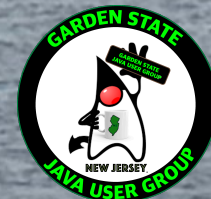
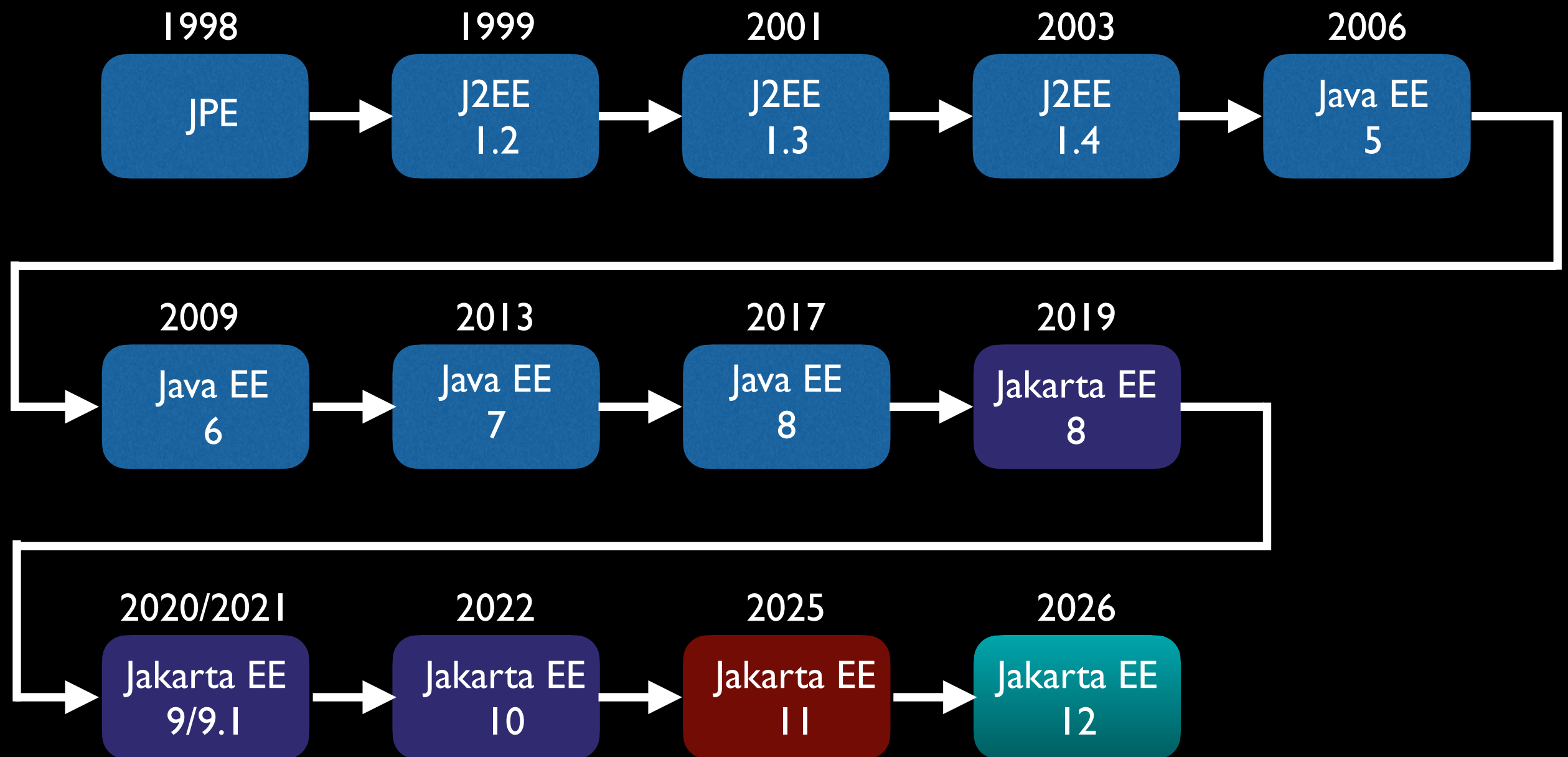


Getting Started with Jakarta EE 11 and What's New in Jakarta EE 12

Michael P. Redlich | @mpredli



Evolution of Java EE/Jakarta EE





Hi, I'm Mike



Java Champion



Co-Director

InfoQ^{ueue}

Lead Java Queue
News Editor



Contract Developer Advocate
Technical Writer



JAKARTA^{EE}
Committer:
Jakarta Data
Jakarta NoSQL



Vice President

ExxonMobil

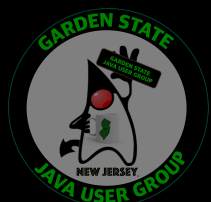


Helidon Revealed

A Practical Guide to Oracle's Microservices
Framework
Michael Redlich



Author



InfoQ



Poll

- I'm a Jakarta EE Expert!
- I've built an application using Jakarta EE
- I'm familiar with Jakarta EE, but haven't used it
- I've heard some things about Jakarta EE
- What the heck is Jakarta EE?



