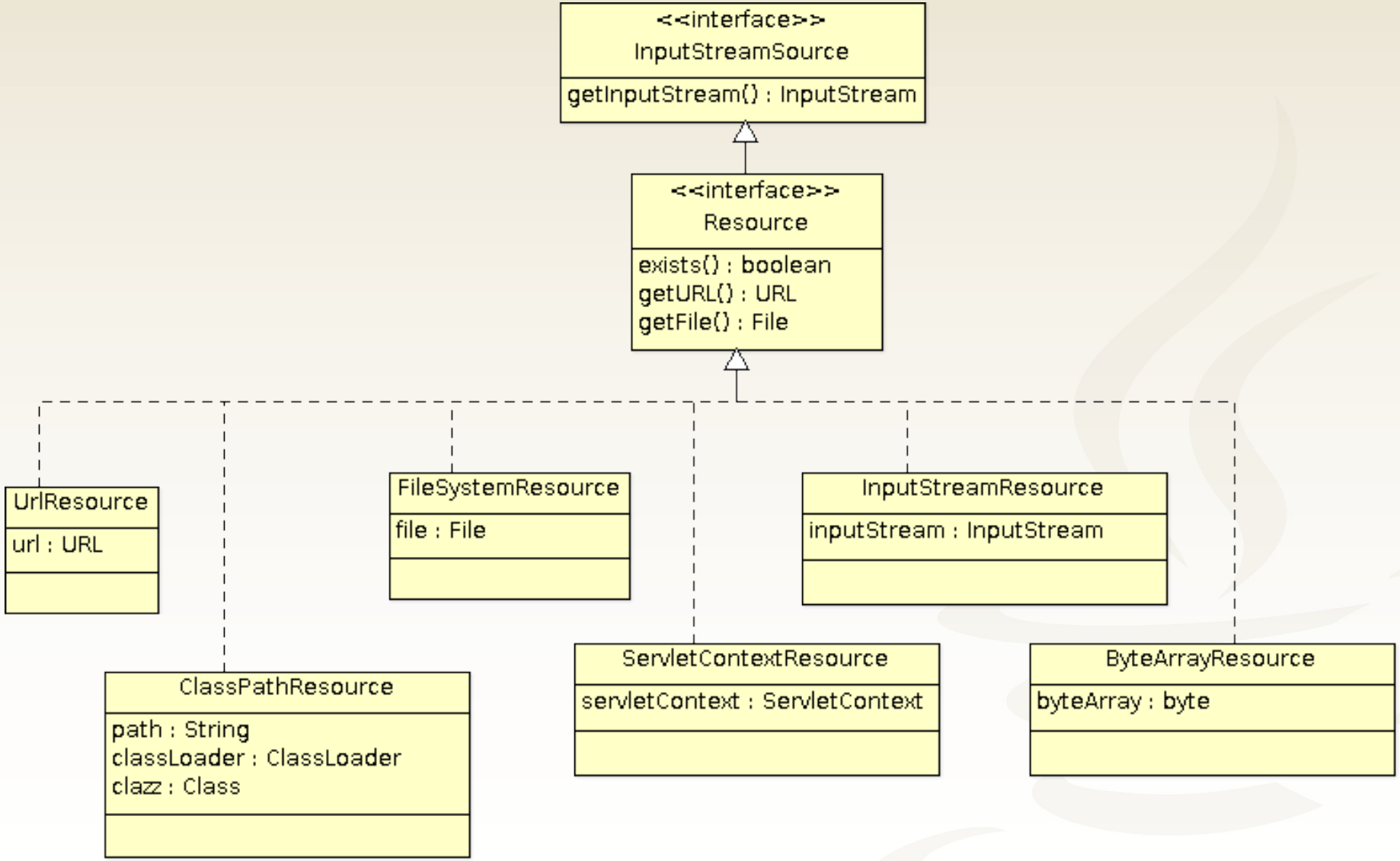
Resource Yükleme Kabiliyeti

Spring Resource API

- **Classpath, Filesystem veya ServletContext**'e relatif resource'lara erişimi kolaylaştırır
- Spring'in kendi içinde de sıkça kullan bir yöntemdir
- Temeli yine **java.net.URL, java.io.InputStream** gibi sınıflara dayanır
- Resource'ların nereden ve nasıl yükleneceği **başındaki prefix**'e bakılarak anlaşılır

