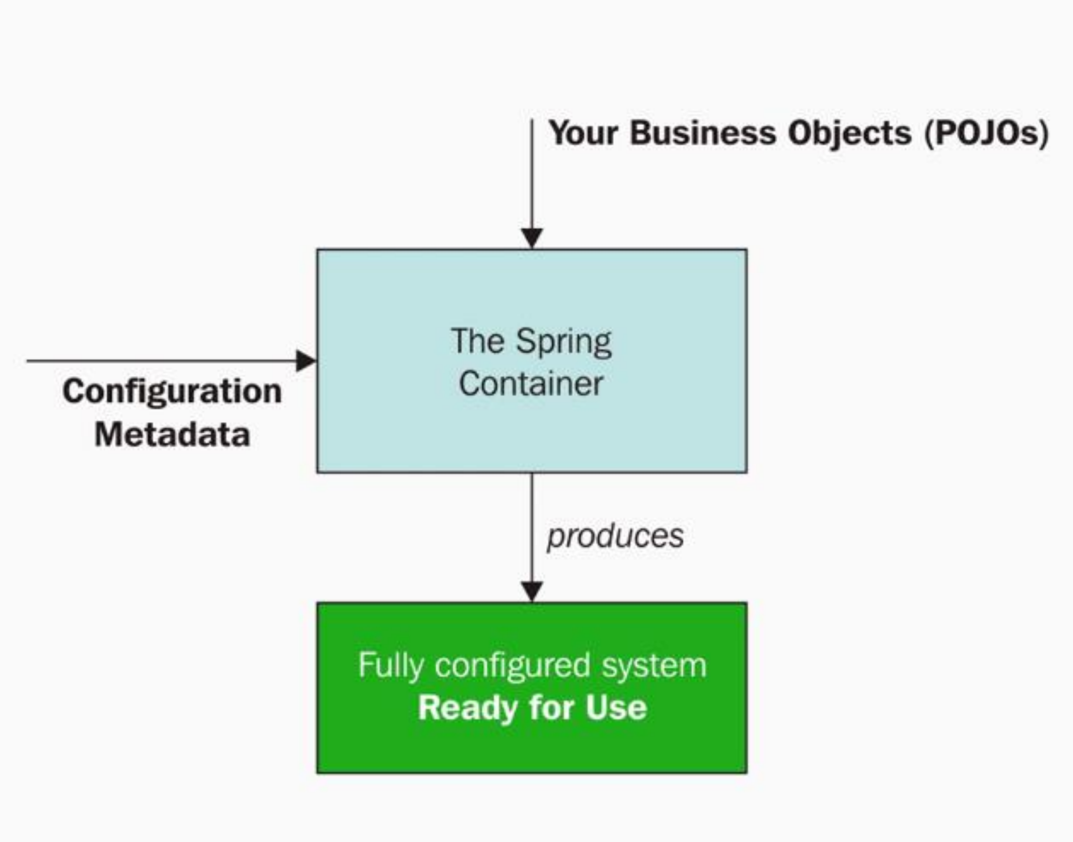
Spring Bean Lifecycle

Presented by Hosein Zare
Twitter: @zare88r

High Level View



1 - تعریف Bean ها

- جهت تعریف POJO ها به عنوان spring bean کافی است annotation
@Component را بالای سر آن بگذاریم

```java
@Component
public class Class1 {

}


public static void main(String[] args) {
    ConfigurableApplicationContext context =
            SpringApplication.run(FodApplication.class, args);
    context.getBean(Class1.class);
}
```

