**Java EE 7: Key features, Specifications, Projects, Mailing List Archives**

**Java EE 7 (**[**JSR 342**](http://jcp.org/en/jsr/detail?id=342)**)**

* The main theme is to easily run applications on private or public clouds
* The platform will define application metadata descriptor to describe PaaS execution environment such as multi-tenancy, resources sharing, quality-of-service, and dependencies between applications
* Embrace latest standards like HTML5, WebSocket, JSON and have a standards-based API for each one of them
* Remove inconsistencies between Managed Beans, EJB, Servlets, JSF, CDI, and JAX-RS
* Possible inclusion of JAX-RS 2.0 in the Web Profile, revised JMS 2.0 API
* Technology Refresh for several existing technologies (more on this below) and possible inclusion of Concurrency Utilities for Java EE (JSR 236) and JCache (JSR 107)
* Status
  + Approved by the JCP
  + Spec lead: Linda DeMichiel, Bill Shannon (Oracle)
  + Project: [javaee-spec.java.net](http://java.net/projects/javaee-spec/pages/Home)
  + Mailing Lists Archive: [jsr342-expert@javaee-spec.java.net](http://java.net/projects/javaee-spec/lists/jsr342-experts/archive), [users@javaee-spec.java.net](http://java.net/projects/javaee-spec/lists/users/archive)

**JPA 2.1 (**[**JSR 338**](http://jcp.org/en/jsr/detail?id=338)**)**

* Support for multi-tenancy
* Support for stored procedures and vendor function
* Update and Delete Critieria queries
* Support for schema generation
* Persistence Context synchronization
* CDI injection into listeners
* Status
  + Approved by the JCP
  + Spec lead: Linda DeMichiel (Oracle)
  + Project: [jpa-spec.java.net](http://java.net/projects/jpa-spec/)
  + Mailing Lists Archive: [jsr338-experts@jpa-spec.java.net](http://java.net/projects/jpa-spec/lists/jsr338-experts/archive), [users@jpa-sepc.java.net](http://java.net/projects/jpa-spec/lists/users/archive).

**JAX-RS 2.0 (**[**JSR 339**](http://jcp.org/en/jsr/detail?id=339)**)**

* Client API - low level using builder pattern and possibly a higher level on top of that
* Hypermedia - easily create and process links associated with resources
* Form or Query parameter validation using Bean Validation
* Closer integration with @Inject, etc
* Server-side asynchronous request processing
* Server-side content negotiation using "qs"
* Status
  + Approved by the JCP, Early Draft [Available](http://jcp.org/aboutJava/communityprocess/edr/jsr339/index.html), [Draft Javadocs](http://jax-rs-spec.java.net/nonav/2.0/apidocs/index.html)
  + Spec lead: Santiago Pericas-Geersten, Marek Potociar (Oracle)
  + Project: [jaxrs-spec.java.net](http://jax-rs-spec.java.net/)
  + Mailing List Archive: [jsr339-experts@jax-rs-spec.java.net](http://java.net/projects/jax-rs-spec/lists/jsr339-experts/archive), [users@jax-rs-spec.java.net](http://java.net/projects/jax-rs-spec/lists/users/archive)

**Servlets 3.1 (**[**JSR 340**](http://jcp.org/en/jsr/detail?id=340)**)**

* Optimize the PaaS model for Web applications
* Multi tenancy for security, session, resources, etc.
* Asynchronous IO based on NIO2
* Simplfiied asynchronous Servlets
* Utilize Java EE concurrency utilities
* Enable support for WebSockets
* Status:
  + Approved by the JCP
  + Spec lead: Shing-Wai Chan, Rajiv Mordani (Oracle)
  + Project: [servlet-spec.java.net](http://java.net/projects/servlet-spec/)
  + Mailing List Archive: [jsr340-experts@servlet-spec.java.net](http://java.net/projects/servlet-spec/lists/jsr340-experts/archive), [users@servlet-spec.java.net](http://java.net/projects/servlet-spec/lists/users/archive)

