

Tip Güvenli Konfigürasyon Property Yönetimi



Tip Güvenli Konfigürasyon Yönetimi

application.properties

```
petclinic.period=1000  
petclinic.adminRoles=VET,ADMIN  
  
petclinic.fileTransfer.enabled=true  
petclinic.fileTransfer.downloadPath=/downloads  
petclinic.fileTransfer.uploadPath=/uploads
```

@Component

```
public class FileTransferManager {  
  
    @Value("${petclinic.fileTransfer.enabled}")  
    private boolean fileTransferEnabled;  
  
    @Value("${petclinic.fileTransfer.downloadPath}")  
    private String downloadPath;  
  
    @Value("${petclinic.fileTransfer.uploadPath}")  
    private String uploadPath;  
    ...  
}
```

Tip Güvenli Konfigürasyon Yönetimi

```
@ConfigurationProperties(prefix="petclinic")
public class PetClinicProperties {

    private long period;

    private List<String> adminRoles = new ArrayList<>();

    private FileTransfer fileTransfer = new FileTransfer();

    //getter & setters...

    public static class FileTransfer {

        private boolean enabled;

        private String downloadPath;

        private String uploadPath;

        // getter & setters...

    }
}
```

Tip Güvenli Konfigürasyon Yönetimi

```
@Component
public class FileTransferManager {

    @Autowired
    private PetClinicProperties petClinicProperties;

    public void checkUploadsFolder() {
        if(petClinicProperties.getFileTransfer().isEnabled()) {
            //...
        }
    }
}
```

ConfigurationProperties Sınıflarının Tanıtılması

- Konfigürasyon sınıfları **@EnableConfigurationProperties** ile tanıtılmalıdır

```
@EnableConfigurationProperties(value={PetClinicProperties.class})
```

```
@SpringBootApplication
```

```
public class PetClinicApplication {
```

```
...
```

```
}
```

Eğer @ConfigurationProperties anotasyonuna sahip sınıf @Component olarak tanımlanmış ise burada tekrar belirtilmesine gerek yoktur

Bean Tanımında @Configuration Properties Kullanımı

- @ConfigurationProperties anotasyonunu **bean factory metotlarında** da kullanabiliriz
- Bu durumda bean yaratılırken **prefix ile eşleşen property**'ler ilgili bean instance'ına enjekte edilecektir

```
@Configuration
public class CustomDataSourceConfiguration {
    @ConfigurationProperties(prefix="app.datasource")
    @Bean
    public DataSource dataSource() {
        return new DriverManagerDataSource();
    }
}
```

app.datasource.url=jdbc:mysql://localhost/test
app.datasource.username=dbuser
app.datasource.password=dbpass

İletişim

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

