



spring

Spring Framework

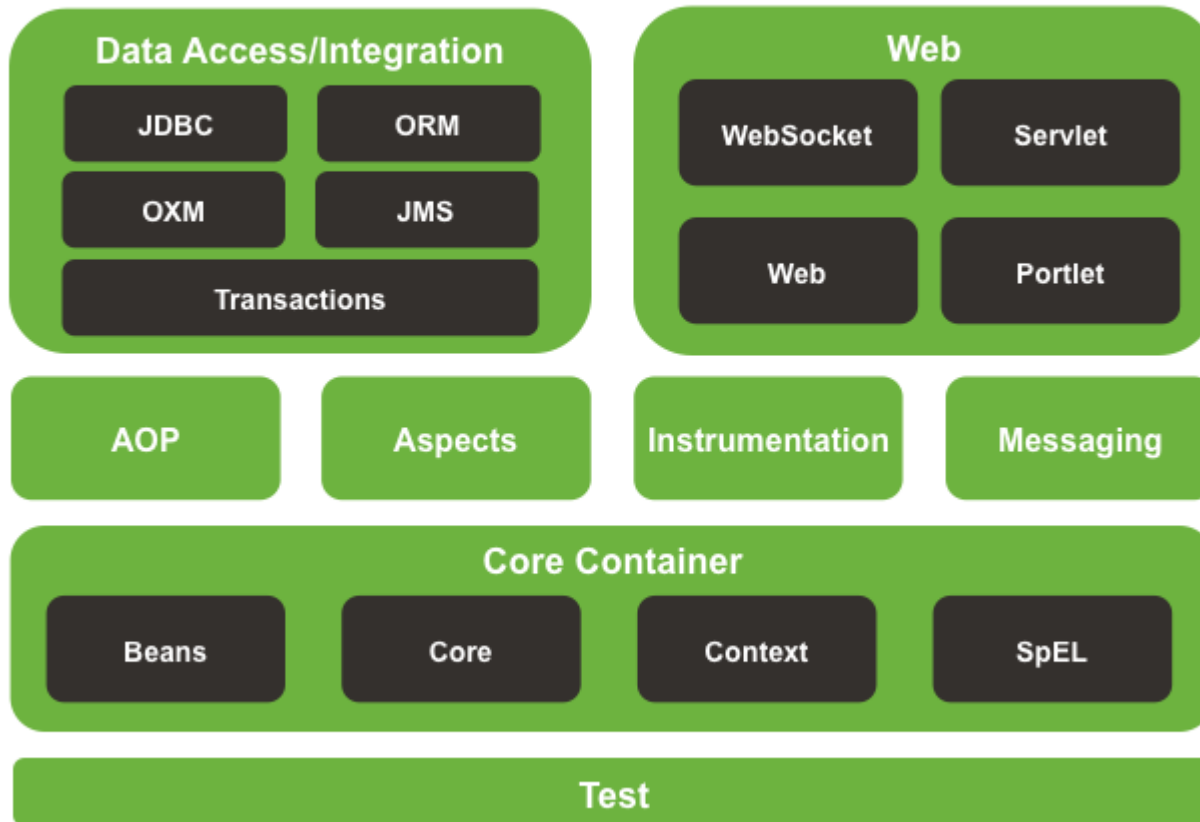
O que é Spring Framework?

- É um framework de apoio ao desenvolvimento de aplicações corporativas em Java que surgiu como uma alternativa à complexa plataforma J2EE, ganhando extrema popularidade.
- Provê uma série de funcionalidades, dentre as quais destacam-se **desenvolvimento de aplicações web e serviços REST, injeção de dependências, gerenciamento de transações, suporte a testes automatizados e suporte a programação orientada a aspectos.**
- Suporta diversos produtos Java populares tais como JPA, Hibernate, JSF, entre outros.
- Software livre, desenvolvido pela Pivotal.
- Pode ser utilizado em contêineres web, dispensando servidores de aplicações JEE como Glassfish e JBoss. Também pode ser utilizado em aplicações desktop.

Módulos



Spring Framework Runtime



Spring Framework 4 e Java

- Exige JDK 6+.
- Suporta Java 8+.
- Suporta uma série de especificações JEE 7 tais como JPA 2.1, JMS 2.0, JTA 1.2 e Bean Validation 1.1.

Outros projetos Spring

- **Spring MVC** para desenvolvimento de aplicações web (módulo do Spring Framework).
- **Spring Security** para inserção de funcionalidades de autenticação e autorização.
- **Spring Data** para aplicações que usam novas tecnologias de armazenamento de dados como bancos NoSQL e serviços na nuvem.
- **Spring Social** para fácil integração com redes sociais.
- **Spring Web Flow** é uma extensão do Spring MVC para permitir a implementação de fluxos (wizards) de telas.
- **Spring Roo** para desenvolvimento RAD ao estilo Ruby on Rails.
- Além de outros.

Configurando Spring

```
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.ComponentScan;
import org.springframework.context.annotation.Configuration;
import org.springframework.web.servlet.ViewResolver;

@Configuration
@ComponentScan(basePackages = "br.edu.fateczl.CrudJavaWebSpring")
public class WebConfig implements WebMvcConfigurer {

    @Bean
    public ViewResolver internalResourceViewResolver() {
        InternalResourceViewResolver resolver = new InternalResourceViewResolver();
        resolver.setPrefix("/WEB-INF/view/");
        resolver.setSuffix(".jsp");
        return resolver;
    }
}
```

Contexto

- O **contexto do Spring** é o ambiente que realiza a instanciação de objetos (chamados de beans Spring) e a injeção de dependências.
- Anotações são utilizadas para indicar as classes dos beans Springs:
 - **@Service**: bean da camada de serviço.
 - **@Repository**: bean da camada de persistência.
 - **@Controller**: bean que atua como controlador web.
 - **@Component**: bean genérico.
- A anotação **@Autowired** é utilizada para realizar a injeção de dependências.

Entendendo as configurações

- Spring MVC adota o padrão front controller. **DispatcherServlet** é o objeto que captura as requisições HTTP e, com base nas URLs, executa os métodos definidos nos controladores web (classes marcadas com **@Controller**).
- Devido à herança de **AbstractAnnotationConfigDispatcherServletInitializer**, **DispatcherServlet** utilizará as configurações definidas em nossa classe **SpringMVCServlet**.

Entendendo as configurações

- Front controller no Spring MVC

