Haaga-Helia

Server Programming

JDBC, Databases

Juha Hinkula

Haaga-Helia

Spring Boot: JDBC

- Spring provides template class called *JdbcTemplate*
- *JdbcTemplate* takes care of connection handling and exception handling
  - `Import org.springframework.jdbc.core.JdbcTemplate`
- Dependency (Spring Boot automatically creates JdbcTemplate)

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-jdbc</artifactId>
</dependency>
```

Haaga-Helia

Spring Boot: JDBC

- Datasource configuration is controlled by using *application.properties* file
- Example:

```
spring.datasource.url=jdbc:mysql://localhost/test
spring.datasource.username=dbuser
spring.datasource.password=dbpass
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
spring.datasource.initialization-mode=always
```

Haaga-Helia

Spring Boot: JDBC

- JdbcTemplate can be injected to your classes

```
@Autowired
private JdbcTemplate jdbcTemplate;
```

- Excecute single SQL statement (typically DDL) by using excecute method

```
jdbcTemplate.execute("CREATE TABLE customers...");
```

- You can also use *schema.sql* and *data.sql* files in resources folder which Spring Boot will automatically use to initialize database

Haaga-Helia

Spring Boot: JDBC

- Create a repository class and add methods
- Example

```
@Repository
public class StudentRepository
{
    @Autowired
    private JdbcTemplate jdbcTemplate;

    @Transactional(readOnly=true)
    public List<Student> findAll() {
            return jdbcTemplate.query("select * from student", new StudentRowMapper());
    }
}
```

Haaga-Helia

Spring Boot: JDBC

- Spring framework has comprehensive transaction support
- If method is tagged with *@Transactional*, meaning that any failure causes the entire operation to roll back to its previous state, and to throw the original exception

Spring Boot: JDBC

- RowMapper is interface for mapping rows of a ResultSet
- Examle

```
class StudentRowMapper implements RowMapper<Student> {
  @Override
  public Student mapRow(ResultSet rs, int rowNum) throws SQLException {
            Student student = new Student();
            student.setId(rs.getInt("id"));
            student.setName(rs.getString("name"));
            student.setEmail(rs.getString("email"));
            return student;
  }
}
```

Haaga-Helia

Spring Boot: JDBC

- It is recommended to use ? for arguments to avoid SQL injections,
- Example

```
jdbcTemplate.update("INSERT INTO customers(first_name, last_name) VALUES (?,?)",
"John", "West");
```