

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 are DS and the vast majority will remain DS forever.

WHY use anything else?

by design DS is

... **ultra light weight**, and the DS annotations are merely convenient **CLASS retention annotations** processed at build time which means the runtime overhead is extremely low.

i.e. DS is not an annotation processor!

CDI

as part of its feature set, is...

an **extensible annotation processing** engine.

Custom annotations!

CDI

allows for...

completely internal wiring.

DS - Internal wiring: new

```
1 @Component
2 public class FooImpl {
3     private Pojo pojo;
4
5     public FooImpl() {
6         pojo = new PojoImpl();
7     }
8 }
```

CDI - Internal wiring: @Inject

```
1 public class FooImpl {  
2     private Pojo pojo;  
3  
4     @Inject  
5     public FooImpl(Pojo pojo) {  
6         this.pojo = pojo;  
7     }  
8 }
```


DS - Services: singleton

```
1 @Component
2 public class FooImpl implements Function {
3     ...
4 }
```

OSGi-CDI - Services: singleton

```
1 @Service
2 public class FooImpl implements Function {
3     ...
4 }
```

DS - Services: prototype

```
1 @Component(scope = PROTOTYPE)
2 public class FooImpl implements Function {
3     ...
4 }
```

OSGi-CDI - Services: prototype

```
1 @Service @ServiceInstance(PROTOTYPE)
2 public class FooImpl implements Function {
3     ...
4 }
```

DS - References

```
1 @Reference
2 Pojo pojo;
```

OSGi-CDI - References

```
1 @Inject @Reference  
2 Pojo pojo;
```

DS - Cardinality: mandatory

```
1 | @Reference  
2 | Pojo pojo;
```

OSGi-CDI - Cardinality: mandatory

```
1 @Inject @Reference  
2 Pojo pojo;
```


DS - Cardinality: optional

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

OSGi-CDI - Cardinality: optional

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

DS - Cardinality: multiple

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

OSGi-CDI - Cardinality: multiple

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

DS - Cardinality: at least one (or n)

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

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

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

DS - Reference Policy: greedy

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

OSGi-CDI - Reference Policy: reluctant

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


DS - Dynamic: mandatory

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

OSGi-CDI - Dynamic: mandatory

```
1 @Inject @Reference  
2 Provider<Pojo> pojo;
```

DS - Dynamic: multiple

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

OSGi-CDI - Dynamic: multiple

```
1 @Inject @Reference  
2 Provider<List<Pojo>> pojos;
```

DS - Dynamic: optional

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

OSGi-CDI - Dynamic: optional

```
1 @Inject @Reference  
2 Provider<Optional<Pojo>> pojo;
```

DS - OSGi Logger

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

OSGi-CDI - OSGi Logger

```
1 @Inject  
2 Logger logger;
```


DS - Configuration

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

OSGi-CDI - Configuration

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

Configuration Types

```
1 @Retention(RUNTIME)
2 @BeanPropertyType // OSGi-CDI
3 @ComponentPropertyType // DS
4 public @interface Config {
5     String hostname() default "localhost";
6     int port() default 8080;
7     Config.Protocol protocol() default
8     Config.Protocol.https;
9     public enum Protocol {http, https}
10 }
```

DS - Configuration: typed

```
1 @Activate  
2 Config config;
```

OSGi-CDI - Configuration: typed

```
1 @Inject @ComponentProperties  
2 Config config;
```

DS - Component

```
1 @Component(  
2     configurationPid = {"foo", "bar"},  
3     configurationPolicy = REQUIRE)  
4 public class FooImpl {  
5     ...  
6 }
```

OSGi-CDI - Single Component

```
1 @SingleComponent
2 @PID("foo")
3 @PID(value = "bar", policy = REQUIRED)
4 public class FooImpl {
5     ...
6 }
```

OSGi-CDI - Factory Component

```
1 @FactoryComponent("foo")
2 @PID("bar")
3 public class FooImpl {
4     ...
5 }
```


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>