SPRING FRAMEWORK

CONFIGURATION WITH ANNOTATIONS

APRIL, 2015

ANNOTATION CONFIG

DIY

```
<beans
    xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:context="http://www.springframework.org/schema/context"
    xsi:schemaLocation="
        http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/
                                    spring-beans-3.2.xsd
        http://www.springframework.org/schema/context
        http://www.springframework.org/schema/context/
                                    spring-context-3.2.xsd">

        <context:annotation-config/>

</beans>
```

AUTOWIRED

- **@Autowired**
- @Autowired (required=false)
- @Inject *-- JSR 330*

```
public class App {
   @Autowired
   private Client client;
}
```

AUTOWIRED

- **`@Autowired`**
- `@Autowired (required=false)`
- `@Inject  -- JSR 330`

```
public class App {
    @Autowired
    private Client client;
}
```

Can be used to inject:

- `BeanFactory`
- `ApplicationContext`
- `ResourceLoader`
- `ApplicationEventPublisher`

AUTOWIRED

```
public class App {

    @Autowired                // field
    private Client client;

    @Autowired                // setter
    public void setLogger(...) {
    }

    @Autowired                // constructor
    public App(Client client, ...) {
    }
}
```

QUALIFIER

- `@Qualifier("bean name")`
- `@Resource(name="...")  -- JSR 250`

```
public class App {
    @Autowired
    @Qualifier ("fileEventLogger")
    private EventLogger defaultLogger;
}
```

MORE ANNOTATIONS

INIT & DESTROY

```
public class SomeEventLogger {

  @PostConstruct
  public void init() {}

  @PreDestroy
  public void destroy() {}
}
```

MORE ANNOTATIONS

INIT & DESTROY

```
public class SomeEventLogger {

  @PostConstruct
  public void init() {}

  @PreDestroy
  public void destroy() {}
}
```

VALUES

```
@Value("Hello there!")

@Value("${client.id}")
```

BEANS

- **@Component**
- @Service
- @Repository
- @Controller

BEANS

- **@Component**
- @Service
- @Repository
- @Controller

```
<context:component-scan
                base-package="..."/>
```

BEANS

- **@Component**
- @Service
- @Repository
- @Controller

```
<context:component-scan
                base-package="..."/>
```

```
@Component("some name")
@Scope("prototype")
```

JAVA BASED CONFIGURATION

```
@Configuration
public class AppConfig {

    @Bean
    public Client client() {
        return new Client();
    }
}
```

JAVA BASED CONFIGURATION

```
public static void main (String[] args) {


 ApplicationContext ctx =
       new AnnotationConfigApplicationContext (
                     AppConfig.class);




       new AnnotationConfigApplicationContext (
             AppConfig.class, LoggersConfig.class);
```

JAVA BASED CONFIGURATION

REGISTER

```
ApplicationContext ctx =
        new AnnotationConfigApplicationContext();

ctx.register (AppConfig.class, OtherConf.class);

ctx.register (LoggersConfig.class);

ctx.refresh();
```

JAVA BASED CONFIGURATION

REGISTER

```
ApplicationContext ctx =
        new AnnotationConfigApplicationContext();

ctx.register (AppConfig.class, OtherConf.class);

ctx.register (LoggersConfig.class);

ctx.refresh();
```

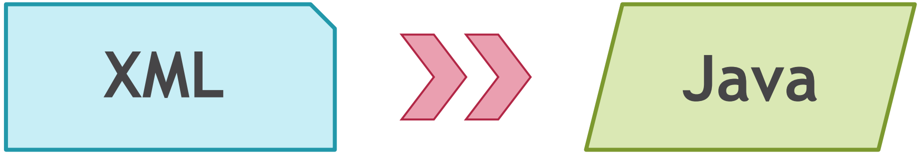
SCAN

```
ctx.scan ("ua.epam.spring.core.beans");

ctx.refresh();
```

PRACTICE

DIY


XML

Java

```
@Configuration
@Import (LoggersConfig.class)
public class AppConfig {
    @Bean
    @Scope ("prototype")
    public Event event() {
        return new Event(new Date());
    }

    @Autowired
    private EventLogger combinedLogger;
```

THANK YOU

YURIY_TKACH@EPAM.COM

APRIL, 2015