Jakarta EE Profiles



Jakarta EE Profiles

Platform

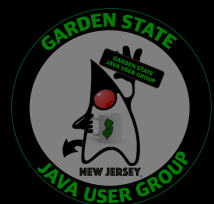
Web Profile

Core Profile



Jakarta EE Platform

- Defines a standard platform for hosting all Jakarta EE specifications
- Designed for developers who require the full set of Jakarta EE specifications for developing enterprise applications



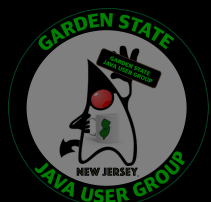
Jakarta EE 11 Platform

	Authentication 3.1	Data 1.0	
Authorization 3.0	Concurrency 3.1	Persistence 3.2	RESTful Web Services 4.0
Activation 2.1	CDI 4.1	Pages 4.0	JSON Processing 2.1
Batch 2.1	Expression Language 6.0	WebSocket 2.2	JSON Binding 3.0
Connectors 2.1	Faces 4.1	Validation 3.1	Annotations 3.0
Mail 2.1	Security 4.0	Debugging Support 2.0	Interceptors 2.2
Messaging 3.1	Servlet 6.1	Enterprise Beans Lite 4.0	Dependency Injection 2.0
Enterprise Beans 4.0	Standard Tag Libraries 3.0	Transactions 2.0	CDI Lite 4.1

Updated

Not Updated

New

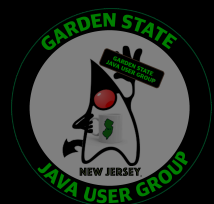


InfoQ



Jakarta EE Web Profile

- Defines a subset of the Jakarta EE Platform that contains web technologies specifically targeted for developing web applications



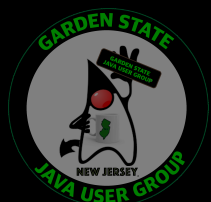
Jakarta EE 11 Web Profile

Authentication 3.1	Data 1.0	
Concurrency 3.1	Persistence 3.2	RESTful Web Services 4.0
CDI 4.1	Pages 4.0	JSON Processing 2.1
Expression Language 6.0	WebSocket 2.2	JSON Binding 3.0
Faces 4.1	Validation 3.1	Annotations 3.0
Security 4.0	Debugging Support 2.0	Interceptors 2.2
Servlet 6.1	Enterprise Beans Lite 4.0	Dependency Injection 2.0
Standard Tag Libraries 3.0	Transactions 2.0	CDI Lite 4.1

Updated

Not Updated

New

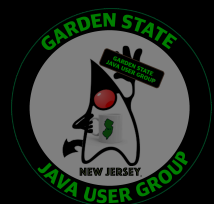


InfoQ



Jakarta EE Core Profile

- Defines a subset of the Jakarta EE Platform specifications targeting smaller runtimes suitable for microservices and ahead-of-time compilation
- Focused on providing a minimal basis for cloud native runtimes, including runtimes that support build time applications
- Introduced in Jakarta EE 10



Jakarta EE 11 Core Profile

RESTful Web Services 4.0

JSON Processing 2.1

JSON Binding 3.0

Annotations 3.0

Interceptors 2.2

Dependency Injection 2.0

CDI Lite 4.1

Updated

Not Updated

New



The Road to Jakarta EE 11



InfoQ



Jakarta EE 11 Release Timeline (i)



Milestone 1
December 2023



Milestone 2
March 2024



Milestone 3
April 2024



Milestone 4
May 2024

Jakarta EE 11 Release Timeline (ii)



