

Data Population Kabiliyeti



Sample Data Population

- Spring Boot classpath'de **schema.sql** ve **data.sql** script dosyaları mevcut ise bu dosyaların içindeki sql ifadelerini bootstrap sırasında çalıştıracaktır
- Dosyaların lokasyonu veya isimleri uygulamaya göre özelleştirilebilir

```
spring.datasource.schema=classpath:/schema.sql  
spring.datasource.data=classpath:/data.sql
```

Sample Data Population

- Eğer classpath'de mevcut ise **schema- $\{platform\}$.sql** ve **data- $\{platform\}$.sql** dosyalarındaki ifadeler de çalıştırılacaktır
- **$\{platform\}$** değişkeninin değeri **application.properties** içerisindeki tanımdan çözülür

```
spring.datasource.platform=all
```

hsqldb,h2,oracle,mysql,postgresql
gibi değerler yazılabilir
Bizim belirlediğimiz bir değerde olabilir

Sample Data Population

- Hibernate'de bootstrap sırasında eğer ddl-auto değeri **create** veya **create-drop** durumunda classpath'de **import.sql** isimli bir dosya mevcut ise bu dosyadaki sql ifadelerini çalıştırır

Veritabanı Init Sürecindeki İşlemlerin Sıralaması

- schema-`${platform}`.sql
- schema.sql
- data-`${platform}`.sql
- data.sql
- ddl-auto değerine göre schema değişiklikleri
- ddl-auto değeri create|create-drop ise import.sql

Data Initialization'ın Devre Dışı Bırakılması

- İstenirse veritabanı initialization işlemi **devre dışı** bırakılabilir

```
spring.datasource.initialization-mode=never
```

Alabileceği değerler:

- always
- embedded
- never

DB Migration Desteği

- **schema.sql** ve **data.sql** ile veritabanında incremental değişiklikler yaparak ilerlemek mümkün değildir
- Spring Boot gelişmiş DB migration ihtiyaçları için **flyway** ve **liquibase** araçlarını da desteklemektedir
- Flyway doğrudan **SQL ifadeleri** üzerinden çalışır
- Liquibase ise kendine ait **vendor bağımsız bir DSL**'e sahiptir

DB Migration Desteği

- Devreye almak için ilgili kütüphaneleri **pom.xml**'e eklemek yeterlidir

```
<dependency>  
  <groupId>org.flywaydb</groupId>  
  <artifactId>flyway-core</artifactId>  
</dependency>
```

```
<dependency>  
  <groupId>org.liquibase</groupId>  
  <artifactId>liquibase-core</artifactId>  
</dependency>
```


Flyway

- Flyway SQL migration dosyaları classpath'de **db/migration** dizini altına yer almalıdır
- Dosyaların formatı
V<version>__<desc>.sql şeklinde olmalıdır
 - V1__schema.sql
 - V2__data.sql
 - ...
- Migration işlemleri ile ilgili takip DB'de **flyway_schema_history** isimli tablo içerisinde gerçekleştirilir

Liquibase

classpath:/db/changelog/db.changelog-master.yaml



```
databaseChangeLog:
  - changeSet:
    id: 1
    changes:
      - createTable:
        tableName: person
        columns:
          - column:
            name: id
            type: int
            constraints:
              primaryKey: true
              nullable: false
          - column:
            name: first_name
            type: varchar(255)
  - changeSet:
    id: 2
    changes:
      - addColumn:
        tableName: person
        columns:
          - column:
            name: last_name
            type: varchar(255)
```

İletişim

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

