LIFERAY DEV CON

OSGI CDI Integration Specification

Raymond Augé

<raymond.auge@liferay.com>

Why CDI in OSGi?

Reduce developer friction

Important Java specification

Benefit from extensive feature set

LIFERAY DEV CON

But Declarative Services (DS)?

We  DS!

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

But DS .... 

by design 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, private wiring.**

DS - Internal wiring - new

```
@Component
public class BeanImpl {

    Pojo pojo;

    BeanImpl() {
        pojo = new PojoImpl();
    }

}
```

CDI - Internal wiring - @Inject

```
public class BeanImpl {

    @Inject
    Pojo pojo;

}
```

DS - Services: singleton

```
@Component(service = BeanImpl.class)
public class BeanImpl {

    @Reference
    Pojo pojo;

}
```

CDI - Services: singleton

```
@Service
public class BeanImpl {

    @Inject
    @Reference
    Pojo pojo;

}
```

DS - Services: prototype

```
@Component(
        scope = ServiceScope.PROTOTYPE,
        service = BeanImpl.class
)
public class BeanImpl  {

        @Reference
        Pojo pojo;

}
```

CDI - Services: prototype

```
@Service
@ServiceInstance(ServiceScope.PROTOTYPE)
public class BeanImpl {

    @Inject
    @Reference
    Pojo pojo;

}
```

DS - References

```
@Component(service = BeanImpl.class)
public class BeanImpl  {

    @Reference
    Pojo pojo;

}
```

CDI - References

```
public class BeanImpl {

    @Inject
    @Reference
    Pojo pojo;

}
```

DS - Cardinality: mandatory

```
@Reference
Pojo pojo;
```

CDI - Cardinality: mandatory

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

DS - Cardinality: optional

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

CDI - Cardinality: optional

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

DS - Cardinality: multiple

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

CDI - Cardinality: multiple

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

DS - Cardinality: at least one (or n)

```
@Reference(
        cardinality =
                ReferenceCardinality.AT_LEAST_ONE)
List<Pojo> pojo;
```

CDI - Cardinality: at least one (or n)

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

DS - Reference Policy: greedy

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

CDI - Reference Policy: reluctant

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

GREEDY by default

DS - Dynamic: mandatory

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

CDI - Dynamic: mandatory

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

DS - Dynamic: multiple

```
@Reference(
        policy =
                ReferencePolicy.DYNAMIC)
volatile List<Pojo> pojoA;
```

CDI - Dynamic: multiple

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

DS - Dynamic: optional

```
@Reference(
        policy =
                ReferencePolicy.DYNAMIC,
        cardinality =
                ReferenceCardinality.OPTIONAL)
Pojo pojoA;
```

CDI - Dynamic: optional

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

DS - OSGi Logger

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

CDI - OSGi Logger

```
@Inject
Logger logger;
```

DS - Configuration

```
@Activate
void activate(Map<String, Object> props) {
    //
}

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

CDI - Configuration

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

CDI - Components

CDI Bundle

Container Component

```
@ApplicationScoped
@Service
class Controller {
 @Inject @Reference
 ConfigurationAdmin cm;
 @Inject DRManager brm;
 @Inject @Reference
 Optional<Pojo> pojo;
}
class DRManager {
 @Inject @Reference
 Provider<List<DR>> foo;
}
class Web {
 @Inject Controller ctlr;
}
```

Factory component

```
@FactoryComponent
@Service
class DR {}
```

```
@SingleComponent
class Pojo {
  @Inject @Reference
  Other other;
}
```

Single component

DS - Component

```
@Component(
    configurationPid =
        {"foo.pid", "bar.pid"},
    configurationPolicy =
        ConfigurationPolicy.REQUIRE
)
public class BeanImpl  {
    //...
}
```

CDI - Single Component

```
@SingleComponent
@PID("foo.pid")
@PID(
    value = "bar.pid",
    policy =
        ConfigurationPolicy.REQUIRED)
public class BeanImpl  {
    //...
}
```

CDI - Factory Component

```
@FactoryComponent("foo.pid")
@PID("bar.pid")
public class BeanImpl  {
    //...
}
```

Can I see the spec?

```
https://osgi.org/specification/osgi.enterprise/7
.0.0/service.cdi.html
```

Can I see the RI?

`https://github.com/apache/aries/tree/trunk/cdi`

Please rate this session in the Liferay Events App

Download on the App Store

ANDROID APP ON Google play