

JEE6 Features

- [Introduction:](#)
- [Presentation:](#)
- [Dependency Injection and Data Validation:](#)
- [Component Model and Services:](#)
- [Persistence:](#)

Introduction:

The latest version of Java Platform, Enterprise Edition ([Java EE](#)) offers several new features including dependency injection, bean validation and significant enhancements in [EJB](#), [Servlets](#), [JSF](#), and [JSP](#) technologies. Sun Microsystems released the Java EE 6 version on Thursday. This article gives an overview of the new features in the latest release. We will follow-up with more detailed coverage of the features in subsequent news posts.

The three main goals for the Java EE 6 platform are flexible technology stack, enhanced extensibility, and ease of development.

Technology Stack:

Java EE 6 introduces the concept of profiles, configurations of the Java EE platform that are designed for specific classes of applications. A profile includes a subset of Java EE technologies, additional technologies that have gone through the Java Community Process (JCP), but are not part of the Java EE platform, or both. The new release includes the first of these profiles called [Web Profile](#) which is a subset of the Java EE platform designed for web application development. The Web Profile is the light-weight version of Java EE and includes only those technologies needed in most web applications, and does not include the enterprise technologies that these web applications typically don't need.

Enhanced Extensibility:

The new version also includes more extensibility points and service provider interfaces than the previous version. This allows the Java developers to plug in technologies and frameworks into Java EE 6 implementations in a standard way.

Ease of Development:

The developers can use Annotations to define web components such as servlets ([@WebServlet](#)) and servlet filters ([@WebFilter](#)). A new set of annotations for dependency injection has been standardized, making injectable classes much more portable across frameworks. Also, Java EE application packaging requirements have been simplified. For example, you can add an enterprise bean directly to a web archive (WAR) file. You no longer need to package an enterprise bean in a Java archive (JAR) file and then put the JAR file in an enterprise archive (EAR) file.

The new features in Java EE 6 are mainly divided into four categories: Presentation, Dependency Injection and Data Validation, Component Model and Services, and Persistence.

Presentation:

Some of the significant enhancements made in Java EE 6 are in the presentation (web) tier. Two key improvements in the area of extensibility are web fragments and shared framework pluggability which are provided by the new Servlet 3.0 ([JSR 315](#)) technology. Servlet 3.0 also offers support for asynchronous processing and annotations. With the asynchronous processing support, a servlet no longer has to wait for a response from a resource such as a database before its thread can continue processing. This enables long-lived client connections such as those in chat room applications.

Another Java EE 6 web tier technology is JSF 2.0 ([JSR 314](#)), the latest version of JSF technology. JSF 2.0 simplifies page and component authoring through Facelets, Templating, and Composite Components. It also provides the built-in support for Ajax technology and Annotations. JavaServer Pages 2.2 specification is also part of the new release.

Dependency Injection and Data Validation:

The Contexts and Dependency Injection for the Java EE Platform (CDI), JSR 299 specification, supplies a set of services to Java EE components. These services allow Java EE components, including EJBs and JSF components to be injected and to interact by firing and observing events. CDI also unifies and simplifies the EJB and JSF programming models to allow enterprise beans to replace JSF managed beans in a JSF application.

The new Bean Validation specification ([JSR 303](#)) makes validation simpler and reduces the duplication and errors. It provides a standard framework for validation, in which the same set of validation rules can be shared by all the layers of an application (web, controller, service, and domain layers).

Component Model and Services:

On the server side, Java EE 6 release includes EJB version 3.1 ([JSR 318](#)) and Java API for RESTful Web Services (JAX-RS) specification ([JSR 311](#)). Some improvements made in EJB 3.1 are:

- **No-interface view:** This simplifies the creation of an enterprise bean using only a bean class without having to write a separate business interface.
- **Singletons:** The new Singleton design, using the [@Singleton](#) annotation, lets you easily share state between multiple instances of an enterprise bean component or between multiple enterprise bean components in an application.

- **Asynchronous session bean invocation:** Using the [@Asynchronous](#) annotation, the session bean methods can now be invoked asynchronously.
- **Simplified Packaging:** The new packaging mechanism removes the restriction that enterprise bean classes must be packaged in an ejb-jar file. The EJB classes can now be directly placed in a WAR file.
- **EJB Lite:** This model is a subset of EJB 3.1 for inclusion in the Java EE profiles.

JAX-RS specification enables the development of lightweight web services that conform to the [REST](#) style of software architecture. It provides the API for building RESTful web services in Java with a set of annotations and associated classes and interfaces.

Persistence:

The persistence module includes JPA version 2.0 ([JSR 317](#)) which has several enhancements to JPQL and offers a new Criteria API which can be used for dynamically constructing object-based queries. There is also a JPA 2.0 metamodel API that enables the developers to dynamically access the database metamodel information. JPA 2.0 adds two annotations in support of the new collection mappings: [@ElementCollection](#) (used to specify that the basic or embeddable objects in the collection are stored in a separate table called a collection table) and [@CollectionTable](#) (used to specify details about the collection table, such as its columns).

The new Java EE 6 [SDK](#) can be [downloaded](#) from Java website. For more InfoQ coverage on Java EE 6 release and the new features, check out [this link](#).