Core Profile
December 2024



Web Profile
April 2025



Platform
June 2025

What's New in Jakarta EE 11

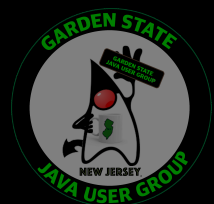


InfoQ



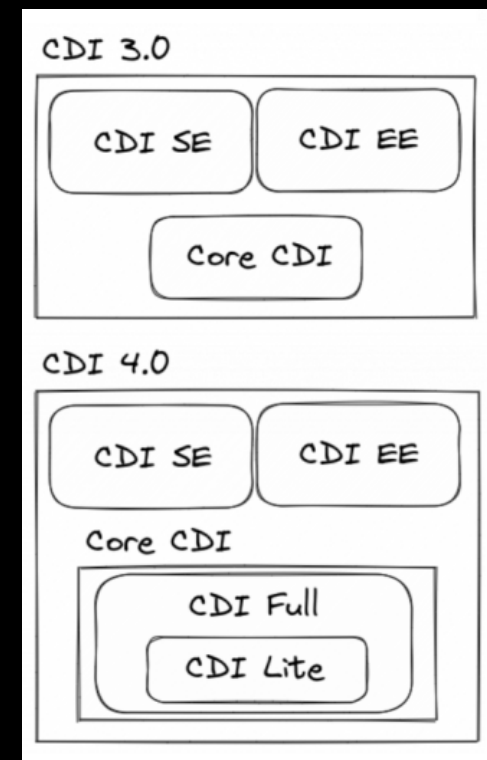
What's New (i)

- One new specification:
 - Jakarta Data
- 16 specifications have been updated and have passed their respective reviews



What's New (ii)

- CDI Lite
- Optional Features
 - Jakarta SOAP with Attachments
 - Jakarta XML Web Services
 - Jakarta XML Binding



What's New (iii)

- Support for Java Records
 - Jakarta Expression Language 6.0
 - Jakarta Persistence 3.2
 - Jakarta Validation 3.1
- Support for Virtual Threads
 - Jakarta Concurrency 3.1



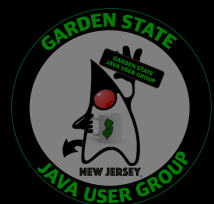
What's New (iv)

- Removal of legacy `@ManagedBean` annotation
- References to `SecurityManager` class have been removed
- Initial work on interoperability between MicroProfile JWT Authentication and Jakarta Security specifications



Updated Specifications (i)

- Jakarta Annotations 3.0
- Jakarta Authentication 3.1
- Jakarta Authorization 3.0
- Jakarta Concurrency 3.1
- Jakarta Contexts and Dependency Injection 4.1
- Jakarta Data 1.0
- Jakarta Expression Language 6.0
- Jakarta Faces 4.1*
- Jakarta Interceptors 2.2
- Jakarta Pages 4.0



Updated Specifications (ii)

- Jakarta Persistence 3.2
- Jakarta RESTful Web Services 4.0
- Jakarta Security 4.0
- Jakarta Servlet 6.1
- Jakarta Validation 3.1
- Jakarta WebSocket 2.2





Getting Started with Jakarta EE 11



InfoQ



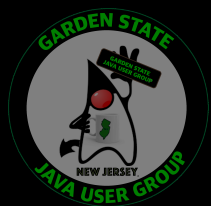


InfoQ



Payara Cloud

- Next generation of fully-managed cloud native application runtime
- Offers a flexible and powerful way to easily run Jakarta EE apps on the cloud without Kubernetes
- Free 15-day trial



