Overview

Spring Initializr

Automatic configuration

Annotations

Profiles

Spring Initializr

Demo

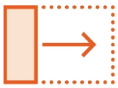
Spring Initializr

Project in IntelliJ

Launch application

Spring Boot CLI

Spring Boot CLI



Generate a project using the CLI

Some developers prefer over Spring Initilzr

Install CLI using Homebrew

Demo

Initialize a new project

Execute an application

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

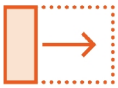
Beans

Add your own beans, like a datasource bean

Database Support

Default embedded database support backs away

Auto-Configuration Insights



Start application with --debug switch

Add a simple property to application.properties

Use the Spring Boot Actuator

Demo

Enable debug logging

Demo auto-configuration

Annotations

@SpringBootApplication Annotation

@SpringBootConfiguration

Replaces @Configuration and annotates a class as a configuration

@EnableAutoConfiguration

Tells Spring Boot to configure beans

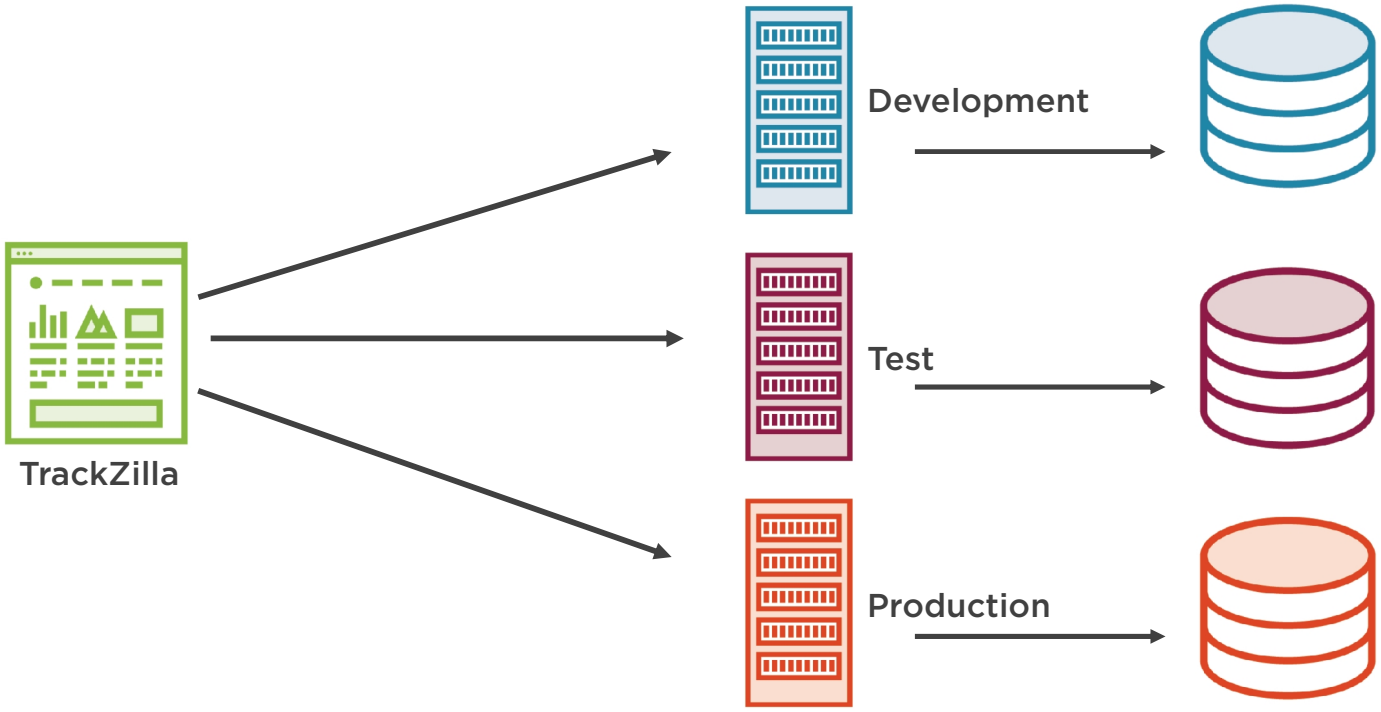
@ComponentScan

Tells Spring Boot to scan current package and subpackages

Spring Boot Properties

Spring Boot Profiles

Different Environments



```
spring.profiles.active = dev

applications-{profile}.properties

applications-dev.properties
applications-test.properties
applications-prod.properties
```

- Define active profile
- Naming format
- Dev profile
- Test/QA profile
- Prod profile

Summary

Spring Initializr

Spring Boot CLI

Auto-configuration

Annotations

- @SpringBootApplication

Spring Boot profiles