모든 applicationContext는 Resource를 얻는데 사용될 수 있다

김재진(nije1983@gmail.com)
http://springstudyclub.tistory.com

4. The ResourceLoader

- Resource 획득
  - getResource(String Location) 호출
  - 특정 접두사를 가지지 않을 경우 applicationContext에 적합한 Resource를 획득
    - Ex) FileSystemXmlApplicationContext >> FileSystemResource

김재진(nije1983@gmail.com)
http://springstudyclub.tistory.com

4. The ResourceLoader

접두사를 사용 한 Resource 획득

- getResource(String Location) 호출 시 다음과 같은 접두사를 사용해 applicationContext의 타입에 상관없이 강제로 해당 Resource를 획득

접두사	예제	상세설명
classpath:	Classpath:com/myapp/config.xml	classpath로부터 로드
file:	File:/data/config.xml	파일시스템으로 부터 URL처럼 로드
http:	http://myserver/logo.png	URL처럼 로드
(none)	/data/config.xml	기초적인 applicationContext에 의해 결정

김재진(nije1983@gmail.com)
http://springstudyclub.tistory.com

5. ResourceLoaderAware 인터페이스

- *ResourceLoader 참조가 제공될 것이라고 기대되는 객체의 확인을 위한 특별한 표시자 인터페이스*
  - Ex)ApplicatationContext

- ResourceLoaderAware를 구현하여 bean으로 등록
  - Applicataion Context는 자신을 인자로 제공하는 setResourceLoader()을 호출

김재진(nije1983@gmail.com)
http://springstudyclub.tistory.com

6. Resource as dependencies

- 동적 처리에 의한 Resource Class
  - ResourceLoader 인터페이스 구현
  - XML 설정을 이용

```
bean id="myBean" class="...">
  <property name="template" value="some/resource/path/myTemplate.txt"/>
</bean>
```

  - 접두사를 이용해 특정 Resouce 타입의 사용

```
<property name="template" value="classpath:some/resource/path/myTemplate.txt">
```

```
<property name="template" value="file:/some/resource/path/myTemplate.txt"/>
```

김재진(nije1983@gmail.com)
http://springstudyclub.tistory.com

7. 1 Construction Application contexts

- Applicatation Context 는 일반적으로 context를 정의하는 XML파일의 경로로 String이나 String[] 을 가진다

- 경로문자열이 특정한 접두사를 가지지 않을 때 Applicateion Context에 맞게 적절한 Resource 가 해당경로로 부터 만들어지고, 빈이 정의되며 관계를 맺는다

Spring Framework

김재진(nije1983@gmail.com)
http://springstudyclub.tistory.com

7. 1 Construction Application contexts

- 경로 문자열에 특정 접두사(classpath, URL)를 사용하는 것은 실질적인 Resource의 정의를 해당 경로로 부터 로드한다

```
ApplicationContext ctx =
    new FileSystemXmlApplicationContext("classpath:conf/appContext.xml");
```

- 특정 접두사를 이용해 Resource를 로드 하더라도 context의 타입이 접두사에 영향을 받지 않는다.

김재진(nije1983@gmail.com)
http://springstudyclub.tistory.com

7. 1.1 ClassPathXmlApplicationContext 인트턴스 생성하기

ClassPathXmlApplicationContext의 생성자

Constructor Summary
ClassPathXmlApplicationContext() Create a new ClassPathXmlApplicationContext for bean-style configuration.
ClassPathXmlApplicationContext(ApplicationContext parent) Create a new ClassPathXmlApplicationContext for bean-style configuration.
ClassPathXmlApplicationContext(String configLocation) Create a new ClassPathXmlApplicationContext, loading the definitions from the given XML file and automatically refreshing the context.
ClassPathXmlApplicationContext(String[] configLocations) Create a new ClassPathXmlApplicationContext, loading the definitions from the given XML files and automatically refreshing the context.
ClassPathXmlApplicationContext(String[] configLocations, ApplicationContext parent) Create a new ClassPathXmlApplicationContext with the given parent, loading the definitions from the given XML files and automatically refreshing the context.
ClassPathXmlApplicationContext(String[] configLocations, boolean refresh) Create a new ClassPathXmlApplicationContext, loading the definitions from the given XML files.
ClassPathXmlApplicationContext(String[] configLocations, boolean refresh, ApplicationContext parent) Create a new ClassPathXmlApplicationContext with the given parent, loading the definitions from the given XML files.
ClassPathXmlApplicationContext(String[] paths, Class clazz) Create a new ClassPathXmlApplicationContext, loading the definitions from the given XML files and automatically refreshing the context.
ClassPathXmlApplicationContext(String[] paths, Class clazz, ApplicationContext parent) Create a new ClassPathXmlApplicationContext with the given parent, loading the definitions from the given XML files and automatically refreshing the context.
ClassPathXmlApplicationContext(String path, Class clazz) Create a new ClassPathXmlApplicationContext, loading the definitions from the given XML file and automatically refreshing the context.

김재진(nije1983@gmail.com)
http://springstudyclub.tistory.com

7.2 Wildcards in Resource Paths

- Resource 의 경로는 Resource와 1:1로 맵핑되는 간단한 경로
- 대안으로 classpath* 접두사나 내부 Ant스타일의 정규표현식 포함
- 컴포넌트 스타일의 애플리케이션을 조합할 때 사용

김재진(nije1983@gmail.com)
http://springstudyclub.tistory.com

7.2.1 Ant Style Patterns

```
/WEB-INF/*-context.xml
com/mycompany/**/applicationContext.xml
file:C:/some/path/*-context.xml
classpath:com/mycompany/**/applicationContext.xml
```

김재진(nije1983@gmail.com)
http://springstudyclub.tistory.com

7.2.2 classpath* 접두사

- 모든 classpath내의 자원을 통합하여 Applicatationcontext 를 정의

- classpath* 접두사의 경로내 마지막 경로에 *를 이용해 PathMatcher 패턴으로 조합가능

김재진(nije1983@gmail.com)
http://springstudyclub.tistory.com

7.3 FileSystemResource 경고

- 이전버전과의 호환의 이유로 FileSystemApplicatationContext 가 ResoueceLoader 일때 문자열 경로의 첫 문자가 '/' 일때 절대경로와 상대경로를 구분하지 않는다

- 절대경로를 이용하기 위해서 file: 접두사를 이용하여 ResourceLoader가 UrlResource를 사용하게 해야한다