

## Spring Boot and Web Annotations

Use annotations to configure your web application.

**T** **@SpringBootApplication** - uses `@Configuration`, `@EnableAutoConfiguration` and `@ComponentScan`.

**T** **@EnableAutoConfiguration** - make Spring guess the configuration based on the classpath.

**T** **@Controller** - marks the class as web controller, capable of handling the requests.

**T** **@RestController** - a convenience annotation of a `@Controller` and `@ResponseBody`.

**M** **T** **@ResponseBody** - makes Spring bind method's return value to the web response body.

**M** **@RequestMapping** - specify on the method in the controller, to map a HTTP request to the URL to this method.

**P** **@RequestParam** - bind HTTP parameters into method arguments.

**P** **@PathVariable** - binds placeholder from the URI to the method parameter.

## Spring Framework Annotations

Spring uses dependency injection to configure and bind your application together.

**T** **@Configuration** - used to mark a class as a source of the bean definitions.

**T** **@ComponentScan** - makes Spring scan the packages configured with it for the `@Configuration` classes.

**T** **@Import** - loads additional configuration. This one works even when you specify the beans in an XML file.

**T** **@Component** - turns the class into a Spring bean at the auto-scan time.

**T** **@Service** - tells Spring that it's safe to manage `@Components` with more freedom than regular components.

**C** **F** **M** **@Autowired** - wires the application parts together, on the fields, constructors, or methods in a component.

**M** **@Bean** - specifies a returned bean to be managed by Spring context. The returned bean has the same name as the factory method.

**M** **@Lookup** - tells Spring to return an instance of the method's return type when we invoke it.

**T** **M** **@Primary** - gives higher preference to a bean when there are multiple beans of the same type.

**C** **F** **M** **@Required** - shows that the setter method must be configured to be dependency-injected with a value at configuration time.

**C** **F** **M** **@Value** - used to assign values into fields in Spring-managed beans. It's compatible with the constructor, setter, and field injection.

**T** **M** **@DependsOn** - makes Spring initialize other beans before the annotated one.

**T** **M** **@Lazy** - makes beans to initialize lazily. `@Lazy` annotation may be used on any class directly or indirectly annotated with `@Component` or on methods annotated with `@Bean`.

**T** **M** **@Scope** - used to define the scope of a `@Component` class or a `@Bean` definition and can be either singleton, prototype, request, session, globalSession, or custom scope.

**T** **@Profile** - adds beans to the application only when that profile is active.

**Legend:** **T** - Class **F** - Field Annotation **C** - Constructor Annotation **M** - Method **P** - Parameter