

Bootiful Spring Applications

Developing CRUD REST Services using Spring DATA JPA



Rajapandian C

IT CREDITI

c.rajapandian@centrico.tech

Overview



Features of Spring Boot

- Spring Boot starters
- Auto configuration

Bootstrapping an application

- Spring Initializr

Data access

REST API development

Testing

Spring Boot Overview

Spring

Spring Boot

Spring MVC

Spring Data

Spring Security

Spring Session

Spring AMQP

Spring Batch

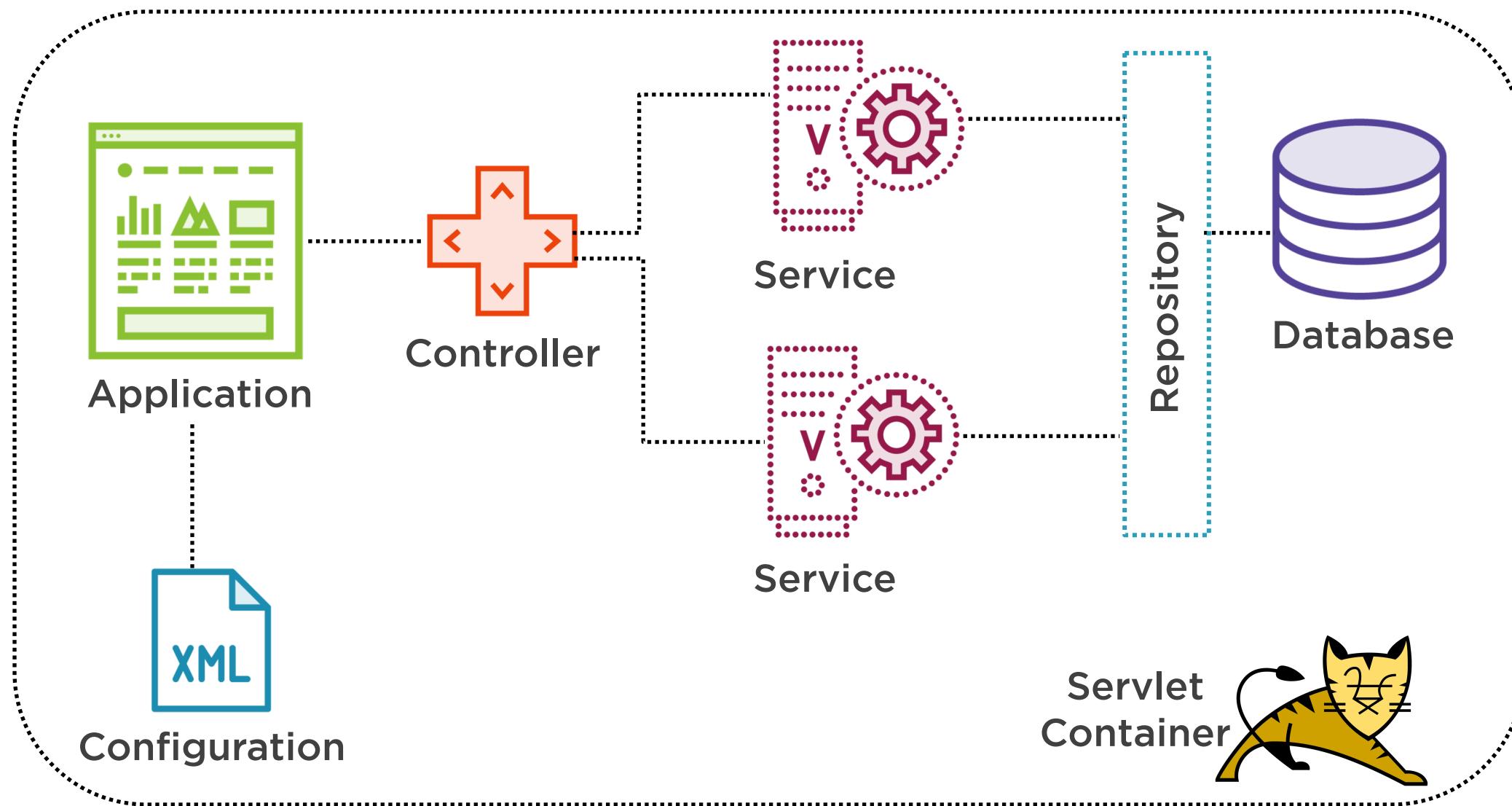
XML config files

Configuration needed

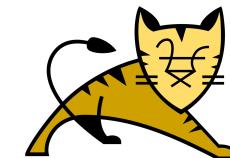
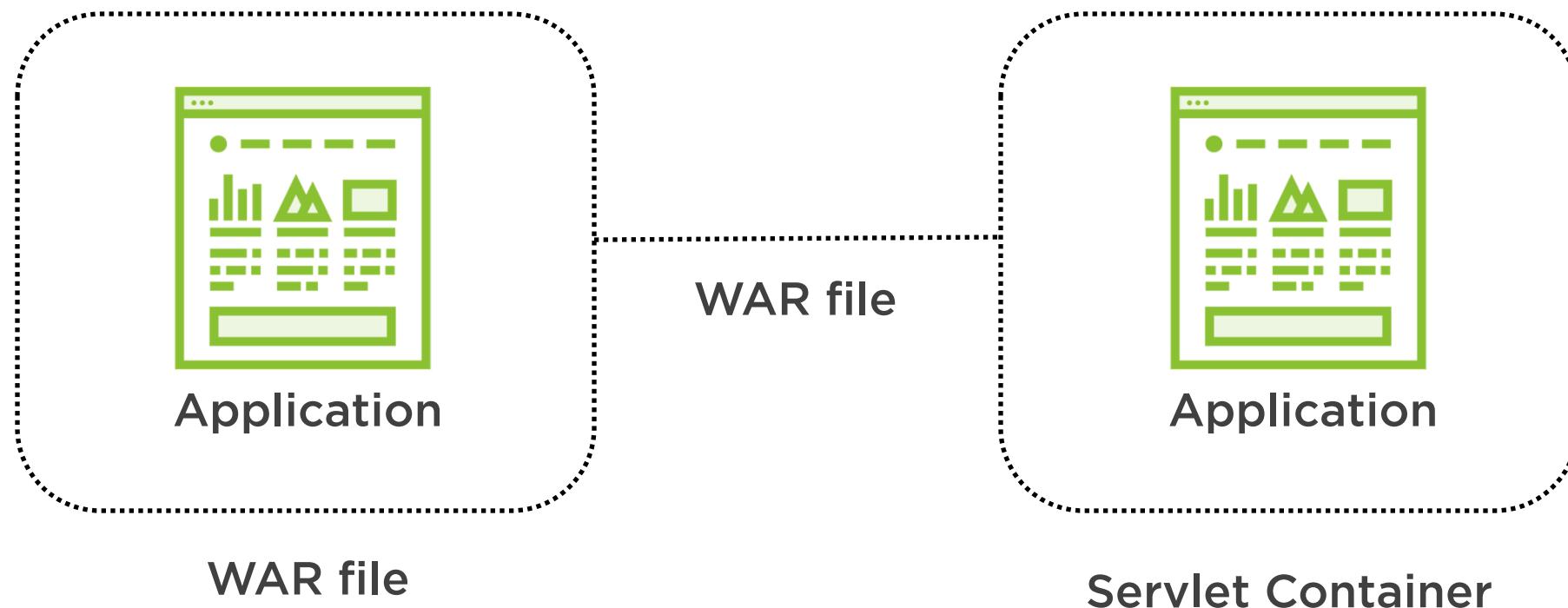
Annotation based

Configuration Challenges

Structure

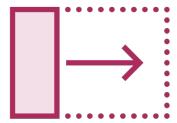


Simple Deployment



Features of Spring Boot

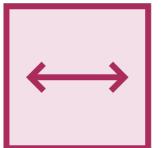
Features



Automatic configuration



Starter dependencies

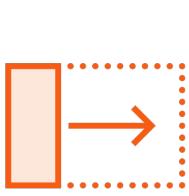


Command line interface



Actuator

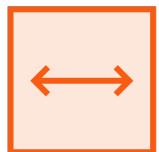
Automatic Configuration



Time saving feature



Configures application based on libraries



Maven dependencies configured automatically

Starter Dependency

spring-boot-starter-web

spring-boot-starter-test

spring-boot-starter-data-jpa

h2

Bootstrapping a Simple Application



Spring Initializr

Auto Configuration in Action

Auto-configuration is a very useful and powerful feature of Spring Boot, which takes a “convention-over-configuration” approach.

How it Works

Process

Finds JARs on the classpath
and auto-configures bean

Auto-configures

Data source for Hibernate or
DispatcherServlet for Spring
MVC

