

Autowire con l'annotation @Autowired

Cosa imparerai

- Come implementare l'autowire senza notazione XML

Autowire con l'annotation @Autowired

Quando definiamo l'autowire con la notazione XML utilizziamo l'attributo **autowire** sull'elemento `<bean .../>`.

```
<bean id="rubricaService" class="it.test.service.RubricaService" autowire="constructor">
  <constructor-arg name="messaggiService" ref="messaggiService"></constructor-arg>
</bean>
```

L'annotation **@Autowired** consente di implementare l'autowire senza utilizzare la notazione XML.

Possiamo utilizzare l'annotation su:

- ☐ Costruttore
- ☐ Metodo set
- ☐ Metodi che non implementano il set di variabili
- ☐ Variabili di istanza che siano dei bean (anche in combinazione con l'autowired sul costruttore)
- ☐ Array e collection di bean

Autowire con l'annotation @Autowired

@Autowired sul costruttore

A partire da Spring Framework 4.3, l'annotazione **@Autowired** sul costruttore non è necessaria se il bean di destinazione definisce solo un costruttore (1).

Se sono disponibili più costruttori e non esiste un costruttore predefinito, almeno uno dei costruttori deve essere annotato con *@Autowired* per indicare al contenitore quale usare (2) e (3).

```
@Component(value = "ordine")
public class OrdineService {
    private ClienteService cliente;

    1 public OrdineService(ClienteService cliente) {
        super();
        this.cliente = cliente;
    }

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

CIAO sono un ordine del cliente Mario Rossi

```
@Component(value = "ordine")
public class OrdineService {
    private ClienteService cliente;

    2 public OrdineService(ClienteService cliente) {
        super();
        this.cliente = cliente;
    }

    public OrdineService() {
        super();
    }

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

```
Exception in thread "main" java.lang.NullPointerException:
"it.test.service.ClienteService.getCliente(java.lang.Lo
"this.cliente" is null
    at it.test.service.OrdineService.sayHello(Ordin
    at it.test.Main.main(Main.java:14)
```

```
@Component(value = "ordine")
public class OrdineService {
    private ClienteService cliente;

    3 @Autowired
    public OrdineService(ClienteService cliente) {
        super();
        this.cliente = cliente;
    }

    public OrdineService() {
        super();
    }

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

CIAO sono un ordine del cliente Mario Rossi

Autowire con l'annotation @Autowired

@Autowired sul metodo set

In questo caso, l'annotation viene applicata sul metodo set del bean da iniettare.

```
@Component(value = "ordine")
public class OrdineService {
    private ClienteService cliente;

    @Autowired
    public void setCliente(ClienteService cliente) {
        this.cliente = cliente;
    }

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

Autowire con l'annotation @Autowired

@Autowired su un metodo che non implementa il set di variabili

In questo caso, l'annotation viene applicata su un metodo normale, che ha come argomento (o argomenti) dei bean.

```
@Component(value = "ordine")
public class OrdineService {
    private ClienteService cliente;
    private ProdottoService prodotto;

    @Autowired
    public void recuperaServices(ClienteService cliente, ProdottoService prodotto) {
        this.cliente = cliente;
        this.prodotto = prodotto;
    }

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

Autowire con l'annotation @Autowired

@Autowired su variabili di istanza che siano dei bean

In questo applichiamo l'annotation sulla variabile di istanza (1).

È possibile utilizzare anche l'annotation sulla variabile di istanza e sul costruttore insieme (2).

```
@Component(value = "ordine")
public class OrdineService {
    @Autowired
    private ClienteService cliente;
    1 @Autowired
    private ProdottoService prodotto;

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

```
@Component(value = "ordine")
public class OrdineService {
    private ClienteService cliente;

    @Autowired
    private ProdottoService prodotto;
    2 @Autowired
    public OrdineService(ClienteService cliente) {
        super();
        this.cliente = cliente;
    }

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

Autowire con l'annotation @Autowired

@Autowired su array e collection di bean

In questo applichiamo l'annotation sulla variabile di istanza che sia un array o una collection di bean.

```
@Component(value = "ordine")
public class OrdineService {
    private ClienteService cliente;

    @Autowired
    private ProdottoService[] prodotti;

    @Autowired
    public OrdineService(ClienteService cliente) {
        super();
        this.cliente = cliente;
    }

    public String sayHello() {
        return "CIAO sono un ordine del cliente " +
            cliente.getClient(1231) + " " +
            prodotti[0].sayHello();
    }
}
```

```
@Component(value = "ordine")
public class OrdineService {
    private ClienteService cliente;

    @Autowired
    private List<ProdottoService> prodotti;

    @Autowired
    public OrdineService(ClienteService cliente) {
        super();
        this.cliente = cliente;
    }

    public String sayHello() {
        return "CIAO sono un ordine del cliente " +
            cliente.getClient(1231) + " " +
            prodotti.get(0).sayHello();
    }
}
```



Di cosa abbiamo parlato in questa lezione

- Come implementare l'autowire senza notazione XML