**Expression Language 3.0 (**[**JSR 341**](http://jcp.org/en/jsr/detail?id=341)**)**

* Separate ELContext into parsing and evaluation contexts
* Customizable EL coercion rules
* Reference static methods and members directly in EL expressions
* Adding operators like equality, string concatenation, and sizeof etc.
* Integration with CDI such as generating events before/during/after the expressions are evaluated
* Status
  + Approved by the JCP
  + Spec lead: Kin-man Chung (Oracle)
  + Project: [el-spec.java.net](http://java.net/projects/el-spec/)
  + Mailing List Archive: [jsr-341-experts@el-spec.java.net](http://java.net/projects/el-spec/lists/jsr341-experts/archive), [users@el-spec.java.net](http://java.net/projects/el-spec/lists/users/archive)

**Java Message Server 2.0 (**[**JSR 343**](http://jcp.org/en/jsr/detail?id=343)**)**

* Ease of development - changes to the JMS programming model to make the application development simpler and easier
* Remove/Clarify ambiguities in the existing specification
* Integration with CDI
* Clarification of the relationship between JMS and other Java EE specs
* A new mandatory API to allow any JMS provider to be integrated with any Java EE container
* Multi-tenancy and other cloud-related features from the platform
* Status
  + Approved by the JCP
  + Spec lead: Nigel Deakin (Oracle)
  + Project: [jms-spec.java.net](http://jms-spec.java.net)
  + Mailing List Archive: [jsr-343-experts@jms-spec.java.net](http://java.net/projects/jms-spec/lists/jsr343-experts/archive), [users@jms-spec.java.net](http://java.net/projects/jms-spec/lists/users/archive)

**Java Server Faces 2.2 (**[**JSR 344**](http://jcp.org/en/jsr/detail?id=344)**)**

* Ease of Development - making configuration options dynamic, make cc:interface in composite components optional, shorthand URLs for Facelet tag libraries, integration with CDI, OSGi support for JSF artifacts
* Support implementation of Portlet Bridge 2.0 (JSR 329)
* Support for HTML5 features like HTML5 Forms, Metadata, Heading and Section content model
* Flow management, Listener for page navigation events, and new components like FileUpload and BackButton
* Status
  + Approved by the JCP, Early Draft [Available](http://jcp.org/aboutJava/communityprocess/edr/jsr344/)
  + Spec lead: Ed Burns (Oracle)
  + Project: [javaserverfaces-spec-public.java.net](http://javaserverfaces-spec-public.java.net/)
  + Mailing List Archive: [jsr344-experts@javaserverfaces-spec-public.java.net](http://java.net/projects/javaserverfaces-spec-public/lists/jsr344-experts/archive), [users@javaserverfaces-spec-public.java.net](http://java.net/projects/javaserverfaces-spec-public/lists/users/archive)

**EJB 3.2 (**[**JSR 345**](http://www.jcp.org/en/jsr/detail?id=345)**)**

* Enhancements to the EJB architecture to enable PaaS, such as multi-tenancy
* Factorization of container-managed transactions to use outside EJB
* Further use of annotations
* Alilgnment and integration with other specifications in the platform
* Status
  + Approved by the JCP
  + Spec lead: Marina Vatkina (Oracle)
  + Project: [ejb-spec.java.net](http://java.net/projects/ejb-spec/)
  + Mailing List Archive: [jsr-345-experts@ejb-spec.java.net](http://java.net/projects/ejb-spec/lists/jsr345-experts/archive), [users@ejb-spec.java.net](http://java.net/projects/ejb-spec/lists/users/archive)

**CDI 1.1 (**[**JSR 346**](http://jcp.org/en/jsr/summary?id=346)**,** [**more details**](http://in.relation.to/Bloggers/CDI11)**)**

* Global ordering of interceptors and decorators
* API for managing built-in contexts
* Embedded mode to allow startup outside Java EE container
* Declarative control over which packages/beans are scanned in an archive
* Injection for static members such as loggers
* Send Servlet events as CDI event
* Status
  + Approved by the JCP, Early Draft [Available](http://jcp.org/aboutJava/communityprocess/edr/jsr346/index.html)
  + Spec lead: Pete Muir (RedHat)
  + Project: [CDI 1.1 Development](https://github.com/jboss/cdi/wiki)
  + Mailing List Archive: [cdi-dev](http://lists.jboss.org/pipermail/cdi-dev/), [@jsr346](https://twitter.com/#%21/jsr346)