Payara Qube at a Glance

Package your Java app

Build your Jakarta EE, Spring, or Quarkus application

Upload to Payara Qube

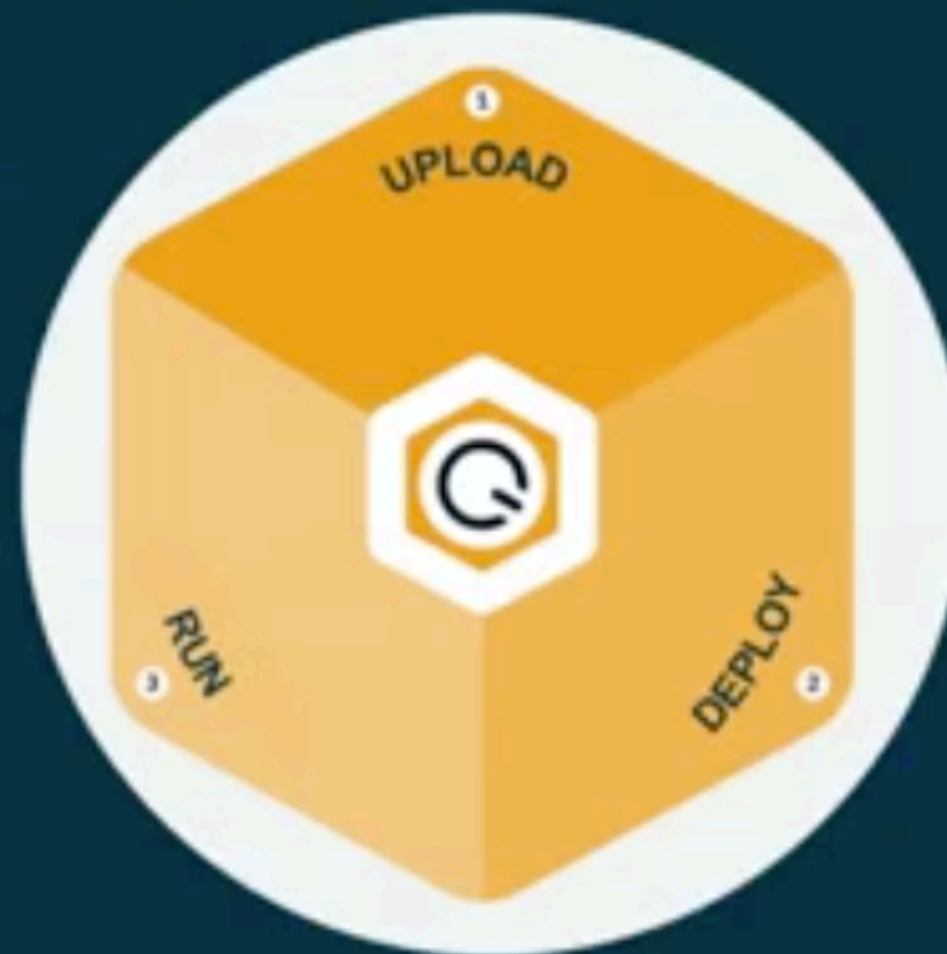
Application automatically containerized

Built-in metrics and monitoring dashboards

Track the performance and health of applications (Kubernetes & Log metrics)

Application and Access Management

Web-based management console to configure and manage namespaces and applications



One-click deployment

Automated deployment across any environment

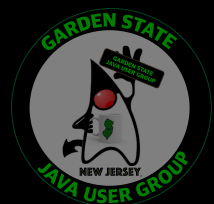


<https://www.payara.fish/products/cloud-native-java-deployment-platform/payara-qube/>



Jakarta EE 11 Demos

- JDK 21
- Jakarta EE 11
- Payara Micro 7.2025.1.Alpha2
- <https://github.com/mpredli01/jakartaee-11/blob/main/README.adoc>





Jakarta EE 12



What's New? (so far)

- Three new specifications:
 - Jakarta Query
 - Jakarta Portlet
 - Jakarta Portlet Bridge
- References to `SecurityManager` class will be removed



Considerations

- Deprecations:
 - Application Client Container
 - JNDI
- Support for HTTP/3
- Continued effort to be more CDI-centric
- Additional support for virtual threads



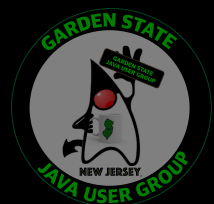
Current Status

- Updates planned for 24 specifications
- A plan to move the Jakarta Authentication and Jakarta Authorization under the auspices of Jakarta Security



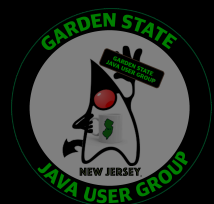
Planned Specification Updates (i)

- Jakarta Activation 2.2
- Jakarta Annotations 3.1
- Jakarta Batch 2.2
- Jakarta Concurrency 3.2
- Jakarta Connectors 2.2
- Jakarta Contexts and Dependency Injection 5.0
- Jakarta Data 1.1
- Jakarta Enterprise Beans 4.1
- Jakarta Expression Language 6.1



Planned Specification Updates (ii)

- Jakarta Faces 5.0
- Jakarta JSON Binding 3.1
- Jakarta JSON Processing 2.2
- Jakarta Mail 2.2
- Jakarta MVC 3.1
- Jakarta NoSQL 1.1
- Jakarta Pages 4.1
- Jakarta Persistence 4.0
- Jakarta RESTful Web Services 5.0
- Jakarta Security 5.0
- Jakarta Servlet 6.2



Planned Specification Updates (iii)

- Jakarta Standard Tag Library 3.1
- Jakarta Transactions 2.1
- Jakarta Validation 4.0
- Jakarta WebSocket 2.3



