1. Какво представляват Beans в Spring?
2. PostConstruct анотация?
3. PreDestroy анотация?
4. Има ли разлика между appilcation.properties и application.yaml (отнася се за предназначението).
5. Какви Beans scopes познавате, дайте примери и какви са разликите между тях?
6. Какво е Spring Boot?
7. Само на Java ли се пишат спринг приложения?

1.Какво представляват Beans в Spring?

1. Overview

Bean is a key concept of the Spring Framework. So understanding this notion is crucial to get the hang of the framework and use it in an effective way. Unfortunately, there aren't clear answers to the simple question of what a Spring bean really is. Some explanations go to such a low level that the big picture is missed, whereas others are too vague. This tutorial will try to shed light on the topic, starting with a description in the official documentation.

Bean е ключова концепция на Spring Framework. Така че разбирането на това понятие е от решаващо значение, за да овладеете рамката и да я използвате по ефективен начин. За съжаление, няма ясни отговори на простия въпрос какво всъщност представлява Spring Bean . Някои обяснения стигат до толкова ниско ниво, че голямата картина се пропуска, докато други са твърде неясни. Този урок ще се опита да хвърли светлина върху темата, като се започне с описание в официалната документация.

2. Bean Definition

Here's a definition of beans in [the Spring Framework documentation](https://docs.spring.io/spring/docs/current/spring-framework-reference/core.html#beans-introduction):

In Spring, the objects that form the backbone of your application and that are managed by the Spring IoC container are called beans. A bean is an object that is instantiated, assembled, and otherwise managed by a Spring IoC container.

This definition is concise and gets to the point but fails to elaborate on an important element: the Spring IoC container. Let's take a closer look to see what it is and the benefits it brings in.

Ето дефиниция за bean в документацията на Spring Framework:

В Spring обектите, които формират гръбнака на вашето приложение и които се управляват от Spring IoC контейнера, се наричат beans. Bean е обект, който се инстанцира, сглобява и управлява по друг начин от Spring IoC контейнер.

Тази дефиниция е кратка и стига до същността, но не успява да разработи важен елемент: контейнера Spring IoC. Нека разгледаме по-отблизо, за да видим какво представлява и ползите, които носи.

3. Inversion of Control

Simply put, [Inversion of Control](https://www.baeldung.com/inversion-control-and-dependency-injection-in-spring) (IoC) is **a process in which an object defines its dependencies without creating them.** This object delegates the job of constructing such dependencies to an IoC container.

Най-просто казано, инверсията на контрола (IoC) е процес, при който обект дефинира своите зависимости, без да ги създава. Този обект делегира работата по конструирането на такива зависимости на IoC контейнер.

Resource: <https://www.baeldung.com/spring-bean>