

Questions

- What is loose coupling?
- What is a Dependency?
- What is IOC (Inversion of Control)?
- What is Dependency Injection?
- Can you give few examples of Dependency Injection?
- What is Auto Wiring?

Questions

- What are the important roles of an IOC Container?
- What are Bean Factory and Application Context?
- Can you compare Bean Factory with Application Context?
- How do you create an application context with Spring?

Tight Coupling

```
public class TodoBusinessService {  
    TodoDataServiceImpl dataService = new TodoDataServiceImpl();  
  
    public class ComplexAlgorithmImpl {  
        BubbleSortAlgorithm bubbleSortAlgorithm  
            = new BubbleSortAlgorithm();  
    }  
}
```

Loose Coupling

```
@Component
public class TodoBusinessService {

    @Autowired
    TodoDataService dataService; // = new TodoDataService()

    public TodoBusinessService(TodoDataService dataService) {
        this.dataService = dataService;
    }

    public interface TodoDataService {
        List<String> retrieveTodos(String user);
    }
}
```

Loose Coupling

```
@Component
public class ComplexAlgorithmImpl {

    @Autowired
    private SortAlgorithm sortAlgorithm;

    public interface SortAlgorithm {
        public int[] sort(int[] numbers);
    }

    public class QuickSortAlgorithm implements SortAlgorithm {

    public class BubbleSortAlgorithm implements SortAlgorithm {
```

Dependency Injection

- We use Spring Framework to instantiate beans and wire dependencies

```
ComplexAlgorithmImpl binarySearch =  
    new ComplexAlgorithmImpl(new QuickSortAlgorithm());
```

```
@Component  
public class ComplexAlgorithmImpl {  
  
    @Autowired  
    private SortAlgorithm sortAlgorithm;
```

Inversion of Control

```
public class ComplexAlgorithmImpl {  
  
    BubbleSortAlgorithm bubbleSortAlgorithm  
        = new BubbleSortAlgorithm();  
  
    @Component  
    public class ComplexAlgorithmImpl {  
  
        @Autowired  
        private SortAlgorithm sortAlgorithm;
```

IOC Container

- Find Beans
- Wire Dependencies
- Manage Lifecycle of the Bean

IOC Container

```
@Component
public class ComplexAlgorithmImpl {

    @Autowired
    private SortAlgorithm sortAlgorithm;

@Component
public class QuickSortAlgorithm implements SortAlgorithm {

public interface SortAlgorithm {
    public int[] sort(int[] numbers);
}
```

Application Context

- Bean Factory++
 - Spring's AOP features
 - I18n capabilities
 - `WebApplicationContext` for web applications etc

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:aop="http://www.springframework.org/schema/aop"
  xmlns:context="http://www.springframework.org/schema/context"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/aop
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context"
  >
```

```
ApplicationContext context =
    new ClassPathXmlApplicationContext(
        new String[] { "BusinessApplicationContext.xml",
            "Other-Configuration.xml" });
```

```
@Configuration  
class SpringContext {  
}
```

```
ApplicationContext ctx =  
    new AnnotationConfigApplicationContext(  
        SpringContext.class);
```

Unit Tests

```
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(classes = JavaTestContext.class)
public class DependencyInjectionJavaContextExamples {

    @RunWith(SpringJUnit4ClassRunner.class)
    @ContextConfiguration(locations = { "/TestContext.xml" })
    public class TodoBusinessTest {
```