



INJEÇÃO DE DEPENDÊNCIAS E SEUS CONCEITOS PARTE 2



O que preciso saber?

PRIMEIRA SEMANA

1. Component-Based Architectural Style
2. Inversion of Control
3. Dependency Inversion
4. Class Coupling

SEGUNDA SEMANA

5. Service Locator
6. Dependency Injection
7. Post-Construction Resolve

RESTO DA SUA VIDA

Service Locator Pattern

A idéia de um service locator é ter um objeto que saiba como obter todos os serviços(instâncias de objetos) que uma aplicação pode precisar.

Service Locator é um Anti-Pattern ?

Vantagens

- Ajuda a desacoplar classes de suas dependências.
- Ajuda a escrever classes que dependem de abstrações ao invés de implementações.
- Resolve o problema de como instanciar novas classes, não é mais necessário criar milhões de factories.
- Torna possível testar as classes utilizando mocks.

A top-down view of a wooden desk with a warm, orange-toned overlay. On the desk is an open laptop with a silver keyboard, a white ceramic cup of dark coffee, two wooden pencils, a small lined notepad, and several pieces of crumpled paper. The text "SHOW ME THE CODE!" is centered in a bold, white, sans-serif font.

SHOW ME THE CODE!

Service Locator é um Anti-Pattern ?

Desvantagens

- Agora todo o seu sistema está acoplado ao service locator.
- Esconde as dependências das classes as tornando mais complexas para entender e testar.
- Ao contrário de um container de DI não é possível validar se todos os registros estão corretos.

Injeção de Dependências

Dependency Injection - DI

A injeção de dependências separa a criação(new) das dependências de uma classe do comportamento da mesma, o que permite que as classes sejam fracamente acopladas e sigam os princípios de inversão de dependência e responsabilidade única. Ele contrasta diretamente com Service Locator, que permite às classes saber o que usam para encontrar dependências.

Injeção de Dependências

Dependency Injection - DI

A injeção de dependência tem quatro papéis:

- Os objetos de serviço a serem utilizados.
- Os objetos clientes que dependem dos serviços que utilizam .
- As interfaces que definem como os clientes podem utilizar seus serviços
- O injetor, que é responsável pela construção dos serviços e injetá-los no cliente.

A top-down view of a wooden desk with a warm, orange-toned overlay. On the desk is an open laptop with a silver keyboard, a white ceramic cup of dark coffee, two wooden pencils, a small lined notepad, and several pieces of crumpled paper. The text "SHOW ME THE CODE!" is centered in a bold, white, sans-serif font.

SHOW ME THE CODE!