

Accessing Databases with Spring Data



Jesper de Jong

Software Architect

@jesperdj | www.linkedin.com/in/jespdj



Overview



Spring Data

MongoDB

- NoSQL database
- Spring Data MongoDB

PostgreSQL

- Relational database
- Spring Data JPA

Spring Boot Application Configuration

- Profiles





Introducing Spring Data



```
// Querying with JDBC
```

```
var cn = DriverManager.getConnection(  
    url, username, password);
```

```
var ps = cn.prepareStatement(  
    "select * from events where " +  
    "organizer_id=?");  
ps.setInt(1, 101);
```

```
var rs = ps.executeQuery();  
while (rs.next()) {  
    int id = rs.getInt(1);  
    String name = rs.getString(2);  
    // ...  
}
```

```
rs.close();  
ps.close();  
cn.close();
```

◀ Get a connection

◀ Prepare a statement

◀ Execute it

◀ Get the results

◀ Close the result set, statement
and connection



```
public interface EventRepository extends JpaRepository<Event, Integer> {  
    List<Event> findByOrganizerId(int organizerId);  
}
```

Defining a Repository

Just define an interface, Spring Data will implement it automatically



Spring Data in the Demo Application

MongoDB

NoSQL database

Spring Data MongoDB

PostgreSQL

Relational database

Spring Data JPA



Running MongoDB in Docker



| Storing Registrations in MongoDB



| Storing Event Information in PostgreSQL



| Externalized Configuration and | Initializing the Database



Configuration Using Spring Profiles



Summary



Spring Data

Running databases in Docker

Externalized configuration and profiles



Up Next:

Calling REST Web Services

