

# Spring ApplicationContext Environment



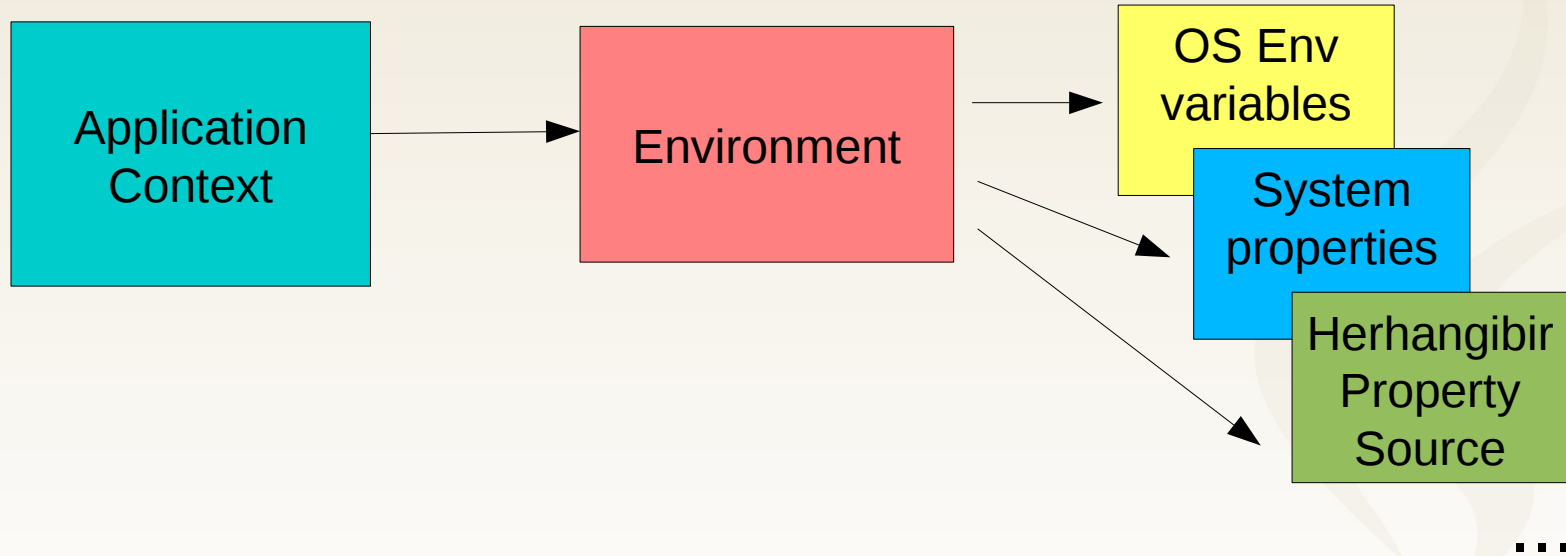
# Spring 3 ve Environment Kabiliyeti

- Spring 3 ile birlikte ApplicationContext'e **Environment** isimli bir soyutlama eklendi
- Environment **uygulamanın çalıştığı ortamı** ifade etmektedir
- Uygulamanın ortamsal iki temel özelliğini yönetir
  - Profiller
  - Property değerleri

# Spring ve Environment Kabiliyeti

- Spring 3 ile birlikte Environment üzerinden **property değerlerinin çözümlemesi** çok daha esnek bir hale geldi
- Artık sadece işletim sisteminin ortam değişkenleri, JVM sistem değişkenleri ve properties dosyalarındaki tanımlar dışında **farklı kaynaklardan da property değerleri** yüklenebilir
- Bu property kaynaklarının **öncelik sıralaması** değiştirilebilir

# Spring ve Environment Kabiliyeti



```
ApplicationContext context = new  
    ClassPathXmlApplicationContext("/appcontext/beans.xml");
```

```
Environment env = context.getEnvironment();  
String foo = env.getProperty("foo");
```

# PropertySource

## Konfigürasyonu: Web

ApplicationContextInitializer arayüzü implement edilerek custom PropertySource nesnesi environment'a eklenebilir

```
public class MyInitializer implements
ApplicationContextInitializer<ConfigurableWebApplicationContext> {

    public void initialize(ConfigurableWebApplicationContext ctx) {
        PropertySource ps = new MyPropertySource();
        ctx.getEnvironment().getPropertySources().addFirst(ps);
    }
}
```

ApplicationContextInitializer web uygulamalarında web.xml'de context param vasıtası ile devreye sokulabilir

```
<context-param>
    <param-name>contextInitializerClasses</param-name>
    <param-value>x.y.z.MyInitializer</param-value>
</context-param>
```

# PropertySource

## Konfigürasyonu: Standalone

Standalone ortamda ise custom PropertySource **GenericXmlApplicationContext** üzerinden eklenebilir

```
GenericXmlApplicationContext context =
    new GenericXmlApplicationContext();

context.load(new ClassPathResource("/appcontext/beans.xml"));

context.getEnvironment()
    .getPropertySources().addFirst(new MyPropertySource());

context.refresh();
```

Refresh metodu çağrılana kadar ApplicationContext konfigürasyonu yapılabilir. Refresh metodu ile bean'ler yaratılacaktır.

# İletişim

- **Harezmi** Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- <http://www.java-egitimleri.com>
- [info@java-egitimleri.com](mailto:info@java-egitimleri.com)

