

Injeção de Dependências no Android com Hilt

Pedro Henrique Silva

Senior Software Engineer

Sobre Mim



- Sou de Araraquara interior de SP
- Tenho 31 anos
- Moro em São Paulo capital há 5 anos
- Trabalho com desenvolvimento desde 2014
- Foco em Android desde 2015

<https://www.linkedin.com/in/pedro-henrique-silva/>

Objetivo Geral

O objetivo principal desse curso é entender o que é injeção de dependências e aplicar em aplicativos Android utilizando Hilt.

Pré-requisitos

- Conhecimento em Android
- Conhecimento em Kotlin

Injeção de Dependências no Android com Hilt

Etapa 1

O que é injeção de dependências?

Etapa 2

Visão geral sobre o Hilt

Etapa 3

Primeiros passos com Hilt

Etapa 4

Refatorando um projeto

Etapa 1

O que
é injeção de dependências?

Injeção de Dependências no Android com Hilt

Etapa 1

O que é injeção de dependências?

Etapa 2

Visão geral sobre o Hilt

Etapa 3

Primeiros passos com Hilt

Etapa 4

Refatorando um projeto

Injeção de dependências

Conjunto de padrões e princípios com o intuito de criar softwares sustentáveis com paradigma orientado a objetos, aplicando principalmente o princípio de inversão de controle (IoC).

Inversão de Controle (IoC)

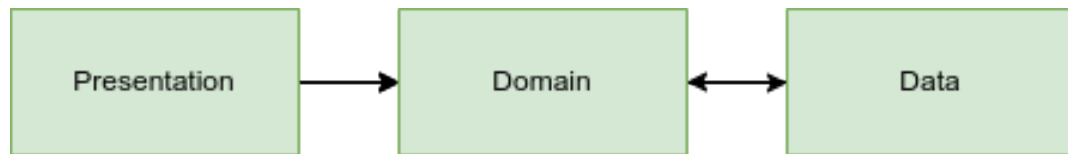
Basicamente é quando tiramos o controle da criação de objetos em uma classe e delegamos essa responsabilidade para seu exterior, invertendo assim a sequência de controle e o quanto uma classe conhece outra.

Técnicas de injeção de dependência

- Via construtor
- Propriedade
- Método

Dependency Inversion Principle

- Módulos de alto nível não devem depender de módulos de baixo nível e ambos devem depender de abstrações;
- Abstrações não devem depender de detalhes, mas detalhes devem depender de abstrações.



Single Responsibility Principle

Cada módulo ou classe deve ter responsabilidade sobre uma única parte da funcionalidade fornecida pelo software.

Etapa 2

Visão geral sobre o Hilt

Injeção de Dependências no Android com Hilt

~~Etapa 1~~

~~O que é injeção de dependências?~~

Etapa 2

Visão geral sobre o Hilt

Etapa 3

Primeiros passos com Hilt

Etapa 4

Refatorando um projeto

Hilt

O Hilt é uma biblioteca de injeção de dependências para Android que reduz o código boilerplate criado por injeções manuais no projeto.

O Hilt foi criado com base na biblioteca de injeção de dependências Dagger, oferecendo uma maneira padrão de incorporar o Dagger a um app Android.

Dagger

Biblioteca popular para Java que foi amplamente utilizada no Android.

Utiliza processamento de anotações para gerar código que a gente teria que escrever caso a injeção de dependências fosse feita manualmente.

Módulos do Hilt

São os "containers" com as definições das dependências e como elas são construídas.

Geralmente utilizamos quando precisamos quando é necessário prover uma instância de uma classe de biblioteca externa ou implementação de uma abstração.

Os módulos sempre são atrelados/instalados em um componente.

Componentes Hilt

Definem o ciclo de vida e escopo das instâncias providas pelos módulos.

Vantagens

- Interoperável com Dagger
- Suporte a componentes do AndroidX
- Redução de boilerplate causados pela injeção manual
- Redução de boilerplate por conta do Dagger
- Testabilidade

Etapa 3

Primeiros passos com Hilt

Injeção de Dependências no Android com Hilt

~~Etapa 1~~

~~O que é injeção de dependências?~~

~~Etapa 2~~

~~Visão geral sobre o Hilt~~

Etapa 3

Primeiros passos com Hilt

Etapa 4

Refatorando um projeto

Etapa 4

Refatorando um projeto

Injeção de Dependências no Android com Hilt

~~Etapa 1~~

~~O que é injeção de dependências?~~

~~Etapa 2~~

~~Visão geral sobre o Hilt~~

~~Etapa 3~~

~~Primeiros passos com Hilt~~

Etapa 4

Refatorando um projeto

Para saber mais

- <https://developer.android.com/codelabs/android-hilt>
- <https://developer.android.com/training/dependency-injection>
- <https://developer.android.com/training/dependency-injection/hilt-android>
- <https://dagger.dev/hilt/>
- Livro - Arquitetura limpa: O guia do artesão para estrutura e design de software

Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)

