OSGi CDI Integration Specification

Raymond Augé - Sr. Software Architect

Why CDI In OSGi?

- Reduce developer friction
- Important Java specification
- Benefit from extensive feature set

But Declarative Services (DS)?

Liferay loves DS!

99% of all **Liferay** bundles (jars) are DS and the vast majority will remain DS forever.

WHY use anything else?

by design DS is...

ultra light weight, DS annotations are syntax sugar, **CLASS retention** and processed at build time, runtime overhead is extremely low, does not provide an integration SPI, does not provide intra-bundle dependency injection.

CDI

as part of its feature set, is...

- **extensible** (CDI has a full fledged SPI)
- **annotation processing** engine
- **intra-bundle** dependency injection

Custom annotations!

CDI

allows for...

completely internal wiring.

DS - Internal wiring: new

```java
@Component
public class FooImpl {
  private Pojo pojo;

  public FooImpl() {
    pojo = new PojoImpl();
  }
}
```

CDI - Internal wiring: @Inject

```java
public class FooImpl {
  private Pojo pojo;

  @Inject
  public FooImpl(Pojo pojo) {
    this.pojo = pojo;
  }
}
```

DS - Services: singleton

```java
@Component
public class FooImpl implements Function {
  ...
}
```

OSGi-CDI - Services: singleton

```java
@Service
public class FooImpl implements Function {
  ...
}
```

DS - Services: prototype

```java
@Component(scope = PROTOTYPE)
public class FooImpl implements Function {
  ...
}
```

OSGi-CDI - Services: prototype

```java
@Service @ServiceInstance(PROTOTYPE)
public class FooImpl implements Function {
  ...
}
```

DS - References

```
@Reference
Pojo pojo;
```

OSGi-CDI - References

```
@Inject @Reference
Pojo pojo;
```

DS - Cardinality: mandatory

```
@Reference
Pojo pojo;
```

OSGi-CDI - Cardinality: mandatory

```
@Inject @Reference
Pojo pojo;
```

DS - Cardinality: optional

```
@Reference(cardinality = OPTIONAL)
Pojo pojo;
```

OSGi-CDI - Cardinality: optional

```
@Inject @Reference
Optional<Pojo> pojo;
```

DS - Cardinality: multiple

```
@Reference
List<Pojo> pojos;
```

OSGi-CDI - Cardinality: multiple

```
@Inject @Reference
List<Pojo> pojos;
```

DS - Cardinality: at least one (or n)

```
@Reference(cardinality = AT_LEAST_ONE)
List<Pojo> pojos;
```

OSGi-CDI - Cardinality: at least one (or n)

```java
@Inject @Reference @MinimumCardinality(1)
List<Pojo> pojos;
```

DS - Reference Policy: greedy

```
@Reference(policyOption = GREEDY)
Pojo pojo;
```

OSGi-CDI - Reference Policy: reluctant

```
@Inject @Reference @Reluctant
Pojo pojo;
```

DS - Dynamic: mandatory

```java
@Reference(policy = DYNAMIC)
volatile Pojo pojo;
```

OSGi-CDI - Dynamic: mandatory

```java
@Inject @Reference
Provider<Pojo> pojo;
```

DS - Dynamic: multiple

```
@Reference(policy = DYNAMIC)
volatile List<Pojo> pojos;
```

OSGi-CDI - Dynamic: multiple

```java
@Inject @Reference
Provider<List<Pojo>> pojos;
```

DS - Dynamic: optional

```java
@Reference(policy = DYNAMIC, cardinality = OPTIONAL)
volatile Pojo pojo;
```

OSGi-CDI - Dynamic: optional

```java
@Inject @Reference
Provider<Optional<Pojo>> pojo;
```

DS - OSGi Logger

```java
@Reference(service = LoggerFactory.class)
Logger logger;
```

OSGi-CDI - OSGi Logger

```
@Inject
Logger logger;
```

DS - Configuration

```
@Activate
Map<String, Object> props;
```

OSGi-CDI - Configuration

```java
@Inject @ComponentProperties
Map<String, Object> props;
```

Configuration Types

```java
@Retention(RUNTIME)
@BeanPropertyType // OSGi-CDI
@ComponentPropertyType // DS
public @interface Config {
  String hostname() default "localhost";
  int port() default 8080;
  Config.Protocol protocol() default Config.Protocol.https;

  public enum Protocol {http, https}
}
```

DS - Configuration: typed

```
@Activate
Config config;
```

OSGi-CDI - Configuration: typed

```
@Inject @ComponentProperties
Config config;
```

DS - Component

```java
@Component(
  configurationPid = {"foo", "bar"},
  configurationPolicy = REQUIRE)
public class FooImpl {
  ...
}
```

OSGi-CDI - Single Component

```java
@SingleComponent
@PID("foo")
@PID(value = "bar", policy = REQUIRED)
public class FooImpl {
  ...
}
```

OSGi-CDI - Factory Component

```java
@FactoryComponent("foo")
@PID("bar")
public class FooImpl {
  ...
}
```

The OSGi-CDI Spec

https://osgi.org/specification/osgi.enterprise/7.0.0/service.cdi.html

The OSGi-CDI Reference Implementation

https://github.com/apache/aries-cdi