**Bean Validation 1.1 (**[**JSR 349**](http://jcp.org/en/jsr/detail?id=349)**)**

* Integration with other Java EE specs
  + JAX-RS: Validate parameters and return values on HTTP calls
  + JAXB: Convert constraints into XML schema descriptor
* Method level validation
* Apply constraints on group collection
* Extend the model to support AND and OR style composition
* Status
  + Approved by the JCP
  + Spec lead: Emmanuel Bernard (RedHat)
  + Project: [beanvalidation.org](http://beanvalidation.org/)
  + Mailing List Archive: [beanvalidation-dev](http://lists.jboss.org/pipermail/beanvalidation-dev/)

**JCache (**[**JSR 107**](http://jcp.org/en/jsr/detail?id=107)**)**

* API and semantics for temporary, in-memory caching of Java objects, including object creation, shared access, spooling, invalidation, and consistency across JVMs
* Package: javax.cache
* Status
  + Approved by the JCP
  + Spec lead: Yannis Cosmadopoulos, Cameron Purdy (Oracle) and Gregory Luck (Software AG)
  + Project page: [jsr107spec](https://github.com/jsr107/jsr107spec)
  + Mailing List Archive: [jsr107@googlegroups.com](http://groups.google.com/group/jsr107)

**State Management (**[**JSR 350**](http://jcp.org/en/jsr/detail?id=350)**)**

* API that can be used by applications and Java EE containers to offload the responsibility of statement management into third party providers with different QoS characteristics
* Java SE-based callers can access the state data by querying the state providers
* Providers with different QoS can be added and API callers can query to meet their criteria
* Package: javax.state and javax.state.provider
* Status
  + Approved by the JCP
  + Spec lead: Mitch Upton (Oracle)
  + Project page: [java-state-management.java.net](http://java.net/projects/java-state-managemen/)
  + Mailing List Archive: [jsr-350-experts@java-state-management.java.net](http://java.net/projects/java-state-managemen/lists/jsr-350-experts/archive), [jsr-350-users@java-state-management.java.net](http://java.net/projects/java-state-managemen/lists/jsr-350-users/archive)

**Batch Application for the Java Platform (**[**JSR 352**](http://jcp.org/en/jsr/detail?id=352)**)**

* Programming model for batch applications and a runtime for scheduling and executing jobs
* Defines Batch Job, Batch Job Step, Batch Application, Batch Executor, and Batch Job Manager for the standard programming model
* Package: javax.batch
* Status
  + Approved by the JCP
  + Spec lead: Chris Vignola (IBM)
  + Project page: [jbatch.java.net](http://java.net/projects/jbatch/)
  + Mailing List Archive: [issues@jbatch.java.net](http://java.net/projects/jbatch/lists/issues/archive)

**Concurrency Utilities for Java EE (**[**JSR 236**](http://jcp.org/en/jsr/detail?id=236)**)**

* Provides a clean, simple, independent API by building on JSR 166, making it appropriate for use within any Java EE contianer.
* Package: javax.util.concurrent
* Status
  + Approved by the JCP
  + Spec lead: Anthony Lai, Naresh Revanuru (Oracle)
  + Project page:
  + Mailing List Archive:

**Java API for JSON Processing (**[**JSR 353**](http://jcp.org/en/jsr/detail?id=353)**)**

* Produce and consume JSON text in a streaming fashion(similar to StAX API for XML)
* Build a Java object model for JSON text using API classes(similar to DOM API for XML)
* Status
  + Under JCP Review
  + Spec lead: Jitendra Kotamraju
  + Project Page: [json-processing-spec](http://java.net/projects/json-processing-spec)
  + Mailing List Archive: [jsr353-experts@json-processing-spec.java.net](http://java.net/projects/json-processing-spec/lists/jsr353-experts/archive), [users@json-processing-spec.java.net](http://java.net/projects/json-processing-spec/lists/users/archive)

The Expert Groups for most of the JSRs have already been formed but you can still participate by joining the publicly visible aliases and reviewing the drafts. All the JSRs following JCP 2.8 are run more transparently and some of the highlights on that front are:

* Names of the EG members are publicly visible
* EG business reported on publicly readable alias
* Schedule is public, current and updated regularly
* Public can read/write to a wiki to discuss the status so far
* Discussion board on jcp.org
* Public read-only issue tracker