# Bean Overview

### Tradução e Resumo

Na documentação sobre a **definição de beans** no Spring Framework, é explicado que os **beans** são componentes fundamentais no Spring IoC (Inversion of Control). Um **bean** é qualquer objeto que o contêiner do Spring gerencia. A definição de um **bean** pode incluir metadados que determinam como o Spring deve instanciar, configurar e gerenciar esse objeto. Os **beans** são geralmente configurados por meio de **classes de configuração Java**, anotações ou arquivos XML.

Cada **bean** possui características como um **ID único**, seu **tipo**, o **escopo** (singleton ou prototype), e as **dependências** que precisam ser injetadas. Além disso, você pode configurar como o **bean** será inicializado ou destruído. O Spring permite a personalização de todos esses aspectos para facilitar a criação de aplicações modulares e desacopladas.

### Pontos-Chave

* **Definição de bean**: Metadados que definem como o Spring deve gerenciar um objeto.
* **ID de bean**: Identificador único do **bean** dentro do contêiner.
* **Tipo de bean**: Classe à qual o **bean** pertence.
* **Escopo do bean**: Define o ciclo de vida do **bean** (ex: **singleton**, **prototype**).
* **Dependências do bean**: Objetos necessários que são injetados pelo contêiner.
* **Inicialização e destruição**: Métodos para gerenciar o ciclo de vida completo de um **bean**.

### Exemplo Prático

Aqui está um exemplo básico de como definir um **bean** usando **Java Configuration** no Spring:

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

@Configuration

public class AppConfig {

@Bean

public MyBean myBean() {

return new MyBean();

}

}

public class MyBean {

public void execute() {

System.out.println("Bean executando!");

}

}

Neste exemplo, usamos a anotação @Configuration para marcar a classe AppConfig como uma classe de configuração do Spring, e a anotação @Bean para definir o **bean** MyBean.

### Tópicos Principais

* Definição de **beans** com **Java Configuration**.
* **Identificação de beans** com IDs únicos.
* **Escopo dos beans**: **singleton**, **prototype** e outros.
* **Injeção de dependências** dentro de **beans**.
* **Métodos de inicialização e destruição** de **beans**.

### Exemplo Curto

Aqui está um exemplo de definição de um **bean** com escopo **prototype** e um método de destruição personalizado:

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.context.annotation.Scope;

@Configuration

public class AppConfig {

@Bean

@Scope("prototype")

public MyPrototypeBean myPrototypeBean() {

return new MyPrototypeBean();

}

}

public class MyPrototypeBean {

public void cleanup() {

System.out.println("Bean prototype sendo destruído...");

}

}

Neste exemplo, MyPrototypeBean é definido com o escopo **prototype**, o que significa que uma nova instância será criada a cada vez que o **bean** for solicitado. O método cleanup() pode ser chamado manualmente para realizar a destruição do **bean**, uma vez que o Spring não gerencia a destruição automática de **beans prototype**.

Script

Estou estudando documentação em inglês sobre **Beans (Spring framework) no link https://docs.spring.io/spring-framework/reference/core/beans/basics.html**. Preciso que organize o conteúdo da seguinte forma:

Traduza e resuma o texto para português, destacando os pontos mais importantes. Faça o resumo ser objetivo, com no máximo dois parágrafos.

Adicione exemplos práticos baseados no que foi discutido no texto. Pode incluir exemplos de código ou casos de uso que ajudem a entender o conceito.

Destaque com negrito os conceitos ou termos chave que devem ser memorizados.

Crie uma lista com bullet points dos principais tópicos abordados no texto.

Ao final, inclua um exemplo curto e objetivo em Java, relacionado ao conteúdo resumido, que aplique algum conceito prático discutido no texto.