Spring Resource Tipleri



Resource Path Örnekleri

- **classpath**:some/resource/path/image.png
  - ClasspathResource
  - Uygulamanın classpath'inden bulunup yüklenir
- **file**: /some/resource/path/contents.txt
  - FileSystemResource
  - Dosya sisteminden yüklenir

Resource Path Örnekleri

- **http**://localhost/styles.css
  - URLResource
  - URL olarak network'den yüklenir
- /some/resource/path/invoice.pdf
  - Başında **prefix yoksa** durum biraz farklıdır
  - Resource tipi ve yüklenme şekli **ApplicationContext tipine göre** değişir

ResourceLoader ve ResourceLoaderAware

- **ResourceLoader**, Resource nesnelerini yükler
- **ApplicationContext aynı zamanda bir ResourceLoader** nesnesidir
- **ResourceLoaderAware** arayüzüne sahip bean'lara ResourceLoader otomatik olarak enjekte edilir

ResourceLoader ve ResourceLoaderAware

JAVA Eğitimleri

```
public class UserService implements ResourceLoaderAware {

    private ResourceLoader resourceLoader;

    @Override
    public void setResourceLoader(ResourceLoader
resourceLoader) {
        this.resourceLoader = resourceLoader;
    }

    public InputStream getImage() throws IOException {
        Resource resource =
            resourceLoader.getResource("/some/path/image.png");
        return resource.exists()?
                        resource.getInputStream():null;
    }
}
```

Resource Bağımlılıkları

```
public class Foo {
    private Resource image;

    public void setImage(Resource image) {
        this.image = image;
    }
}
```

Buradaki değişken Resource tipindedir

```
<bean id="foo" class="x.y.Foo">
  <property name="image"
value="some/resource/path/image.png"/>
</bean>

<property name="image"
value="classpath:some/resource/path/image.png">

<property name="image"
value="file:/some/resource/path/image.png"/>
```

Konfigürasyon Metadata ve Resource Path Kullanımı

- Resource path ile **ApplicationContext dosyalarının yeri** de belirtilebilir
- Resource path'inde **wildcard** kullanılabilir
- file:/some/path/beans-*.xml
- classpath:/some/path/**/applicationContext.xml

Konfigürasyon Metadata ve Resource Path Örnekleri

```
ApplicationContext ctx = new
ClassPathXmlApplicationContext("beans/appContext.xml");
```

```
ApplicationContext ctx = new
FileSystemXmlApplicationContext("beans/appContext.xml");
```

```
ApplicationContext ctx = new
FileSystemXmlApplicationContext("classpath:beans/appCont
ext.xml");
```

Konfigürasyon metadata'nın mutlaka classpath'den
Yüklenmesi istenirse classpath: öneki kullanılabilir

classpath*: Ön Eki Ne İşe Yarar?

- **classpath:**/appcontext/beans-*.xml şeklinde bir kullanımda Spring classpath'de **ilk bulduğu /appcontext/ dizini**nde duracaktır
- Burası da web uygulamalarında muhtemelen **WEB-INF/classes** dizini olacaktır
- Spring'in taramaya devam etmesi ve **WEB-INF/lib dizini altındaki JAR dosyaları**nın içerisini de taraması istenebilir

classpath*: Ön Eki Ne İşe Yarar?

JAVA Eğitimleri

- Böyle bir durumda **classpath*: ön eki** devreye girmektedir
- Resource'ları bulmak için eşleşen **bütün classpath lokasyonları**na bakılır
  - classpath*:/appcontext/beans-*.xml

classpath*: Ön Eki Ne İşe Yarar? (İkaz)

- JAR içindeki resource'lara **pattern ile erişilebilme**si için **root path altında en azından bir dizin**de olmaları gerekir
  - classpath*:/beans-*.xml, jar içerisindeki resourceları bulamaz
  - Sadece expanded dizindeki resourceları bulabilir

İletişim

- Harezmi Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- http://www.java-egitimleri.com
- info@java-egitimleri.com