CDI @RequestScoped

Example cdi-request-scope can be browsed at https://github.com/apache/tomee/tree/master/examples/cdi-request-scope

This example show the use of `@RequestScoped` annotation for injected objects. An object which is defined as `@RequestScoped` is created once for every request and is shared by all the bean that inject it throughout a request.

Example

This example depicts a similar scenario to cdi-application-scope. A restaurant guest orders a soup from the waiter. The order is passed to the chef who prepares it and passes it back the waiter who in turn delivers it to the guest.

Waiter

The `Waiter` session bean receives a request from the test class via the `orderSoup()` method. A `Soup` insance will be created in this method and will be shared throughout the request with the `Chef` bean. The method passes the request to the `Chef` bean. It then returns the name of the soup to the test class.

```
@Stateless
public class Waiter {

    @Inject
    private Soup soup;

    @EJB
    private Chef chef;

    public String orderSoup(String name){
        soup.setName(name);
        return chef.prepareSoup().getName();
    }
}
```

Soup

The `Soup` class is an injectable POJO, defined as `@RequestScoped`. This means that an instance will be created only once for every request and will be shared by all the beans injecting it.

```
@RequestScoped
public class Soup {

    private String name = "Soup of the day";

    @PostConstruct
    public void afterCreate() {
        System.out.println("Soup created");
    }

    public String getName() {
        return name;
    }

    public void setName(String name){
        this.name = name;
    }
}
```

Chef

The `Chef` class is a simple session bean with an injected `Soup` field. Normally, the soup parameter would be passed as a `prepareSoup()` argument, but for the need of this example it's passed by the request context.

```
@Stateless
public class Chef {

    @Inject
    private Soup soup;

    public Soup prepareSoup() {
        return soup;
    }
}
```

Test Case

This is the entry class for this example.

```
public class RestaurantTest {

    private static String TOMATO_SOUP = "Tomato Soup";
    private EJBContainer container;

    @EJB
    private Waiter joe;

    @Before
    public void startContainer() throws Exception {
        container = EJBContainer.createEJBContainer();
        container.getContext().bind("inject", this);
    }

    @Test
    public void orderSoup(){
        String soup = joe.orderSoup(TOMATO_SOUP);
        assertEquals(TOMATO_SOUP, soup);
        soup = joe.orderSoup(POTATO_SOUP);
        assertEquals(POTATO_SOUP, soup);
    }

    @After
    public void closeContainer() throws Exception {
        container.close();
    }
}
```

Running

In the output you can see that there were two Soup instances created - one for each request.

```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running org.superbiz.cdi.requestscope.RestaurantTest
Apache OpenEJB 7.0.0-SNAPSHOT    build: 20111224-11:09
http://tomee.apache.org/
INFO - openejb.home = C:\Users\Daniel\workspaces\openejb\openejb\examples\cdi-request-
scope
INFO - openejb.base = C:\Users\Daniel\workspaces\openejb\openejb\examples\cdi-request-
scope
INFO - Using 'javax.ejb.embeddable.EJBContainer=true'
INFO - Configuring Service(id=Default Security Service, type=SecurityService,
provider-id=Default Security Service)
INFO - Configuring Service(id=Default Transaction Manager, type=TransactionManager,
provider-id=Default Transaction Manager)
INFO - Found EjbModule in classpath:
```

```
c:\Users\Daniel\workspaces\openejb\openejb\examples\cdi-request-scope\target\classes
INFO - Beginning load: c:\Users\Daniel\workspaces\openejb\openejb\examples\cdi-
request-scope\target\classes
INFO - Configuring enterprise application:
c:\Users\Daniel\workspaces\openejb\openejb\examples\cdi-request-scope
INFO - Configuring Service(id=Default Managed Container, type=Container, provider-
id=Default Managed Container)
INFO - Auto-creating a container for bean cdi-request-scope.Comp:
Container(type=MANAGED, id=Default Managed Container)
INFO - Configuring Service(id=Default Stateless Container, type=Container, provider-
id=Default Stateless Container)
INFO - Auto-creating a container for bean Chef: Container(type=STATELESS, id=Default
Stateless Container)
INFO - Enterprise application
"c:\Users\Daniel\workspaces\openejb\openejb\examples\cdi-request-scope" loaded.
INFO - Assembling app: c:\Users\Daniel\workspaces\openejb\openejb\examples\cdi-
request-scope
INFO - Jndi(name="java:global/cdi-request-
scope/Chef!org.superbiz.cdi.requestscope.Chef")
INFO - Jndi(name="java:global/cdi-request-scope/Chef")
INFO - Jndi(name="java:global/cdi-request-
scope/Waiter!org.superbiz.cdi.requestscope.Waiter")
INFO - Jndi(name="java:global/cdi-request-scope/Waiter")
INFO - Created Ejb(deployment-id=Chef, ejb-name=Chef, container=Default Stateless
Container)
INFO - Created Ejb(deployment-id=Waiter, ejb-name=Waiter, container=Default Stateless
Container)
INFO - Started Ejb(deployment-id=Chef, ejb-name=Chef, container=Default Stateless
Container)
INFO - Started Ejb(deployment-id=Waiter, ejb-name=Waiter, container=Default Stateless
Container)
INFO - Deployed
Application(path=c:\Users\Daniel\workspaces\openejb\openejb\examples\cdi-request-
scope)
Soup created
Soup created
INFO - Undeploying app: c:\Users\Daniel\workspaces\openejb\openejb\examples\cdi-
request-scope
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.412 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
```