Jakarta EE 11: Going Beyond the Era of Java EE

Michael P. Redlich | @mpredli
Inspiration

• Jakarta EE 11: The First Big Leap for Jakarta
• Steve Millidge, CEO at Payara
• March 30, 2023
“From the initial lift and shift to the new namespace change in Jakarta EE 9, to the simplification work done in Jakarta EE 10, a lot of effort has gone into making Jakarta EE a solid basis for open source developers to build on.”
“With that done, there’s now the opportunity to start taking Jakarta EE beyond the Java EE era. With Java 21 [on the horizon], there’s now the opportunity to make sure Jakarta EE is always leveraging the latest and greatest capabilities of the new Java version, build new specifications and further unify and simplify the platform.”
Hi, I’m Mike 😊

- Java Champion
- Lead Java Queue News Editor: InfoQ
- Contract Dev Advocate and Tech Writer: Payara
- Director: Garden State Java User Group
- Leadership Council: Jakarta EE Ambassadors
- Committer: Jakarta NoSQL and Jakarta Data
- Senior Research Technician: ExxonMobil Technology & Engineering (retired with 33½ years service)
Objectives

- Evolution of Java EE/Jakarta EE
- Jakarta EE Profiles
- What’s New in Jakarta EE 11
- Updated Specifications in Jakarta EE 11
- Highlighted Specifications with Demos
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?
Evolution of Java EE/
Jakarta EE
Jakarta EE Evolution

JPE → J2EE 1.2 → J2EE 1.3 → J2EE 1.4 → Java EE 5

- Profiles, CDI, JAX-RS, Bean Validation
- WebSocket, JSON, Concurrency, Batch, pruning
- HTTP/2, SSE, Security, pruning
- Open-source governance

Java EE 6 → Java EE 7 → Java EE 8 → Jakarta EE 8

- Namespace transition
- New features, updates
- Community Innovation

Jakarta EE 9.x → Jakarta EE 10 → Jakarta EE 11

Michael P. Redlich | @mpredli
Jakarta EE Evolution

- **Java EE 6**
  - Profiles, CDI, JAX-RS, Bean Validation

- **Java EE 7**
  - WebSockets, JSON, Concurrency, Batch, pruning

- **Java EE 8**
  - HTTP/2, SSE, Security, pruning

- **Java EE 5**
  - EJB 3, JPA, JSF, JAXB, JAX-WS

- **Java EE 9**
  - New features, updates

- **Java EE 10**
  - Community Innovation

- **Java EE 11**
  - Open-source governance

- **Jakarta EE 8**
  - Jakarta EE 9.x

- **Jakarta EE 10**
  - Jakarta EE 11
Jakarta EE Evolution

- **JPE**
  - Profiles, CDI, JAX-RS, Bean Validation

- **J2EE 1.2**
  - Servlet, JSP, EJB, JMS

- **J2EE 1.3**
  - CMP, JCA

- **JAX-WS**
  - WebSocket, JSON, Concurrency, Batch, pruning

- **Java EE 6**
  - HTTP/2, SSE, Security, pruning

- **Java EE 7**
  - Open-source governance

- **Java EE 8**
  - Jakarta EE 8

- **Java EE 9.x**
  - Jakarta EE 9.x

- **Jakarta EE 10**
  - Community Innovation

- **Jakarta EE 11**
  - Jakarta EE 11
Jakarta EE Evolution

JPE -> J2EE 1.2 -> J2EE 1.3 -> J2EE 1.4 -> Java EE 5

Profiles, CDI, JAX-RS, Bean Validation -> WebSocket, JSON, Concurrency, Batch, pruning -> HTTP/2, SSE, Security, pruning

Java EE 6 -> Java EE 7 -> Java EE 8 -> Jakarta EE 8

Namespace transition -> New features, updates -> Community Innovation

Jakarta EE 9.x -> Jakarta EE 10 -> Jakarta EE 11

May 11, 2006 J2SE 5.0 (Tiger)
December 10, 2009 | Java SE 6 (Mustang)

