Jakarta EE

Mark Thomas
Introductions

- Mark Thomas
- Tomcat committer since 2003
- ASF member since 2007
- markt@apache.org
- ASF, Eclipse, VMware
History
History

• Java EE
  – Collection of individual specifications
  – Platform specification

• Java Community Process
  – Produced the specifications
  – Each specification had a lead and an expert group
  – Essentially Oracle controlled

• Open source (mostly)
History

- Technology Compatibility Kit
  - TCK
  - Used to demonstrate compliance with the specification
  - Restrictively licensed

- In 2017 Oracle announce donation of Java EE
  - Eclipse
  - Becomes Jakarta EE
History

• Java EE 8
  – 2017/08
  – Java 8

• Jakarta EE 8
  – Identical to Java EE 8
  – Just a rebrand
  – 2019/09
  – A lot of behind the scenes admin
History

- **Jakarta EE 9**
  - 2020/12
  - Packages renamed from javax... to jakarta...
  - Java 8

- **Jakarta EE 9.1**
  - 2021/05
  - No functional changes
  - Added Java 11
• Jakarta EE 10
  – 2022/H2
  – Java 11 and 17
  – First release with functional changes
Specifications
Specifications

- Servlet
- Pages (JSPs)
- Expression Language
- WebSocket
- Authentication (JASPI)C
- Annotations
Servlet 6.0
Servlet 6.0

- Deprecated classes and methods removed
  - SingleThreadModel
  - HttpSessionContext
  - HttpUtils

- Cookies
  - RFC 6265
  - Generic attribute support
Servlet

- URI normalization and decoding
  - Suspicious URIs / . . ; /
- Container responsible for handling HEAD requests
  - legacy mode still available
- Request and connection identifiers
- Relax wrapping requirements
- Various clarifications
• PropertyNotFoundException for unknown EL identifier
• Deprecations
  – EL alignment
  – isThreadSafe (SingleThreadModel)
  – jsp:plugin
Expression Language 5.0
Expression Language 5.0

- Generics
- Usage improvements
  - Default method support for BeanELResolver
  - Array coercion
- Clarify method matching rules
  - Standard Java plus coercion
- Coerce LambdaExpression to FunctionalInterface invocation
Expression Language 5.0

- Deprecate `getFeatureDescriptors()`
  - JPMS remove java.desktop dependency
- Add `MethodReference` class
- `ELResolver.getType()` changes
  - Must return null if resolved property is read-only.
  - Changes `StaticFieldELResolver`
WebSocket 2.1
WebSocket 2.1

- Client TLS configuration
  - SSLContext
- Remove register at deployment only restriction
- Packaging improvements
  - JPMS
- Clarifications
Others
Others

- Annotations
  - Added @Nullable and @Nonnull
  - Minor changes

- Authentication
  - Simplify the typical case
Jakarta EE 11
Jakarta EE 11

- Remove SecurityManager
- Servlet
  - Low level HTTP API
  - Lots of clarifications
  - Minor improvements
- JSP
  - Remove deprecated code
Jakarta EE 11

• Expression Language
  – Remove deprecated code
  – Usage improvements

• WebSocket
  – Clarifications
  – Interaction with Servlet and originating HTTP request
  – Enhancements
Jakarta EE 11

• Annotations
  – Not known

• Authentication
  – Possible merger with Authorization
THANK YOU

markt@apache.org

https://tomcat.apache.org/