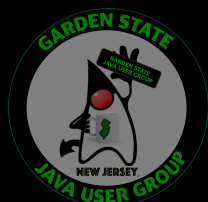
MicroProfile to Jakarta EE

- MicroProfile JWT Authentication to Jakarta Security
- MicroProfile REST Client to Jakarta RESTful Web Services
- MicroProfile Config to Jakarta Config



Status of Jakarta EE Specifications

Jakarta Activation 2.2	Jakarta Annotations 3.1	Jakarta Authentication 3.1
Jakarta Authorization 3.0	Jakarta Batch 2.2	Jakarta Concurrency 3.2
Jakarta Config 1.0	Jakarta Connectors 2.2	Jakarta CDI 5.0
Jakarta Data 1.1	Jakarta Debugging 2.0	Jakarta DI 2.0
Jakarta Deployment 1.7	Jakarta Enterprise Beans 4.1	Jakarta Enterprise WS 2.0
Jakarta EL 6.1	Jakarta Faces 5.0	Jakarta Interceptors 2.2
Jakarta JSON Binding 3.1	Jakarta JSON Processing 2.2	Jakarta Mail 2.2
Jakarta Managed Beans 2.0	Jakarta Management 1.1	Jakarta Messaging 3.1
Jakarta MVC 3.1	Jakarta NoSQL 1.1	Jakarta Pages 4.1
Jakarta Persistence 4.0	Jakarta Portlet 4.0	Jakarta Portlet Bridge 7.0
Jakarta Query 1.0	Jakarta RESTful WS 5.0	Jakarta RPC
Jakarta Security 5.0	Jakarta Servlet 6.2	Jakarta SOAP 3.0
Jakarta Standard Tags 3.1	Jakarta Transactions 2.1	Jakarta Validation 4.0
Jakarta WS Metadata 3.0	Jakarta WebSocket 2.3	Jakarta XML Binding 4.0
Jakarta XML Registries 1.0	Jakarta XML RPC 1.1	Jakarta XML WS 4.0
Planned Updates	No Planned Updates	New



Jakarta EE 12 Road Map



InfoQ



Jakarta EE 12 Draft Release Timeline (i)



Milestone 1
Q3 2025



Milestone 2
Q3 2025



Milestone 3
Q4 2025



Milestone 4
Q4 2025



InfoQ



Jakarta EE 12 Draft Release Timeline (ii)



Core Profile
June/July 2026



Web Profile
June/July 2026



Platform
June/July 2026

Jakarta Query



What is Jakarta Query?

- Defines an object-oriented query language designed for use with Jakarta Data, Jakarta Persistence, Jakarta NoSQL and other persistence solutions written in a JVM language.

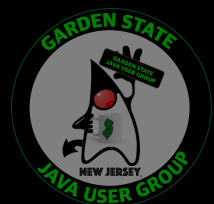


Jakarta Portlet



What is Jakarta Portlet?

- Enables the development of modular, reusable web components known as portlets
- A migration from JSR 362, Portlet Specification 3.0, under Java EE 8

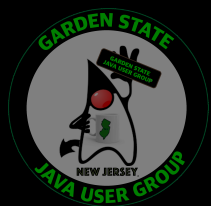


Jakarta Portlet Bridge



What is Jakarta Portlet Bridge?

- Defines the necessary behavior and APIs for Jakarta Faces applications to operate within a portlet environment
 - provides a bridge that aligns the Jakarta Faces lifecycle with the Jakarta Portlet lifecycle.
- A migration from JSR 378, Portlet 3.0 Bridge for JavaServer Faces 2.2 Specification, under Java EE 8



Jakarta EE Future Directions Group



InfoQ



What is Jakarta Future Directions Group?

- Formed in July 2024
- Bring together people from inside and outside of Java-related working groups, influencers, and other interested parties focusing on enterprise and cloud-native Java

<https://projects.eclipse.org/interest-groups/jakarta-ee-future-directions>



Roles

- Ideation
- Research
- Presentation of recommendations
- Work closely with the Jakarta EE and MicroProfile working groups
 - to ensure the ideas are scoped, prioritised and understood



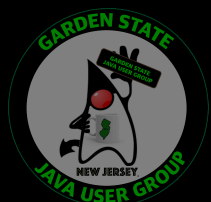
Recommendations

- Non-binding on the working groups
- Intended to provide insights and ideas that working groups may not have time to consider



