

What's new and noteworthy in Java EE 8?



13.03.2018

Dirk Weil, GEDOPLAN GmbH

Dirk Weil

- ≡ GEDOPLAN GmbH, Bielefeld
 - ≡ GEDOPLAN IT Consulting
Consulting, coaching, concepts, reviews, development
 - ≡ GEDOPLAN IT Training
Java, JEE, tools trainings in Berlin, Bielefeld, on-site
- ≡ JEE since 1998
- ≡ Speaker and author



New and noteworthy in CDI 2.0

☰ Observer ordering

```
@ApplicationScoped
public class EventProducer {

    @Inject
    Event<DemoEvent> eventSource;

    public void fireSyncEvent() {
        this.eventSource.fire(new DemoEvent());
    }
}
```

```
@ApplicationScoped
public class EventObserver {

    void a(@Observes @Priority(Interceptor.Priority.APPLICATION + 1) DemoEvent e) {
        ...
    }

    void b(@Observes @Priority(Interceptor.Priority.APPLICATION + 2) DemoEvent e) {
        ...
    }
}
```

New and noteworthy in CDI 2.0

≡ Asynchronous events

```
@ApplicationScoped
public class EventProducer {

    @Inject
    Event<DemoEvent> eventSource;

    public void fireAsyncEvent() {
        CompletionStage<DemoEvent> completionStage
            = this.eventSource.fireAsync(new DemoEvent());
    }
}
```

```
@ApplicationScoped
public class EventObserver {

    void x(@ObservesAsync DemoEvent e) {
        ...
    }

    void y(@ObservesAsync DemoEvent e) {
        ...
    }
}
```

New and noteworthy in CDI 2.0

≡ SE container

```
SeContainerInitializer seContainerInitializer
    = SeContainerInitializer.newInstance();

try (SeContainer container = seContainerInitializer.initialize()) {

    HelloWorldService helloWorldService
        = container.select(HelloWorldService.class).get();

    this.log.debug(helloWorldService.getHelloWorld());
}
```

```
<dependency>
  <groupId>org.jboss.weld.se</groupId>
  <artifactId>weld-se-shaded</artifactId>
  <version>3.0.3.Final</version>
</dependency>
```

```
<dependency>
  <groupId>org.apache.openwebbeans</groupId>
  <artifactId>openwebbeans-se</artifactId>
  <version>2.0.4</version>
</dependency>
```

New and noteworthy in CDI 2.0

- ≡ Java 8 alignment
 - ≡ Repeatable qualifier annotations
 - ≡ `Instance.stream()`
- ≡ Various minor adjustments, e. g.
 - ≡ Configurator interfaces
 - ≡ Annotation literals
- ≡ No context control in SE ☹
 - ≡ Keep using DeltaSpike `ContextControl`

```
@Qualifier
@Retention(...) @Target(...)
@Repeatable(List.class)
public @interface Greeting {

    GreetingType type();

    @Retention(...) @Target(...)
    @interface List {
        Greeting[] value();
    }
}
```

New and noteworthy in Bean Validation 2.0

≡ New constraints

- ≡ `@Email`, `@NotEmpty`, `@NotBlank`, `@Positive`,
`@PositiveOrZero`, `@Negative`, `@NegativeOrZero`,
`@PastOrPresent`, `@FutureOrPresent`

≡ Container element constraints

```
List<@Positive @Max(6) Integer> marks;
```

≡ Java 8 alignment

- ≡ Repeatable constraint annotations
- ≡ Support for Java 8 date and time types
- ≡ Support for `Optional<T>`

```
Optional<@Email String> getEmail() {  
    ...  
}
```

New and noteworthy in JPA 2.2

≡ Java 8 alignment

≡ Repeatable annotations

```
public class Employee {  
    @AttributeOverride(name = "street", column = @Column(name = "HOME_STREET"))  
    @AttributeOverride(name = "zipCode", column = @Column(name = "HOME_ZIPCODE"))  
    @AttributeOverride(name = "city", column = @Column(name = "HOME_CITY"))  
    private Address homeAddress;  
}
```

≡ Query.getResultStream()

```
public class DocumentRepository {  
    public Stream<Document> findAllAsStream() {  
        return entityManager  
            .createQuery("select x from Document x", Document.class)  
            .getResultStream();  
    }  
}
```

