YAML ile Konfigürasyon Property Tanımlarının Yönetimi

harezmi bilişim çözümleri

Konfigürasyon Ayarlarının YAML ile Yönetilmesi

JAVA Eğitimleri

- YAML **JSON**'ın superset'idir
- **Hiyerarşik konfigürasyon verisi**nin tanımlanması için oldukça uygun bir formattır
- Spring Boot, **YAML formatı**nı da destekler

YAML Örneği

application.yml

```yaml
server:
    port: 8081
    context-path: /petclinic
    tomcat:
        basedir: /tmp
        max-connections: 1000

logging:
    file: petclinic.log
    config: classpath:/custom-log4j.xml
```

application.properties 

```properties
server.port=8081
server.context-path: /petclinic
server.tomcat.basedir: /tmp
server.tomcat.max-connections: 1000

logging.file: petclinic.log
logging.config: classpath:/custom-log4j.xml
```

Spring Profile Kabiliyeti ve YAML

- Tek bir YAML dosyasında **birden fazla Spring profili** için property tanımları yapılabilir

```
server:
    address: 192.168.1.1
---
spring:
    profiles:dev
server:
    address: 127.0.0.1
---
spring:
    profiles:prod
server:
    address: 107.180.27.155
```

YAML ve Default Profil Kullanımı

- Tanımlı profillerden hiç birisi aktif değilse, varsa **default profil tanımı** devreye alınır

```
spring:
    profiles:default
server:
    port: 8081
---
spring:
    profiles:dev
server:
    port: 8082
---
spring:
    profiles:prod
server:
    port: 80
```

Spring Profile Kabiliyeti ve YAML

- Güncel Spring Boot sürümünde Spring **profile tanımları**na karşılık gelen YAML dosyaları da application.properties'e benzer biçimde **ayrı ayrı tanımlanabilmekte**dir
  - application.yaml
  - application-dev.yaml
  - application-test.yaml
  - application-prod.yaml

İletişim

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