Jakarta EE Evolution

JPE → J2EE 1.2 → J2EE 1.3 → J2EE 1.4 → Java EE 5

Profiles, CDI, JAX-RS, Bean Validation

WebSocket, JSON, Concurrency, Batch, pruning

HTTP/2, SSE, Security, pruning

Open-source governance

Java EE 6 → Java EE 7 → Java EE 8 → Jakarta EE 8

Namespace transition → New features, updates → Community Innovation

Jakarta EE 9.x → Jakarta EE 10 → Jakarta EE 11
Java EE Evolution

JPE → J2EE 1.2 → J2EE 1.3 → J2EE 1.4 → Java EE 5

Profiles, CDI, JAX-RS, Bean Validation

WebSocket, JSON, Concurrency, Batch, pruning

HTTP/2, SSE, Security, pruning

Open-source governance

Java EE 6 → Java EE 7 → Java EE 8 → Jakarta EE 8

Namespace transition → New features, updates → Community Innovation

Jakarta EE 9.x → Jakarta EE 10 → Jakarta EE 11

August 17, 2017

Java SE 8
Jakarta EE Evolution

JPE → J2EE 1.2 → J2EE 1.3 → J2EE 1.4 → Java EE 5

- Servlet, JSP, EJB, JMS
- CMP, JCA
- JAX-WS
- EJB 3, JPA, JSF, JAXB, JAX-WS

Java EE 6 → Java EE 7 → Java EE 8

- Profiles, CDI, JAX-RS, Bean Validation
- WebSocket, JSON, Concurrency, Batch, pruning
- HTTP/2, SSE, Security, pruning
- Open-source governance

JAKARTA EE

Jakarta EE 9.x → Jakarta EE 10 → Jakarta EE 11

- Namespace transition
- New features, updates
- Community Innovation

September 10, 2019
Java SE 8
Objectives

• Evolution of Java EE/Jakarta EE
• Jakarta EE Profiles
• What’s New in Jakarta EE 11
• Updated Specifications in Jakarta EE 11
• Highlighted Specifications with Demos
Jakarta EE Profiles
Jakarta EE Profiles

- Jakarta EE Platform
- Jakarta EE Web Profile
- Jakarta EE Core Profile
Jakarta EE Platform

• Defines a standard platform for hosting all Jakarta EE applications

• Designed for developers who require the full set of Jakarta EE specifications for developing enterprise applications
<table>
<thead>
<tr>
<th>Authorization 3.0</th>
<th>Concurrency 3.1</th>
<th>Data 1.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activation 2.1</td>
<td>CDI 4.1</td>
<td>Persistence 3.2</td>
</tr>
<tr>
<td>Batch 2.1</td>
<td>Expression Language 6.0</td>
<td>RESTful Web Services 4.0</td>
</tr>
<tr>
<td>Connectors 2.1</td>
<td>Faces 4.1</td>
<td>Pages 4.0</td>
</tr>
<tr>
<td>Mail 2.1</td>
<td>Security 4.0</td>
<td>WebSocket 2.2</td>
</tr>
<tr>
<td>Messaging 3.1</td>
<td>Servlet 6.1</td>
<td>Validation 3.1</td>
</tr>
<tr>
<td>Enterprise Beans 4.0</td>
<td>Standard Tag Libraries 3.0</td>
<td>Debugging Support 2.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Enterprise Beans Lite 4.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dependency Injection 2.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CDI Lite 4.1</td>
</tr>
</tbody>
</table>
Jakarta EE Web Profile

- Defines a subset of the Jakarta EE Platform that contains web technologies specifically targeted for developing web applications
<table>
<thead>
<tr>
<th>Blue</th>
<th>Blue</th>
<th>Blue</th>
<th>Blue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authentication 3.1</td>
<td>Data 1.0</td>
<td>Persistence 3.2</td>
<td>RESTful Web Services 4.0</td>
</tr>
<tr>
<td>Concurrency 3.1</td>
<td></td>
<td>Pages 4.0</td>
<td>JSON Processing 2.1</td>
</tr>
<tr>
<td>CDI 4.1</td>
<td></td>
<td>WebSocket 2.2</td>
<td>JSON Binding 3.0</td>
</tr>
<tr>
<td>Expression Language 6.0</td>
<td></td>
<td>Validation 3.1</td>
<td>Annotations 3.0</td>
</tr>
<tr>
<td>Faces 4.1</td>
<td></td>
<td>Debugging Support 2.0</td>
<td>Interceptors 2.2</td>
</tr>
<tr>
<td>Security 4.0</td>
<td></td>
<td>Enterprise Beans Lite 4.0</td>
<td>Dependency Injection 2.0</td>
</tr>
<tr>
<td>Servlet 6.1</td>
<td></td>
<td>Transactions 2.0</td>
<td>CDI Lite 4.1</td>
</tr>
<tr>
<td>Standard Tag Libraries 3.0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Updated
- Not Updated
- New
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
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
Objectives

