

## 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.

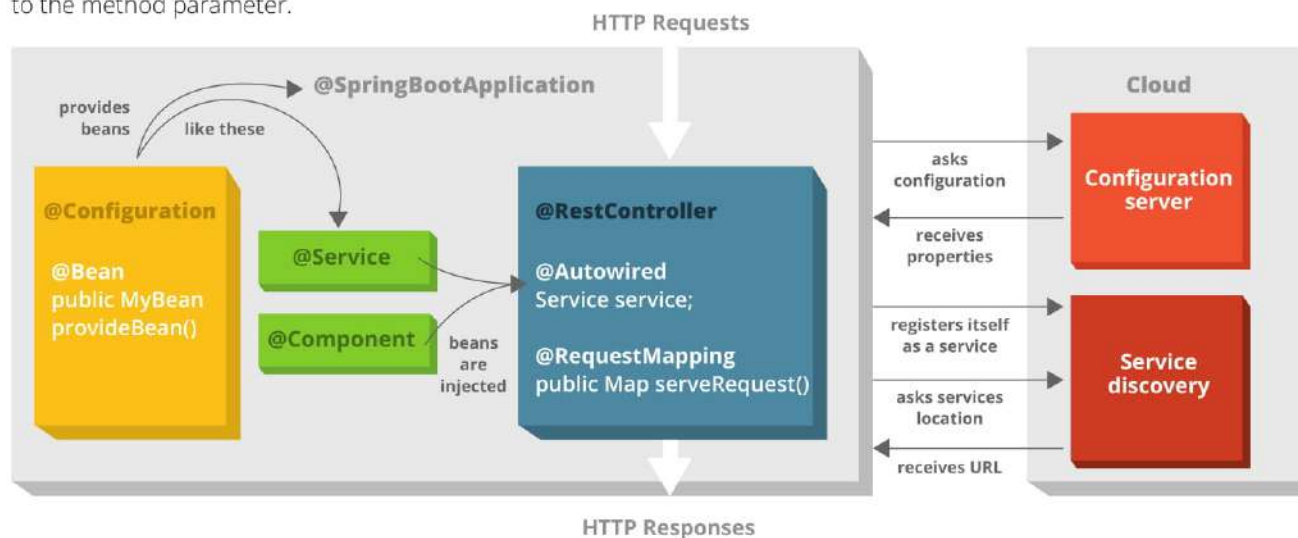
**T** **@Controller** - marks the class as web controller, capable of handling the requests. **T** **@RestController** - a convenience annotation of a **@Controller** and **@ResponseBody**.

**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 Cloud annotations

Make your application work well in the cloud.

**T** **@EnableConfigServer** - turns your application into a server other apps can get their configuration from.

Use **spring.application.cloud.config.uri** in the client **@SpringBootApplication** to point to the config server.

**T** **@EnableEurekaServer** - makes your app an Eureka discovery service, other apps can locate services through it.

**T** **@EnableDiscoveryClient** - makes your app register in the service discovery server and discover other services through it.

**T** **@EnableCircuitBreaker** - configures Hystrix circuit breaker protocols.

**M** **@HystrixCommand(fallbackMethod = "fallbackMethodName")** - marks methods to fall back to another method if they cannot succeed normally.

## Spring Framework annotations

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

**T** **@ComponentScan** - make Spring scan the package for the **@Configuration** classes.

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

**M** **@Bean** - indicates that a method produces a bean to be managed by the Spring container.

**T** **@Component** - turns the class into a Spring bean at the auto-scan time. **T** **@Service** - specialization of the **@Component**, has no encapsulated state.

**C F M** **@Autowired** - Spring's dependency injection wires an appropriate bean into the marked class member.

**T M** **@Lazy** - makes **@Bean** or **@Component** be initialized on demand rather than eagerly.

**C F M** **@Qualifier** - filters what beans should be used to **@Autowire** a field or parameter.

**C F M** **@Value** - indicates a default value expression for the field or parameter, typically something like `"#{systemProperties.myProp}"`

**C F M** **@Required** - fail the configuration, if the dependency cannot be injected.

## Legend

- T** - class
- F** - field annotation
- C** - constructor annotation
- M** - method
- P** - parameter