Participating Members

- Garden State JUG
- IBM
- iJUG
- London Java Community
- Microsoft
- OmniFish
- Open Elements
- Oracle
- Payara

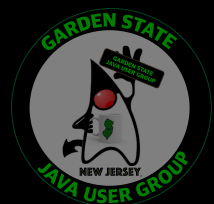


InfoQ



Resources (i)

- Jakarta EE Specifications
- #Hashtag Jakarta EE
 - Ivar Grimstad
- Boost Performance And Developer Productivity With Jakarta EE I I
 - Arjan Tijms



Resources (ii)

- Jakarta EE I I Example Applications
 - Michael Redlich
- Securing an Application with JWT
 - Luis Neto



Want to Get Involved?

- Contribute to open source!



- Share
- Embrace
- Engage
- Develop

Tanja Obradovic
Jakarta EE Program Manager at
Eclipse Foundation

- Share your own Jakarta EE experiences!

Acknowledgements

- Maximillian Arruda
- Sergey Beryozkin
- David Blevins
- Ivar Grimstad
- Ondro Mihályi
- Steve Millidge
- Luis Neto
- Tanja Obradovic
- Reza Rahman
- Otávio Santana
- Arjan Tijms
- Adam Yoho



Contact Info

`mike@redlich.net`

`@mpredli`

`redlich.net`

`redlich.net/portfolio`

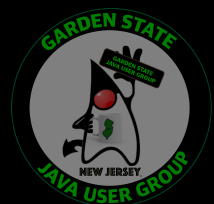
`github.com/mpredli01`



Thanks!



Overview of Updated Jakarta EE 11 Specifications



Jakarta Annotations



What is Jakarta Annotations?

- Defines a collection of annotations representing common semantic concepts that enable a declarative style of programming that applies across a variety of Java technologies



What's New?

- Removal of the deprecated `@ManagedBeans` annotation



Compatible Implementations

- Current version: 3.0
- Compatible implementations:
 - GlassFish 8.0.0-M3



Jakarta Authentication



What is Jakarta Authentication?

- Defines a general low-level SPI for authentication modules:
 - controllers that interact with a caller and the environment of a container to acquire the caller's credentials, validate them, and pass an authenticated identity to the container



What's New?

- Minor updates to support the overall goals of Jakarta Security
- Clarifications around state and concurrency
- References to `SecurityManager` class have been removed



Compatible Implementations

- Current version: 3.1
- Compatible implementations:
 - GlassFish 8.0.0-M6
 - Eclipse Epicyro 3.1.0

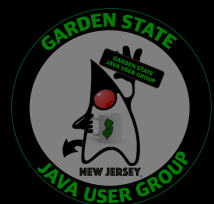


Jakarta Authorization



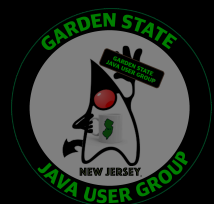
What is Jakarta Authorization?

- Defines a general low-level SPI for authorization modules:
 - repositories of permissions facilitating subject-based security by determining whether a given subject has a defined permission
 - algorithms to transform security constraints for specific containers (such as Jakarta Servlet or Jakarta Enterprise Beans) into these permissions



What's New? (i)

- Future-proof the specification and make it more suitable for cloud deployments
 - achieved by adding an API for programmatically registering policy providers to create a replacement for the deprecated `Policy` class
- Standardize the context ID for Servlet containers



What's New? (ii)

- Several new convenience methods to make the API easier to use
- References to `SecurityManager` class have been removed



Compatible Implementations

- Current version: 3.0
- Compatible implementations:
 - GlassFish 8.0.0-M5
 - Eclipse Exousia 3.0.0-M3

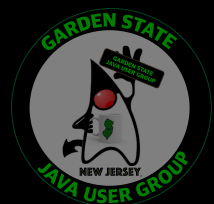


Jakarta Concurrency



What is Jakarta Concurrency?

- Provides a mechanism for using concurrency from application components without compromising container integrity while still preserving the fundamental benefits of the Jakarta EE platform



What's New? (i)

- Integration with Java 21 Virtual Threads
- Support for the Java Flow class and the MicroProfile Reactive Streams Operators and MicroProfile Context Propagation specifications



What's New? (ii)

- Replace more features from EJB
 - such as the `@Schedule` annotation
- Become more CDI-centric
 - by adding the ability to inject default resource implementations of the `ManagedExecutorService` and `ContextService` interfaces



Compatible Implementations

- Current version: 3.1
- Compatible implementations:
 - Open Liberty 24.0.0.6-beta



Jakarta Contexts and Dependency Injection (CDI)



What is Jakarta CDI?