- Evolution of Java EE/Jakarta EE
- Jakarta EE Profiles
- What’s New in Jakarta EE 11
- Updated Specifications in Jakarta EE 11
- Highlighted Specifications with Demos
What's New in Jakarta EE 11
Release Timeline (i)

- First milestone release:
  - December 2023 ✅

- Second milestone release:
  - March 2024 ✅

- Third milestone release:
  - April 2024 ✅
Release Timeline (ii)

• Fourth milestone release:
  • May 2024

• Final GA release:
  • June/July 2024
What's New (i)

• New specifications
  • Jakarta Data | MicroProfile JWT Bridge
  • Removal of legacy @ManagedBean annotation
    • replace with CDI across the entire platform
  • References to SecurityManager class have been removed
What's New (ii)

- Clarify JPMS Requirements
- CDI Lite
- Optional features
  - Jakarta SOAP with Attachments
  - Jakarta XML Web Services
  - Jakarta XML Binding
Spec Updates (i)

• Jakarta Validation
  • supports validating records

• Jakarta Persistence
  • Records will be usable as an embeddable type

• Jakarta Concurrency
  • support for Virtual Threads
Spec Updates (ii)

- Jakarta RESTful Web Services
  - improved integration with Jakarta CDI and Jakarta Concurrency

- Jakarta Security
  - APIs for the authorization theme, including interceptors and an abstraction for the permission store
  - MicroProfile JWT and Jakarta Security
Objectives

• Evolution of Java EE/Jakarta EE

• Jakarta EE Profiles

• What’s New in Jakarta EE 11

• Updated Specifications in Jakarta EE 11

• Highlighted Specifications with Demos
Updated Specifications in Jakarta EE 11
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 Persistence 3.2
Updated Specifications (ii)

- Jakarta RESTful Web Services 4.0
- Jakarta Security 4.0
- Jakarta Server Pages 4.0
- Jakarta Servlet 6.1
- Jakarta Validation 3.1
- Jakarta WebSocket 2.2
Objectives

- Evolution of Java EE/Jakarta EE
- Jakarta EE Profiles
- What’s New in Jakarta EE 11
- Updated Specifications in Jakarta EE 11
- Highlighted Specifications with Demos
Highlighted Specifications with Demos
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
MicroProfile JWT Bridge
What is MicroProfile JWT Bridge? (i)

- Collaboration between Jakarta EE and MicroProfile Working Groups
- Enable Jakarta Security to build on MicroProfile JWT Authentication to provide seamless integrations
  - eliminates duplication of effort and circular dependencies
What is MicroProfile JWT Bridge? (ii)

- New standalone specification in MicroProfile
- Move the optional section of MicroProfile JWT Authentication to the new bridge specification together with TCKs
MicroProfile 6.1

- Telemetry 1.1
- Open API 3.1
- Rest Client 3.0
- Config 3.1
- Fault Tolerance 4.0
- Metrics 5.1
- JWT Authentication 2.1
- Health 4.0

Jakarta EE 10 Core Profile

Standalone

- Open Tracing 3.0
- LRA 2.0
- Reactive Messaging 3.0
- GraphQL 2.0
- Reactive Streams Operators 3.0
- Context Propagation 1.3

Outside umbrella

- New
- Updated
- No change from last release
Live Demo
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.
Compatible Implementations

- Previous version: N/A
- Current version: 1.0.1
- Compatible implementations:
  - Open Liberty 24.0.0.6-beta and Hibernate ORM 6.6.0.Final (Jakarta EE 11)
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)
Live Demo
Jakarta Security
What is Jakarta Security?

