

Eclipse4 Cheat Sheet

Legend

Get Help!

Dependencies

From existing data

```
@Active
```

Preferences

Instance scope: (Same programm but different settings)

```
prefs = InstanceScope.INSTANCE.getNode(<NAME>);
```

Configuration scope: (Same programm but same settings)

```
prefs = ConfigurationScope.INSTANCE.getNode(<NAME>);
```

Default scope: (Can not be changed. Config files)

```
prefs = DefaultScope.INSTANCE.getNode(<NAME>);
```

Eventbroker

Injecting the event service via:

```
@Inject  
private IEventBroker eventBroker;
```

Sending asynchronously:

```
eventBroker.post(<TAG>, <DATA>);
```

Sending synchronously:

```
eventBroker.send(<TAG>, <DATA>);
```

Receiving events on UI thread:

```
@Inject  
private void getNotified(@UIEventTopic(<TAG>)){}
```

Receiving events:

```
@Inject  
private void getNotified(@EventTopic(<TAG>)){}
```

Eclipse framework events:

```
org.eclipse.e4.ui.workbench.UIEvents.
```

Sends out objects from type:

```
org.osgi.service.event.Event.
```

Custom objects

Annotate custom objects with:

```
@Createable  
public class Object {}
```

Create objects in the app context, use:

```
@Createable  
@Singleton  
public class Object {}
```

Create object with IEclipseContext:

```
ContextInjectionFactory.make(Object.class, context);
```

Update

Commit

Publish