

Spring Dependency Injection with java Annotation and Autowiring

What is Autowiring and how it works?

When it sees **@Autowired**, **Spring** will look for a class that matches the property in the applicationContext, and inject it automatically. ... Only one constructor (at max) of any given bean class may carry this annotation, indicating the constructor to **autowire** when used as a **Spring** bean.

The **@Autowired** annotation is performing Dependency Injection.

If **@Autowired** is applied to

- **a field:** then the dependency is stored in this field
- **a setter:** then the setter is invoked, with the parameter that is determined by the same algorithm like for the field dependency injection
- **a constructor:** then the constructor is invoked with the parameters determined by the same algorithm like for the field dependency injection

Constructor Injection:

@Autowired

```
public EngineeringTeam(EmployeeService empService) {  
    this.empService = empService;  
}
```

Note:

Spring will scan for a component that implements EmployeeService interface

our example: MyEmployeeService meets the requirements

- Injecting EmployeeService in to Manager Implementation

- Spring will scan @Components