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

- Previous version: 3.0.0
- Current version: 4.0.0-M2
- Compatible implementations:
  - Eclipse Soteria 3.0.2 (Jakarta EE 10)
  - Eclipse Soteria (TBD) (Jakarta EE 11)
What's New?

• Multiple authentication mechanisms
• Interoperability with MicroProfile JWT Authentication
• Alternative to @RolesAllowed
• References to SecurityManager class have been removed
Jakarta Servlet
What is Jakarta Servlet?

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

• Previous version: 6.0.0
• Current version: 6.1.0
• Compatible implementations:
  • GlassFish 7.0.0-M4 (Jakarta EE 10)
  • Tomcat 11.0.0-M20 (Jakarta EE 11)
What’s New?

- Allow control of status code and response body when sending a redirect
- Add a query string attribute to error dispatches
- Add constants for new HTTP status codes
- References to SecurityManager class have been removed
Live Demo
Objectives

• Evolution of Java EE/Jakarta EE ✅
• Jakarta EE Profiles ✅
• What’s New in Jakarta EE 11 ✅
• Updated Specifications in Jakarta EE 11 ✅
• Highlighted Specifications with Demos ✅
Want to Get Involved?

- Contribute to open source!
  - Share
  - Embrace
  - Engage
  - Develop

- Share your own Jakarta EE experiences!

Tanja Obradovic
Jakarta EE Program Manager at Eclipse Foundation
Acknowledgements (i)

- Steve Millidge
  - CEO at Payara

- Ivar Grimstad
  - Jakarta EE Developer Advocate at Eclipse Foundation

- Adam Yoho
  - Back End Developer at IBM
Acknowledgements (ii)

• Sergey Beryozkin
  • Principal Software Engineer at Red Hat

• Otávio Santana
  • Independent Software Consultant at OS Expert

• Maximillian Arruda
  • Senior Backend Software Engineer at Digibee
Acknowledgements (iii)

- Arjan Tijms
  - Director at OmniFishEE
- Luis Neto
  - Java Software Engineer at Payara
- David Blevins
  - CEO at Tomitribe
Contact Info

mike@redlich.net
@mpredli
redlich.net
redlich.net/portfolio
github.com/mpredli01
Thanks!
Backup
Michael Redlich

Jakarta EE 11: Going Beyond the Era of Java EE

JAVA PRO # JCON2023 www.jcon.one

NOV 21-23

GARDEN STATE

InfoQ payara®
Jakarta EE 11: Going Beyond the Era of Java EE

December 14 2023  3:20 PM GMT
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.
Compatible Implementations

- Previous version: 2.1.0
- Current version: 3.0.0
- Compatible implementations:
  - GlassFish 7.0.0-M1 (Jakarta EE 10)
  - GlassFish 8.0.0-M3 (Jakarta EE 11)
What’s New?

- Removal of the deprecated `@ManagedBeans` annotation
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
Compatible Implementations

• Previous version: 3.0.0
• Current version: 3.1.0-M1
• Compatible implementations:
  • GlassFish 7.0.0-M7 (Jakarta EE 10)
  • GlassFish TBD (Jakarta EE 11)
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
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
Compatible Implementations

- Previous version: 2.1.0
- Current version: 3.0.0
- Compatible implementations:
  - Eclipse Exousia 2.1.0-M1 and GlassFish 7.0.0-M4 (Jakarta EE 10)
  - Eclipse Exousia 3.0.0-M3 and GlassFish 8.0.0-M5 (Jakarta EE 11)
What’s New?

- 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 `java.security.Policy`
- References to `SecurityManager` class have been removed
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
Compatible Implementations

- Previous version: 3.0.0
- Current version: 3.1.0
- Compatible implementations:
  - Open Liberty 22.0.0.5-beta (Jakarta EE 10)
  - Open Liberty 24.0.0.6-beta (Jakarta EE 11)
What’s New?

