This page provides guidance on upgrading to Spring Framework 6.0.

Upgrading to Version 6.0

Core Container

The JSR-330 based `@Inject` annotation is to be found in `jakarta.inject` now. The corresponding JSR-250 based annotations `@PostConstruct` and `@PreDestroy` are to be found in `jakarta.annotation`. For the time being, Spring keeps detecting their `javax` equivalents as well, covering common use in pre-compiled binaries.

Data Access and Transactions

Due to the Jakarta EE migration, make sure to upgrade to Hibernate ORM 5.6.x with the `hibernate-core-jakarta` artifact, alongside switching your `javax.persistence` imports to `jakarta.persistence`. The corresponding Hibernate Validator generation is 7.0.x, based on `jakarta.validation`.

Web Applications

Due to the Jakarta EE migration, make sure to upgrade to Tomcat 10, Jetty 11, or Undertow 2.2.14 with the `undertow-servlet-jakarta` artifact, alongside switching your `javax.servlet` imports to `jakarta.servlet`.

Several outdated Servlet-based integrations have been dropped: e.g. Commons FileUpload and Tiles, as well as FreeMarker JSP support. We recommend `StandardServletMultipartResolver` for multipart file uploads and regular FreeMarker template views if needed, and a general focus on REST-oriented web architectures.

Spring MVC and Spring WebFlux no longer detect controllers based solely on a type-level `@RequestMapping` annotation. That means interfaced based AOP proxying for web controllers may no longer work. Please, enable class based proxying for such controllers or otherwise the interface must also have `@Controller`, see 22154.

`HttpMethod` is a class and no longer an enum. Though the public API has been maintained, some migration might be necessary (i.e. change from `EnumSet<HttpMethod>` to `Set<HttpMethod>`, use `if  else` instead of `switch`). For the rationale behind this decision, see 27697.

The Kotlin extension function to `WebTestClient.ResponseSpec::expectBody` now returns the Java `BodySpec` type, and no longer uses the workaround type `KotlinBodySpec`.Spring 6.0 uses Kotlin 1.6, which fixed the bug that needed this workaround (KT-5464). This means that `consumeWith` is not longer available.