

1.3

Freitag, 5. November 2021 20:26

B) Spring Framework

-Worum handelt es sich bei dem Spring Framework?

Beim Spring Framework handelt es sich um umfangreiche Infrastruktur als Unterstützung für Java Applikationen und Libraries für development.

-Was ist ein Bean im Spring Framework?

-ein Bean bezeichnet man als Objekt, welches vom Spring IoC container instanziiert, zusammengebaut, oder verwaltet wird

-Was ist der Application Context im Spring Framework?

-eines der beiden Java Interfaces, welche den Spring IoC Container repräsentieren

-IoC = Inversion of Control

-Objekte definieren durch IoC andere Objekte als Dependencies

-Der Application Context stellt Funktionalitäten bereit, die Aufgaben und Anforderungen im Context einer Java App erfüllt.

-hält Instanzen von Java Objekten, welche identifiziert wurden, um verwaltet und aufgeteilt zu werden

-Was ist @Component im Spring Framework?

@Component im Spring Framework erlaubt uns ein Objekt als Komponente für unsere Web App zu definieren

@Component

```
Public class ComponentExample{  
}
```

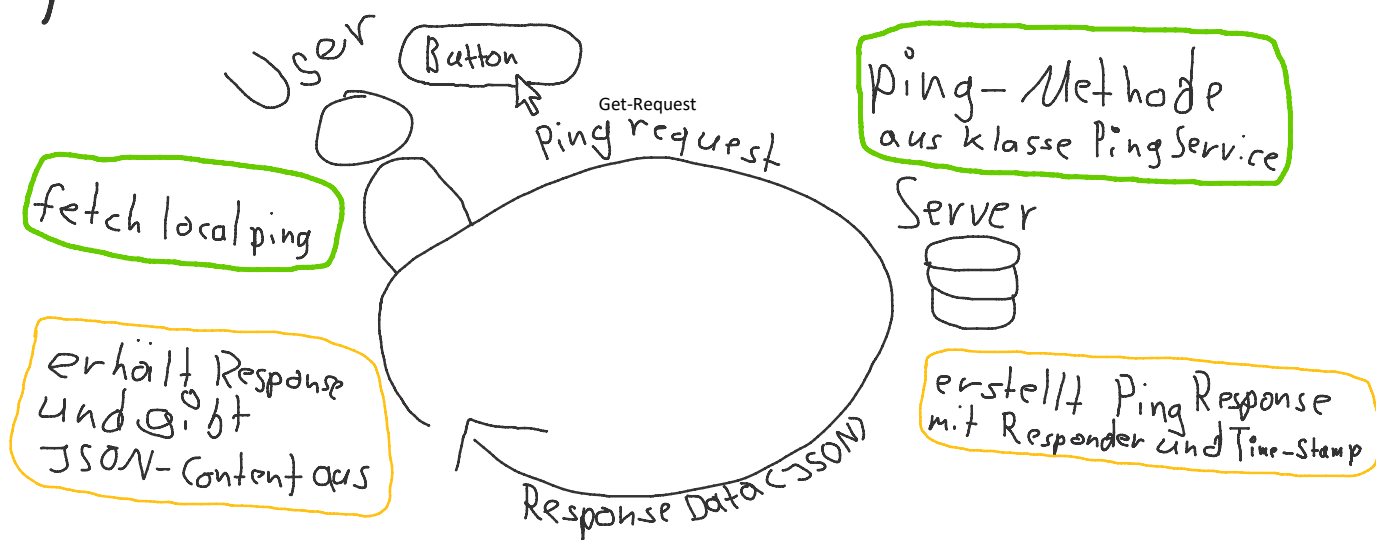
-Was ist @Autowired im Spring Framework?

@Autowired im Spring Framework erlaubt es uns Objekte, die mit @Autowired gekennzeichnet wurden, mit weiteren Beans auszustatten von Spring

-Was ist JSON?

-JSON(JavaScript Objekt Notation) ist ein Format, um Daten zu speichern und zu transportieren ebenfalls wird es oft genutzt um Daten vom Server zu einer Web Page zu schicken.

1.3 c)



Dependency Injection:

Dependency Injection ist ein Entwurfsmuster, welches ein Objekt von einer Klasse zu einer anderen Klasse zu Verfügung stellt ohne die beiden Klassen voneinander abhängig zu machen, sondern zum Beispiel eine Spring Container als zentralen Speicherort dafür zu setzen.

@component markiert die Klasse von der wir ein Objekt haben möchten und macht sie zu einem Spring Bean.

Dieses Objekt wird in einem Spring Container gelegt.

Mit `getBean(unsereKlasse)`; injektieren wir dann dieses Objekt in unsere Applikation gelegt und kann somit von einer Klasse auf ein Objekt einer anderen Klasse zugreifen ohne die beiden Klassen voneinander abhängig zu machen.

@Autowired:

Wenn wir zwei Objekte verschiedener Klassen in unserem Spring Container haben, kriegen wir einen Fehler "NullPointerException" weil objekt1 abhängig von objekt2 im Container ist und daher verbinden wir objekt1 und objekt2 miteinander.

Hier kommt @Autowired ins Spiel, wie das Wort schon sagt "automatisch verdrahtet" sagen wir mit @Autowired Objekt1 sucht automatisch nach Objekt2 im Container um eine Verbindung der Objekte zu garantieren.