• Integration with Java 21 Virtual Threads
• Java Flow/ReactiveStreams and context propagation
• Replace more features from EJB
  • such as Schedule and Lock annotations
• Become more CDI-centric
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)
Compatible Implementations

• Previous version: 4.0.0
• Current version: 4.1.0
• Compatible implementations:
  • Weld 5.0.0.SP2 (Jakarta EE 10)
  • Weld 6.0.0.Beta1 (Jakarta EE 11)
What’s New?

• Add methods to BeanConfigurator interface for applying decorators
• Support for @Priority annotation on producers
Jakarta Expression Language (EL)
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
Compatible Implementations

- Previous version: 5.0.0
- Current version: 6.0.0
- Compatible implementations:
  - Eclipse ExpressLy 5.0.0-M2 (Jakarta EE 10)
  - Apache Tomcat 11.0.0-M18 (Jakarta EE 11)
What's New?

• The `java.desktop` module is no longer required at runtime

• A new property, `length`, is now supported for arrays
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.
Compatible Implementations

- Previous version: 4.0.0
- Current version: 4.1.0
- Compatible implementations:
  - Eclipse Mojarra 4.0.4 (Jakarta EE 10)
  - Eclipse Mojarra 4.1.0 and GlassFish 8.0.0-M6 (Jakarta EE 11)
What's New?

• Require firing events for @Initialized, @BeforeDestroyed and @Destroyed for build-in scopes

• Remove unused PreDestroyCustomScopeEvent and PostConstructCustomScopeEvent classes
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.
Compatible Implementations

- Previous version: 2.1.0
- Current version: 2.2.0
- Compatible implementations:
  - Weld 5.0.0.Beta1 (Jakarta EE 10)
  - Weld 6.0.0.Beta1 (Jakarta EE 11)
What's New?

• Updated dependencies to align with Jakarta EE 11

• Add standard accessor methods to the InvocationContext interface for interceptor bindings
Jakarta Pages
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.
compatible implementations

- Previous version: 3.1.0
- Current version: 4.0.0
- Compatible implementations:
  - Eclipse WaSP 3.1.0-M3 (Jakarta EE 10)
  - Apache Tomcat 11.0.0-M20 (Jakarta EE 11)
What's New?

- Updated `ErrorData` class to add support for the new attribute: `jakarta.servlet.error.query_string`
- All code deprecated as of Jakarta Server Pages 3.1 has been removed
- Formerly known as Jakarta Server Pages
Jakarta Persistence
What is Jakarta Persistence?

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

- Previous version: 3.1.0
- Current version 3.2.0
- Compatible implementations:
  - EclipseLink 4.0.0-M3 and Hibernate ORM 6.0.0.Final (Jakarta EE 10)
  - EclipseLink 5.0.0-B02 and Hibernate ORM 7.0.0.Alpha 2 (Jakarta EE 11)
What’s New?

- Align with Jakarta Servlet and Jakarta Expression Language specifications
- Remove all deprecated code from Jakarta Servlet 3.1
Jakarta RESTful Web Services (JAX-RS)
What is Jakarta JAX-RS?

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

- Previous version: 3.1.0
- Current version: 4.0.0
- Compatible implementations:
  - Eclipse Jersey 3.1.0 (Jakarta EE 10)
  - Eclipse Jersey 4.0.0-M1 and RESTEasy 7.0.0.Alpha1 (Jakarta EE 11)
What’s New?

• Define CDI scopes for all Jakarta REST types
Jakarta Validation
What is Jakarta Validation?

• 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
Compatible Implementations

- Previous version: 3.0.0
- Current version: 3.1.0
- Compatible implementations:
  - Hibernate Validator 7.0.0.Alpha5 (Jakarta EE 9)
  - Hibernate Validator 8.0.1.Final (Jakarta EE 11)
What’s New?

- Support for Java Records
- Formerly known as Jakarta Bean Validation
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) RFC6455
Compatible Implementations

- Previous version: 2.1.0
- Current version 2.2.0
- Compatible implementations:
  - Eclipse Tyrus 2.1.0-M3 (Jakarta EE 10)
  - Eclipse Tyrus 2.2.0-M1 and Apache Tomcat 11.0.0-M20 (Jakarta EE 11)
What's New?

- References to `SecurityManager` class have been removed
Jakarta EE 11
Going Beyond the Era of Java EE

Michael P. Redlich