@Configuration is an analog for XML configuration file – it is configuration using Java class.

@Bean annotation works with @Configuration to create Spring beans.

 @Component annotation marks the Java class as a bean or say component so that the component-scanning mechanism of Spring can add into the application context.

 @Controller  annotation is used to indicate the class is a Spring controller. This annotation can be used to identify controllers for Spring MVC

@Service marks a Java class that performs some service, such as execute business logic, perform calculations and call external APIs.

@Repository annotation works as marker for any class that fulfills the role of repository or Data Access Object.This annotation is used on Java classes which directly access the database.

@GetMapping

This annotation is used for mapping HTTP GET requests onto specific handler methods. @GetMapping is a composed annotation that acts as a shortcut for @RequestMapping(method = RequestMethod.GET)

@PostMapping

This annotation is used for mapping HTTP POST requests onto specific handler methods. @PostMapping is a composed annotation that acts as a shortcut for @RequestMapping(method = RequestMethod.POST)

@PutMapping

This annotation is used for mapping HTTP PUT requests onto specific handler methods. @PutMapping is a composed annotation that acts as a shortcut for @RequestMapping(method = RequestMethod.PUT)

### @ComponentScan

This annotation is used with @Configuration annotation to allow Spring to know the packages to scan for annotated components. @ComponentScan is also used to specify base packages using basePackageClasses orbasePackage attributes to scan. If specific packages are not defined, scanning will occur from the package of the class that declares this annotation.

### @Lazy

This annotation is used on component classes. By default all autowired dependencies are created and configured at startup. But if you want to initialize a bean lazily, you can use @Lazy annotation over the class. This means that the bean will be created and initialized only when it is first requested for. You can also use this annotation on @Configuration classes. This indicates that all @Bean methods within that @Configuration should be lazily initialized.

### @Value

This annotation is used at the field, constructor parameter, and method parameter level. The @Value annotation indicates a default value expression for the field or parameter to initialize the property with. As the @Autowiredannotation tells Spring to inject object into another when it loads your application context, you can also use@Value annotation to inject values from a property file into a bean’s attribute. It supports both #{...} and${...} placeholders.

## Spring Framework Stereotype Annotations

### @Component

This annotation is used on classes to indicate a Spring component. The @Component annotation marks the Java class as a bean or say component so that the component-scanning mechanism of Spring can add into the application context.

### @Controller

The @Controller  annotation is used to indicate the class is a Spring controller. This annotation can be used to identify controllers for Spring MVC or Spring WebFlux.

### @Service

This annotation is used on a class. The @Service marks a Java class that performs some service, such as execute business logic, perform calculations and call external APIs. This annotation is a specialized form of the@Component annotation intended to be used in the service layer.

### @Repository

This annotation is used on Java classes which directly access the database. The @Repository annotation works as marker for any class that fulfills the role of repository or Data Access Object.

This annotation has a automatic translation feature. For example, when an exception occurs in the @Repositorythere is a handler for that exception and there is no need to add a try catch block.

@RequestParamannotation is used bind the parameter values from query string, the @RequestAttribute is used to access the objects which have been populated on the server side.

**Component vs Bean**

@Component used to auto-detect and auto-configure beans using classpath scanning. There’s an implicit one-to-one mapping between the annotated class and the bean (i.e. one bean per class).

@Bean is used to explicitly declare a single bean, rather than letting Spring do it automatically for us.

**@RequestParam** annotation to bind request parameters to a method parameter in your controller.

@RequestHeader that can be used to map controller parameter to request header value.