@Configuration: Used to indicate that a class declares one or more @Bean methods.

These classes are processed by the Spring container to generate bean definitions and

service requests for those beans at runtime.

Important Annotation

@Bean : Indicates that a method produces a bean to be managed by spring container.

Apply at method level.

@Bean(name ="", initMethod = "", destroyMethod = "")

@Configuration

public class AppConfig {

@Bean(name = "comp", initMethod = "turnOn", destroyMethod = "turnOff")

Computer computer(){

return new Computer();

}

}

@PostConstruct = initMethod

@PreDestroy = destroyMethod

@ComponentScan : we can specify the base

packages to scan spring components.

@PropertySource : Adding a property file to Spring.

@PropertySource("classpath:mysqldatabase.properties")

To get the value from properties file :

@Value("${databaseName}") // used only with properties file

private String databaseName;

mysqldatabase.properties file.

databaseName=school

disableStatementPooling=true

@Autowired : Is used for automatically inject

the bean.

@Qualifier("oracleDriver")

@component : Indicates that an annotated class is

a Component.

@Service :

@Repository

Spring MVC

@Controller

@RequestMapping

@PathVariable

@RequestParam

@RequestBody and @ResponseBody

@RequestHeader and @ResponseHeader

@Controller and @RestController.

@Controller is to create a model object and find a view.

The @RestController annotation in Spring MVC is nothing but a

combination of @Controller and @ResponseBody annotation.

It was added into Spring 4.0.

AOP (Aspect oriented programming)

Programmatic Approach

Annotation Approach

Aspect : A Service

Advice : Service Provide

point cut : A point or condition to execute aspects for business methods.

a. MBA

b. ARA / TA

c. MI

Adviser : Pointcut with advice combination.

Pointcut + Advice

Proxy : A weaver,

it will combine service and business

and return proxy object.

Target : Business Object