Spring Annotations

Spring Annotations

- Spring Framework provides support for Annotation Based Metadata to handle RAD.
- Developers may discard XML totally and take full advantage of Spring Annotations.

Spring Annotations

- In Annotation based configuration, there are further 2 options:
  - Java Based Configuration
  - Pure Annotation Based Configuration

Configuring Beans

Configuring Beans

- To configure beans, Spring provides 2 basic annotations:
  - `@Configuration`
  - `@Bean`

@Configuration

@Configuration

- Applied at the class level to introduce a class as a Configuration Unit.
- Classes annotated with `@Configuration` act as entry points of the spring configuration unit.

@Bean

@Bean

- Applied at the method level to indicate that a method is a Bean Creation Method.
- Objects returned by methods annotated with `@Bean` are treated as managed components in the Spring Environment.

Retrieving Beans

Retrieving Beans

- Beans registered in the annotation based configuration unit are obtained using a class `AnnotationConfigApplicationContext`.

AnnotationConfigApplicationContext

`AnnotationConfigApplicationContext`

- A class used to register the configuration specific class so that beans can be obtained against their identities.

Components

Components

- `@Bean` annotation can be used to configure beans in the configuration unit.
- However, developer needs to create these objects explicitly.

Components

- To enable Spring to create Java Objects using Reflection API, Spring provides a stereotype annotation `@Component`.

Components

- `@Component`

  Applied at the class level to mark that class as a Component class.

Scanning Components

Scanning Components

- Once a component is declared, it is to be scanned in the configuration unit and that is accomplished by an annotation `@ComponentScan`.