

Minicurso Spring e Android

Criado por Gabriel Schaidhauer - Aula 1



Framework

O que é um Framework?

Um conjunto de classes que agrupa funcionalidades comuns dentre uma série de aplicações.



Mas como assim?

- Acesso a bancos de dados
- Cálculo de imposto de renda
- Externalização de serviços
- Controle de acesso
- Validação de dados

- ?
- ?
- ?
- ?
- ?

Mas como assim?

- Acesso a bancos de dados
- Cálculo de imposto de renda
- Externalização de serviços
- Controle de acesso
- Validação de dados

- Sim
- Não
- Sim
- Sim
- Sim

Tá, mas não é tipo uma biblioteca?



Tá, mas não é tipo uma biblioteca?



Princípios

Hollywood Principle

“Don’t Call Us, We’ll Call
You!”

- Sugarloaf

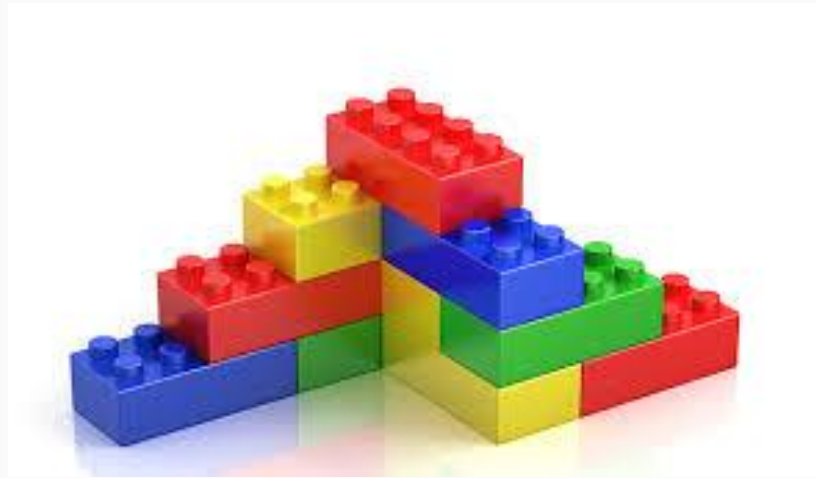


Hollywood Principle

- Inversão de controle
- Interfaces formalizando componentes de acordo com o framework
- Registro de comportamentos
- Registro de classes
- ETC.

Extensibilidade e flexibilidade

Flexibilidade e extensibilidade garantem que será possível criar a sua aplicação, os comportamentos específicos que precisa mesmo usando um framework.



Extensibilidade e flexibilidade

- Herança
- Sobrecarga
- Sobrescrita
- Implementação

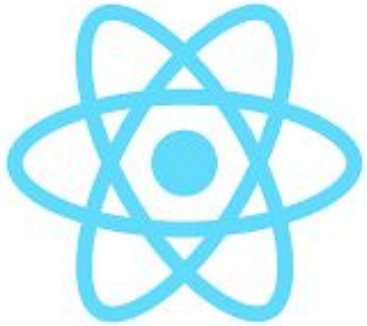
Reusabilidade

Mais do que um princípio
reusabilidade é o objetivo
de um framework.



Exemplos

Exemplos



Spring Framework

Spring Framework

Spring é um framework para aplicações java completo e modular, que pode ser usado em uma aplicação de ponta a ponta, tanto para acesso a dados e exposição de recursos REST quanto para ser ele mesmo o próprio servidor de aplicação.

Spring Framework - Principais Módulos

- Spring Framework
- Spring Data
- Spring Security
- Spring Boot

Spring Framework

O Spring Framework é o core da aplicação contendo as principais funcionalidades necessárias para o desenvolvimento.



