OSGi CDI Integration Specification

Raymond Augé - Sr. Software Architect

Why CDI In OSGi?

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

CDI - Features

as part of its feature set, is...

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

Custom annotations!

CDI - Internal wiring: @Inject

```java
import javax.inject.Inject;

public class PresenterImpl implements Presenter {
  private final Laptop laptop;

  @Inject
  public PresenterImpl(Laptop laptop) {
    this.laptop = laptop;
  }
}
```

CDI - Internal wiring: @Produces

```java
import javax.enterprise.inject.Produces;

@Produces Presentation presentation(
  LocalDate date, Presenter presenter, @Topic String topic,
  Details details) {

  return new Presentation.Builder(date)
    .withPresenter(presenter)
    .withTopic(topic)
    .withDetails(details)
    .build();
}
```

OSGi-CDI - Services: singleton

```java
import org.osgi.service.cdi.annotations.Service;

@Service
public class PresenterImpl implements Presenter {
    ...
}
```

OSGi-CDI - Services: prototype

```java
import org.osgi.service.cdi.annotations.Service;
import org.osgi.service.cdi.annotations.ServiceInstance;

@Service @ServiceInstance(PROTOTYPE)
public class Beer implements Drink {
  ...
}
```

OSGi-CDI - References to OSGi Services

```java
import javax.inject.Inject;
import org.osgi.service.cdi.annotations.Reference;

@Inject @Reference
Presenter presenter;
```

OSGi-CDI - References Cardinality: mandatory

```java
import javax.inject.Inject;
import org.osgi.service.cdi.annotations.Reference;

@Inject @Reference
Presenter presenter;
```

OSGi-CDI - References Cardinality: optional

```java
import javax.inject.Inject;
import org.osgi.service.cdi.annotations.Reference;

@Inject @Reference
Optional<Drink> drink;
```

OSGi-CDI - References Cardinality: multiple

```java
import javax.inject.Inject;
import org.osgi.service.cdi.annotations.Reference;

@Inject @Reference
List<Drink> drink;
```

... implies 0..n (multiple optional)

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

```java
import javax.inject.Inject;
import org.osgi.service.cdi.annotations.MinimumCardinality;
import org.osgi.service.cdi.annotations.Reference;

@Inject @Reference @MinimumCardinality(1)
List<Drink> drink;
```

OSGi-CDI - Reference Policy: reluctant

```java
import javax.inject.Inject;
import org.osgi.service.cdi.annotations.Reference;
import org.osgi.service.cdi.annotations.Reluctant;

@Inject @Reference @Reluctant
Presenter presenter;
```

… reference is Greedy by default

OSGi-CDI - References Dynamic: mandatory

```java
import javax.inject.Inject;
import org.osgi.service.cdi.annotations.Reference;

@Inject @Reference
Provider<Presenter> presenter;
```

OSGi-CDI - References Dynamic: multiple

```java
import javax.inject.Inject;
import org.osgi.service.cdi.annotations.Reference;

@Inject @Reference
Provider<List<Presenter>> presenters;
```

OSGi-CDI - References Dynamic: optional

```java
import javax.inject.Inject;
import org.osgi.service.cdi.annotations.Reference;

@Inject @Reference
Provider<Optional<Presenter>> presenter;
```

OSGi-CDI - Reference: target

```java
import javax.inject.Inject;
import org.osgi.service.cdi.annotations.Reference;

@Inject @Reference(target = "(service.vendor=Chicago JUG)")
List<Presenter> presenters;
```

OSGi-CDI - Reference: target `@BeanPropertyType`

```java
import javax.inject.Inject;
import org.osgi.service.cdi.annotations.Reference;
import org.osgi.service.cdi.propertytypes.ServiceVendor;

@Inject @Reference @ServiceVendor("Chicago JUG")
List<Presenter> presenters;
```

OSGi-CDI - Reference: prototype required

```java
import javax.inject.Inject;
import org.osgi.service.cdi.annotations.MinimumCardinality;
import org.osgi.service.cdi.annotations.PrototypeRequired;
import org.osgi.service.cdi.annotations.Reference;

@Inject @Reference @MinimumCardinality(1) @PrototypeRequired
List<Entry<Map<String, Object>, Drink>> drinks;
```

OSGi-CDI - Reference: any type

```java
import javax.inject.Inject;
import org.osgi.service.cdi.annotations.Reference;
import org.osgi.service.cdi.propertytypes.ServiceVendor;

@Inject @Reference(service = Reference.Any.class)
@ServiceVendor("Chicago JUG")
List<Object> all;
```

*... in support of **whiteboards***

OSGi-CDI - Reference: service events

```java
import javax.inject.Inject;

@Inject @ServiceVendor("Chicago JUG")
void monitorDrinks(BindServiceReference<Drink> drinks) {
  drinks
      .adding(this::doAdd)
      .modified(this::doModified)
      .removed(this::doRemoved)
      .bind();
}
```

OSGi-CDI - OSGi Logger

```java
import javax.inject.Inject;
import org.osgi.service.log.Logger;

@Inject
Logger video;
```

OSGi-CDI - Configuration

```java
import javax.inject.Inject;
import org.osgi.service.cdi.annotations.ComponentProperties;

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

OSGi-CDI - Configuration Types

```java
import org.osgi.service.cdi.annotations.BeanPropertyType;

@Retention(RUNTIME)
@BeanPropertyType
public @interface Details {
  String address();
  String instructions();
}
```

OSGi-CDI - Configuration: typed

```java
import javax.inject.Inject;
import org.osgi.service.cdi.annotations.ComponentProperties;

@Inject @ComponentProperties
Details eventDetails;
```

OSGi-CDI - Single Component

```java
import org.osgi.service.cdi.annotations.PID;
import org.osgi.service.cdi.annotations.SingleComponent;

@SingleComponent
@PID(value = "details", policy = REQUIRED)
@Service
public PresenterImpl implements Presenter {
  @Inject Laptop laptop;
  @Inject @ComponentProperties Details eventDetails;
}
```

OSGi-CDI - Factory Component

```java
import org.osgi.service.cdi.annotations.PID;
import org.osgi.service.cdi.annotations.FactoryComponent;

@FactoryComponent("registration")
@PID(value = "details", policy = REQUIRED)
public class AttendeeImpl implements Attendee {
  @Inject @ComponentProperties Registration registration;
  @Inject @ComponentProperties Details eventDetails;
}
```

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