- Specifies a means for obtaining objects in such a way to maximize reusability, testability and maintainability as compared to traditional approaches such as constructors, factories, and service locators like the Java Naming and Directory Interface (JNDI)



What's New?

- Add methods to **BeanConfigurator** interface for applying decorators
- Support for **@Priority** annotation on producers



Compatible Implementations

- Current version: 4.1
- Compatible implementations:
 - Weld 6.0.0.Beta1



Jakarta Data



What is Jakarta Data?

- Provides an API that allows easy access to database technologies
- Can split the persistence from the model with several features, such as the ability to compose custom query methods on a `Repository` interface where the framework will implement it



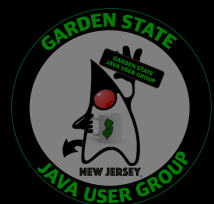
What's New? (i)

- Support for CDI extensions
- New `BasicRepository` interface
- New `insert()` and `update()` methods in the `CrudRepository` interface
- New `@Insert`, `@Update`, `@Delete` and `@Save` annotations



What's New? (ii)

- Jakarta Data Query Language (JDQL)
 - a simple language designed to be used inside the `@Query` annotation to specify the semantics of query methods of Jakarta Data repositories
 - a subset of Jakarta Persistence Query Language (JPQL)



Compatible Implementations

- Current version: 1.0
- Compatible implementations:
 - Open Liberty 24.0.0.6
 - Hibernate ORM 6.6.0.Final



Jakarta Expression Language (EL)

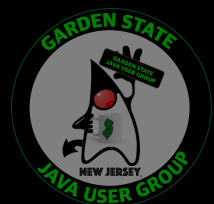


InfoQ



What is Jakarta EL (i)?

- Defines a simple language to meet the needs of the presentation layer in web applications that features:
 - a simple syntax restricted to the evaluation of expressions
 - variables and nested properties



What is Jakarta EL (ii)?

- Defines a simple language to meet the needs of the presentation layer in web applications that features:
 - relational, logical, arithmetic, conditional and empty operators
 - functions implemented as static methods on Java classes

What's New?

- The `java.desktop` module is no longer required at runtime
- A new property, `length`, is now supported for arrays



Compatible Implementations

- Current version: 6.0
- Compatible implementations:
 - Apache Tomcat 11.0.0-M18

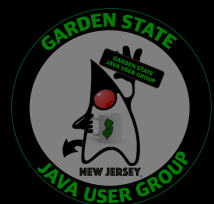


Jakarta Faces



What is Jakarta Faces?

- Defines a Model-View-Controller (MVC) framework for building user interfaces for web applications that include UI components, state management, event handling, input validation, page navigation, and support for internationalization and accessibility
- Formerly known as Jakarta Server Faces



What's New?

- Require firing events for Jakarta CDI `@Initialized`, `@BeforeDestroyed` and `@Destroyed` for build-in scopes
- Deprecate the unused `PreDestroyCustomScopeEvent` and `PostConstructCustomScopeEvent` classes

Compatible Implementations

- Current version: 4.1
- Compatible implementations:
 - GlassFish 8.0.0-M6
 - Eclipse Mojarra 4.1.1



Jakarta Interceptors



What is Jakarta Interceptors?

- Defines a means of interposing on business method invocations and specific events, such as lifecycle and timeout events, that occur on instances of Jakarta EE components and other managed classes

What's New?

- Updated dependencies to align with Jakarta EE 11
- New standard overloaded `getInterceptorBindings()` accessor methods to the `InvocationContext` interface for interceptor bindings



Compatible Implementations

- Current version: 2.2
- Compatible implementations:
 - Weld 6.0.0.Beta 1



Jakarta Pages



InfoQ



What is Jakarta Pages?

- Defines a template engine for web applications that supports mixing of textual content (including HTML and XML) with custom tags, expression language, and embedded Java code, that is ultimately compiled into a Jakarta Servlet
- Formerly known as Jakarta Server Pages



What's New?

- An updated `ErrorData` class to support the new Jakarta Servlet constant field values:
 - `ERROR_QUERY_STRING` and `ERROR_METHOD`
- All code deprecated as of Jakarta Server Pages 3.1 has been removed



Compatible Implementations

- Current version: 4.0
- Compatible implementations:
 - Apache Tomcat 11.0.0-M20

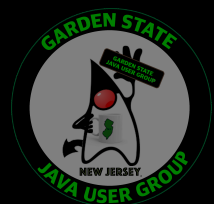


Jakarta Persistence



What is Jakarta Persistence?

