

Usare le annotation JSR 330 in Spring

Cosa imparerai

- Come usare le annotation JSR 330 @Inject, @Named, @ManagedBean

Usare le annotation JSR 330 in Spring

A partire dalla versione 3.0, Spring offre supporto per le annotation JSR-330, ovvero il set di API standard che definisco la Dependency Injection in Java.

Per poter utilizzare queste annotation (e tutte quelle del package javax.annotation.), se usiamo Maven è necessario impostare la dipendenza dall'artifact javax.inject.

```
<dependency>  
  <groupId>javax.inject</groupId>  
  <artifactId>javax.inject</artifactId>  
  <version>1</version>  
</dependency>
```

Usare le annotation JSR 330 in Spring

@Inject

Questa annotation consente di effettuare l'inject di un bean, analogamente all'annotation *@Autowired*.

```
@Component
public class OrdineService {
    @Inject
    private ClienteService cliente;

    @Inject
    private ProdottoService prodotti;

    public String sayHello() {
        return "CIAO sono un ordine del cliente " +
            cliente.getClient(1231) + " " +
            prodotti.sayHello();
    }
}
```

Usare le annotation JSR 330 in Spring

@Named e @ManagedBean

Queste annotation sono l'equivalente dell'annotation *@Component* e sono utilizzate per definire una classe un bean.

```
@Named
public class OrdineService {
    @Inject
    private ClienteService cliente;

    @Inject
    private ProdottoService prodotti;

    public String sayHello() {
        return "CIAO sono un ordine del cliente " +
            cliente.getCliente(1231) + " " +
            prodotti.sayHello();
    }
}
```

```
@ManagedBean
public class OrdineService {
    @Inject
    private ClienteService cliente;

    @Inject
    private ProdottoService prodotti;

    public String sayHello() {
        return "CIAO sono un ordine del cliente " +
            cliente.getCliente(1231) + " " +
            prodotti.sayHello();
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        ApplicationContext context = new ClassPathXmlApplicationContext("services.xml");
        OrdineService o = context.getBean("ordineService", OrdineService.class);
        System.out.println(o.sayHello());
    }
}
```

Di cosa abbiamo parlato in questa lezione

- Come usare le annotation JSR 330 @Inject, @Named, @ManagedBean