Automatic Configuration



Beans

Add your own beans, like a
datasource bean



Database Support

Default embedded database
support backs away

Annotations

@SpringBootApplication Annotation

@SpringBootConfiguration

Replaces @Configuration and annotates a class as a configuration

@EnableAutoConfiguration

Tells Spring Boot to configure beans

@ComponentScan

Tells Spring Boot to scan current package and subpackages

Accessing Data with Spring Boot and H2



H2 Database

H2 Database



Open source database written in Java



In-memory database



Good for POCs, dev environments, simple database



Administer via the H2 console

```
<dependency>  
    <groupId>com.h2database</groupId>  
    <artifactId>h2</artifactId>  
    <scope>runtime</scope>  
</dependency>
```

H2 Dependency
Auto-configures H2 related properties

ORM with JPA

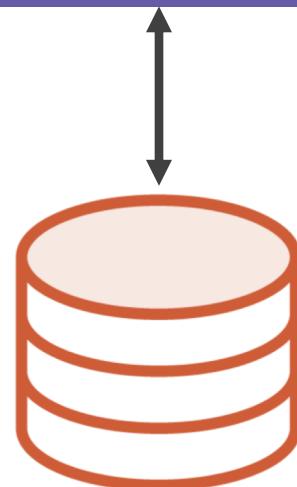
ORM with JPA

Spring Data JPA



Java Persistence API (JPA)

Java Database Connectivity (JDBC)



Spring Boot Properties

H2 Defaults

spring.datasource.url=jdbc:h2:mem:testdb

spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=

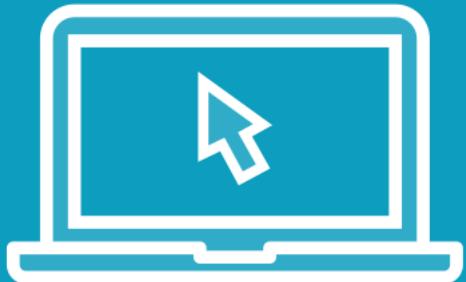
spring.h2.console.enabled=false

Override Defaults

`spring.h2.console.enabled=true`

`spring.h2.console.path=/h2`

Demo



Spring Initializr

Project in IDE

Launch application

H2 database access

Spring Data JPA

Entity

Repositories

Service

Controller

Summary



What we covered:

- Spring Boot
- Extension of Spring Framework

Spring Initializr

Automatic configuration

Annotations

- `@SpringBootApplication`

Starter dependencies