≡ Allow CDI injections into AttributeConverter

New and noteworthy in JPA 2.2

☰ Support for Java 8 date and time types

Java Object Type	JDBC Type
<code>LocalDate</code>	<code>DATE</code>
<code>LocalTime</code>	<code>TIME</code>
<code>LocalDateTime</code>	<code>TIMESTAMP</code>
<code>OffsetTime</code>	<code>TIME_WITH_TIMEZONE (*)</code>
<code>OffsetDateTime</code>	<code>TIMESTAMP_WITH_TIMEZONE (*)</code>

(*) rarely supported by databases

New and noteworthy in JSF 2.3

☰ Java Time Support in `<f:convertDateTime>`

```
<h:outputText id="founded" value="#{countryPresenter.someCountry.founded}">  
  <f:convertDateTime type="localDate" />  
</h:outputText>
```

```
public class Country {  
    private LocalDate founded;
```

Class	type values	Default converter
Date	date, time, both	DateFormat
LocalDate	localDate	DateTimeFormatter
LocalTime	localTime	
LocalDateTime	localDateTime	
OffsetTime	offsetTime	
OffsetDateTime	offsetDateTime	
ZonedDateTime	zonedDateTime	

New and noteworthy in JSF 2.3

- ≡ CDI integration
 - ≡ Deprecate `javax.faces.bean`
 - ≡ Allow injection of specific JSF artifacts (e.g. `FacesContext`)
 - ≡ Support CDI injections into converters

```
@FacesConverter(forClass = Country.class, managed = true)  
public class CountryConverter implements Converter<Country> {  
    @Inject  
    CountryRepository countryRepository;  
}
```

- ≡ broken in current implementations
 - ≡ Mojarra 2.3.3: Injection not executed
 - ≡ MyFaces 2.3.0: NPE

New and noteworthy in JSF 2.3

☰ WebSocket integration

```
<f:websocket
  channel="ticker"
  onmessage="function(message) {
    document.getElementById('coolNewFeature').value=message;
  }" />

<h:inputText id="coolNewFeature" value="" size="25" readonly="true" />
```

```
public class TickerEndpoint {

  @Inject
  @Push(channel = "ticker")
  private PushContext channel;

  public void tick() {
    String message = ...
    this.channel.send(message);
  }
}
```

New and noteworthy in JSF 2.3

≡ Multi-field validation

- ≡ Use non-default bean validation group for one or more input fields

```
<h:inputText value="#{questionnaire.email}">
  <f:validateBean validationGroups="InitialInput" />
</h:inputText>

<h:inputText value="#{questionnaire.comment}">
  <f:validateBean validationGroups="InitialInput" />
</h:inputText>

<f:validateWholeBean value="#{questionnaire}"
  validationGroups="InitialInput" />
```

- ≡ Use `<f:validateWholeBean>` after input fields
- ≡ Enable feature in `web.xml`
- ≡ Works by validating a temporary copy

- ≡ Not fully functional on Mojarra 2.3.3 ☹️
- ≡ Incredibly bloated ☠️

New and noteworthy: JSON-B 1.0

- ≡ Java – JSON binding
- ≡ Similar to JAXB

- ≡ Mapping annotations
 - ≡ `@JsonbVisibility`
 - ≡ `@JsonbNillable`
 - ≡ `@JsonbTransient`
 - ≡ `@JsonbTypeAdapter`
 - ≡ `@JsonbDateFormat`
 - ≡ `@JsonbNumberFormat`

```
@JsonbNillable
public class Country {

    private String id;

    private String name;

    @JsonbTypeAdapter(ContinentConverter.class)
    private Continent continent;

    @JsonbTransient
    private String dummy = "unused";
```

New and noteworthy: JSON-B 1.0

≡ (De)serialization API: `JsonbBuilder`, `Jsonb`

```
Country country = ...
```

```
try (Jsonb jsonb = JsonbBuilder.create()) {  
    String json = jsonb.toJson(country);  
    ...  
}
```

```
String json = ...
```

```
try (Jsonb jsonb = JsonbBuilder.create()) {  
    Country country = jsonb.fromJson(json, Country.class);  
    ...  
}
```

≡ Configuration:

≡ Class level annotations (`@JsonbNillable`, ...)