Spring Data

O Spring Data é um módulo que adiciona a capacidade de interação com bases de dados através de implementações do JPA



Spring Security

Spring security adiciona uma camada de controle de acesso a aplicação bloqueando recursos não públicos e permitindo que sejam utilizados meios de autenticação personalizados



Spring Framework - Funcionamento

- Anotações
- Injeção de dependências
- Beans
- Configuração

Anotações

Anotações são uma das principais formas de interagir com o spring.

Anotações são detectadas durante fases de validação do framework, e aplicados os comportamentos que elas definem.

```
1 @RestController
2 @RequestMapping("/api")
3 public class MyClass {
4
5     @RequestMapping("/")
6     public String test () {
7         return "test";
8     }
9 }
```

Injeção de dependências

Isto permite que seja possível usar instâncias únicas (Singletons) sem precisar instancia-los ou controlar suas instâncias delegando esta responsabilidade para o framework

```
1 public class DependencyInjection {  
2  
3     @Autowired  
4     private MyRepository myRepository;  
5  
6     @Autowired  
7     public DependencyInjection (MyService myService) {  
8  
9         ...  
10    }  
11 }  
12
```


Beans

Beans são instâncias de objetos controladas pelo Spring, e registradas em uma classe de configuração através de um método que configura e exporta cada um dos beans.

```
1  @Configuration
2  public class Application () {
3
4      @Bean
5      public MyClass configureMyClass () {
6          return new MyClass();
7      }
8  }
9
```

Configuração

O Spring possui uma série de configurações tais como dados para envio de email, string de conexão de banco de dados, etc. as quais podem ser realizadas através do java ou arquivos xml ou properties.

```
@Configuration
public class MailConfig {

    @Value("${email.host}")
    private String host;

    @Value("${email.from}")
    private String from;

    @Value("${email.subject}")
    private String subject;

    @Bean
    public JavaMailSender javaMailService() {
        JavaMailSenderImpl javaMailSender = new JavaMailSenderImpl();
        javaMailSender.setHost(host);
        return javaMailSender;
    }

    @Bean
    public SimpleMailMessage simpleMailMessage() {
        SimpleMailMessage simpleMailMessage = new SimpleMailMessage();
        simpleMailMessage.setFrom(from);
        simpleMailMessage.setSubject(subject);
        return simpleMailMessage;
    }
}
```

Servidores de Aplicação

Servidores de Aplicação - Conceito

Servidores de aplicação são softwares criados com o intuito de controlar elementos para que a aplicação rodando neles não precise se preocupar com isso. As coisas que eles controlam podem variar, mas vão desde configurações de segurança até questões de balanceamento de carga e limitação de acessos simultâneos.

Servidores de Aplicação - Vantagens

Dentro do ambiente Java é impensável se trabalhar sem um servidor de aplicação, seja ele o Tomcat, Glassfish ou Wildfly, estes servidores controlam a disponibilização das nossas aplicações em rede, a url de acesso, dentre uma série de coisas que nem sempre acabamos pensando, mas que estão lá para que não precisemos pensar.

Servidores de Aplicação - Desvantagens

- Funcionalidades que não são agnósticas quanto a servidores de aplicação;
- Hospedagens que só trabalham com um servidor de aplicação;
- Complexidade de configuração.

O que fazer então?



Spring Boot

- Servidor de aplicação embutido
- Criado para suportar microsserviços
- Baixa complexidade
- Só é necessário ter o java na máquina

Funcionamento do Spring Boot

Para funcionar é necessário possuir uma classe com um método 'main' inicializando o contexto do Spring Boot.

```
package hello;

import org.springframework.boot.*;
import org.springframework.boot.autoconfigure.*;
import org.springframework.stereotype.*;
import org.springframework.web.bind.annotation.*;

@Controller
@EnableAutoConfiguration
public class SampleController {

    @RequestMapping("/")
    @ResponseBody
    String home() {
        return "Hello World!";
    }

    public static void main(String[] args) throws Exception {
        SpringApplication.run(SampleController.class, args);
    }
}
```

Dúvidas?