- Defines a standard for the management of persistence and object/relational mapping in a Java environment



What's New? (i)

- Support for Java record types as embeddable classes
- Numerous new methods added to classes/interfaces such as:
 - `CriteriaQuery`, `EntityManager`, `PersistenceUnitUtil` and `Metamodel`



What's New? (ii)

- A migration to the `java.time` API
 - deprecate the use of the Java Calendar, `java.util.Date`, `java.sql.Date`, `Time` and `Timestamp` classes
- Remove all deprecated code from Jakarta Servlet 3.1



Compatible Implementations

- Current version 3.2
- Compatible implementations:
 - Hibernate ORM 7.0.0.Alpha 2
 - EclipseLink 5.0.0-B02



Jakarta RESTful Web Services (JAX-RS)



InfoQ



What is Jakarta JAX-RS?

- Provides a foundational API to develop web services following the Representational State Transfer (REST) architectural pattern



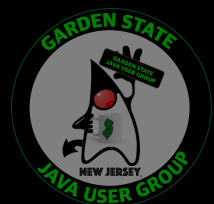
What's New? (i)

- A new convenience method, `containsHeaderString()`, added to the `ContainerResponseContext` interface, for checking header value lists
- A new method, `getMatchedResourceTemplate()`, added to the `UriInfo` interface, to obtain the template for the current URI request



What's New? (ii)

- Additional TCK tests for the **ExceptionHandler** interface and the Multipart/Form-Data API
- The deprecated **@ManagedBean** annotation and dependency on the Jakarta XML Binding specification were removed



Compatible Implementations

- Current version: 4.0
- Compatible implementations:
 - RESTEasy 7.0.0.Alpha1
 - Eclipse Jersey 4.0.0-M1



Jakarta Security



InfoQ



What is Jakarta Security?

- Defines a standard for creating secure Jakarta EE applications in modern application paradigms



What's New?

- Multiple authentication mechanisms
 - a new API that supports the user to choose
- Qualifiers for built-in authentication mechanisms
- References to `SecurityManager` class have been removed



Compatible Implementations

- Current version: 4.0
- Compatible implementations:
 - GlassFish 8.0.0-M6
 - Eclipse Soteria 4.0.0



Jakarta Servlet



InfoQ



What is Jakarta Servlet?

- Defines a server-side API for handling HTTP requests and responses



What's New?

- Allow control of status code and response body when sending a redirect
- Constants to support new HTTP status codes
- References to `SecurityManager` class have been removed



Compatible Implementations

- Current version: 6.1
- Compatible implementations:
 - Tomcat 11.0.0-M20

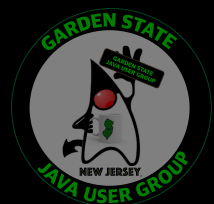


Jakarta Validation



What is Jakarta Validation? (i)

- Provides an object level constraint declaration and validation facility as well as a constraint metadata repository and query API
- Offers method and constructor validation facilities to ensure constraints on their parameters and return values



What is Jakarta Validation? (ii)

- Formerly known as Jakarta Bean Validation



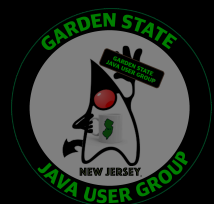
What's New?

- Support for Java Records
- Update dependencies for Jakarta EE 11



Compatible Implementations

- Current version: 3.1
- Compatible implementations:
 - Hibernate Validator 8.0.1.Final



Jakarta WebSocket



What is Jakarta WebSocket?

- Defines an API for client and server endpoints for the WebSocket protocol as defined by the Internet Engineering Task Force (IETF) RFC 6455



What's New? (i)

- Clarify the responsibilities for sending ping and pong messages
- Add a `getSession()` method to the `SendResult` class that returns the websocket `Session` in which the session was sent



What's New? (ii)

- Clarify the behavior of the `maxMessageSize` attribute, defined in the `@OnMessage` annotation, is set to a value larger than `Integer.MAX_VALUE`
- References to `SecurityManager` class have been removed



Compatible Implementations

- Current version 2.2
- Compatible implementations:
 - Apache Tomcat 11.0.0-M20
 - Eclipse Tyrus 2.2.0-M1



A Developer's Guide to Jakarta EE 11



Michael P. Redlich

Wayback Wednesday