≡ `JsonbConfig`

```
JsonbConfig jsonbConfig = new JsonbConfig().withFormatting(true);  
Jsonb jsonb = JsonbBuilder.create(jsonbConfig);
```

New and noteworthy in Servlet 4.0

- ≡ HTTP/2 support
 - ≡ Header compression
 - ≡ Request multiplexing:
Allow multiple parallel requests on same connection
 - ≡ Server push:
Send content probably needed in advance

```
protected void doGet(HttpServletRequest req, HttpServletResponse resp)
    throws IOException {

    PushBuilder pushBuilder = req.newPushBuilder();
    if (pushBuilder != null) {
        pushBuilder.path("/dummy.css").push();
    }
    ...
}
```

- ≡ In general „behind the scenes“

New and noteworthy in JAX-RS 2.1

☰ Reactive client support

```
Client client= ClientBuilder.newClient();

CompletionStage<String> planetearthCompletionStage = client
    .path("http://localhost:8080/whats-new-in-jee8/rest/planetearth")
    .request()
    .accept(MediaType.TEXT_PLAIN)
    .rx()
    .get(String.class)
    .exceptionally(e -> "the ultimate question of life, the universe, and everything");
```

New and noteworthy in JAX-RS 2.1

☰ SSE support

```
@Path("sse")
public class SseResource {
    @Context Sse sse;

    SseBroadcaster sseBroadcaster;

    @PostConstruct
    void init() {
        this.sseBroadcaster = this.sse.newBroadcaster();
    }

    @GET
    @Produces(MediaType.SERVER_SENT_EVENTS)
    public void register(@Context SseEventSink sseEventSink) {
        this.sseBroadcaster.register(sseEventSink);
    }

    public void tick() {
        this.sseBroadcaster.broadcast(this.sse.newEvent("Hello, it is " + new Date()));
    }
}
```

New and noteworthy in JAX-RS 2.1

☰ SSE support

```
try (SseEventSource sseEventSource = SseEventSource
    .target("http://localhost:8080/whats-new-in-jee8/rest/sse")
    .build()) {

    sseEventSource.register(e -> log.debug("Event received: " + e));

    sseEventSource.open();
```

Odds and ends

- ≡ JSON-P
 - ≡ JSON-Pointer
 - ≡ JSON-Patch
 - ≡ JSON-Big-Data

- ≡ Java EE Security API
 - ≡ Authentication Mechanism
 - ≡ Identity Store
 - ≡ Security Context

<TTFN>

- ≡ ⑧ - ⑦ = 3 years, 5 months
- ≡ Evolution or maintenance release?
- ≡ Implementations
 - ≡ GlassFish 5.0 (big areas nonfunctional)
 - ≡ Open Liberty (daily build)
 - ≡ Payara 5.0.0 (still alpha)
 - ≡ WildFly 12.0.0.Final (preview, missing some parts)

The future ...



The screenshot shows the Eclipse website's navigation and content for Eclipse Enterprise for Java. At the top left is the Eclipse logo, consisting of a stylized 'E' icon and the word 'eclipse'. Below the logo is a navigation menu with links for 'GETTING STARTED', 'MEMBERS', 'PROJECTS', and 'MORE' with a dropdown arrow. A dark blue breadcrumb trail reads 'HOME / PROJECTS / ECLIPSE ENTERPRISE FOR JAVA'. The main heading is 'Eclipse Enterprise for Java'. Below this is a secondary navigation menu with tabs for 'Overview', 'Downloads', 'Who's Involved', 'Developer Resources', 'Governance', and 'Contact Us'. The 'Overview' tab is selected. The main content area contains a paragraph describing Eclipse Enterprise for Java (EE4J) as an open source initiative to create standard APIs, implementations, and technology compatibility kits for Java runtimes, based on the Java™ Platform, Enterprise Edition (Java EE) standards, and using Java EE 8 as the baseline.

 **eclipse**

[GETTING STARTED](#) [MEMBERS](#) [PROJECTS](#) [MORE ▾](#)

[HOME](#) / [PROJECTS](#) / [ECLIPSE ENTERPRISE FOR JAVA](#)

Eclipse Enterprise for Java

[Overview](#) [Downloads](#) [Who's Involved](#) [Developer Resources](#) [Governance](#) [Contact Us](#)

Eclipse Enterprise for Java (EE4J) is an open source initiative to create standard APIs, implementations of those APIs, and technology compatibility kits for Java runtimes that enable development, deployment, and management of server-side and cloud-native applications. EE4J is based on the Java™ Platform, Enterprise Edition (Java EE) standards, and uses Java EE 8 as the baseline for creating new standards.

More

≡ github.com/GEDOPLAN/whats-new-in-jee8
Demo project

≡ www.gedoplan-it-training.de
Trainings in Berlin, Bielefeld, inhouse

≡ www.gedoplan-it-consulting.de
Reviews, Coaching, ...
Blog

≡  dirk.weil@gedoplan.de

≡  [@dirkweil](https://twitter.com/dirkweil)