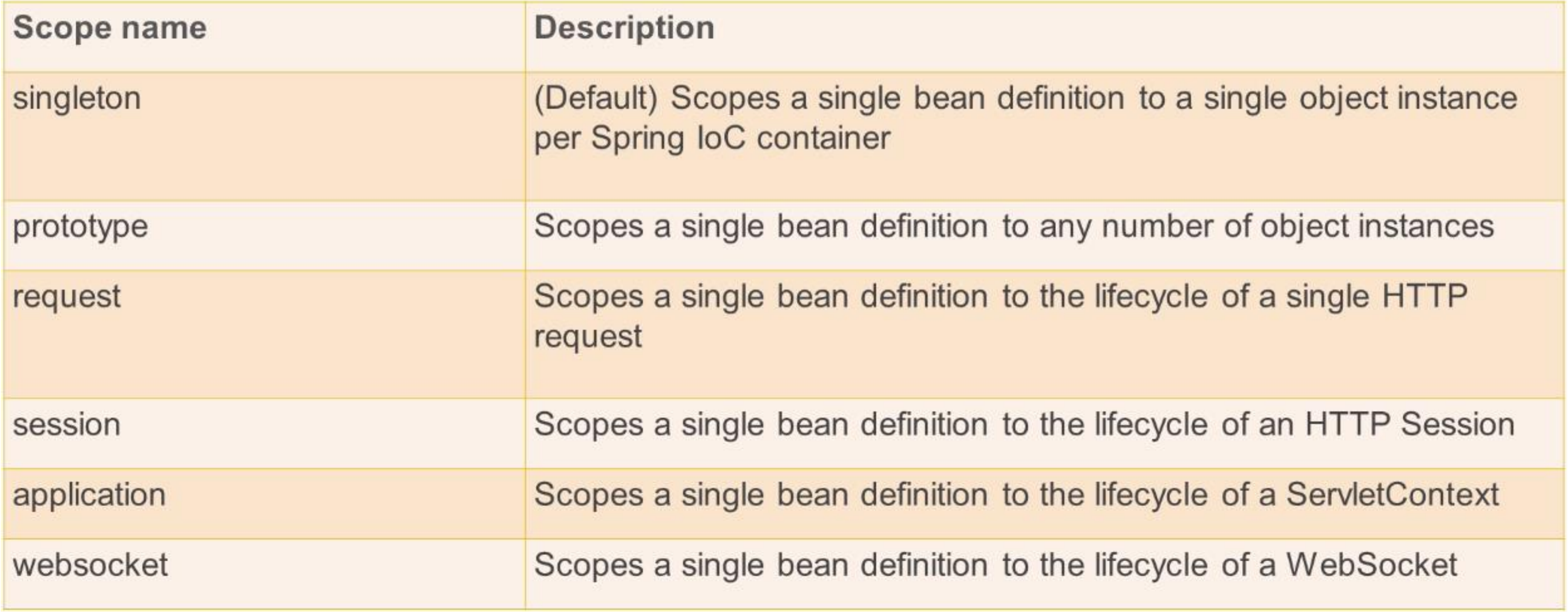
2 - تعریف Bean ها

- روش دوم تعریف bean با استفاده از متود و annotation @Bean میباشد

```
@Configuration
public class GeneralConfiguration {

    @Bean(initMethod = "init" , destroyMethod = "destroy")
    public Class1 class1(){
        return new Class1();
    }
}
```

Scopes



Scope name	Description
singleton	(Default) Scopes a single bean definition to a single object instance per Spring IoC container
prototype	Scopes a single bean definition to any number of object instances
request	Scopes a single bean definition to the lifecycle of a single HTTP request
session	Scopes a single bean definition to the lifecycle of an HTTP Session
application	Scopes a single bean definition to the lifecycle of a ServletContext
websocket	Scopes a single bean definition to the lifecycle of a WebSocket

Scopes

```
@Component
@Scope(scopeName = ConfigurableBeanFactory.SCOPE_PROTOTYPE)
public class Class1 implements InitializingBean {

....

System.out.println(context.getBean(Class1.class));
System.out.println(context.getBean(Class1.class));

...
```

```
com.demisco.fod.Class1@37ff4054
com.demisco.fod.Class1@2472c7d8
```

روش های DI

- سه روش اصلی برای inject کردن وابستگی ها وجود دارد :
  - Constructor
  - Annotation (@Autowired)
  - Setter Method

```
@Bean
public SimpleDateFormat persianDateFormat() {
    SimpleDateFormat simpleDateFormat =
            new SimpleDateFormat("yyyy/MM/dd"
                    , new Locale("fa", "IR"));
    return simpleDateFormat;
}
```

Constructor

```java
@Component
public class Class1 {

    private SimpleDateFormat persianDateFormat;

    public Class1(SimpleDateFormat persianDateFormat) {
        this.persianDateFormat = persianDateFormat;
    }
```

Annotation

```
@Autowired
private SimpleDateFormat persianDateFormat;


@Autowired
@Qualifier("persianDateFormat")
private SimpleDateFormat dateFormat;
```

Setter method

```
private SimpleDateFormat dateFormat;

@Autowired
@Qualifier("persianDateFormat")
public void setDateFormat(SimpleDateFormat dateFormat) {
    this.dateFormat = dateFormat;
}
```

Context Configuration methods

- Xml
- Java Annotations
- Groovy DSL