

# Tapestry5SpringSessionScopedObjects

Spring can have some objects "session scoped", similarly to application state objects in tapestry.

These objects can be shared by both spring servlets and tapestry application.

You can create session scoped object by adding the following annotations to implementation class (or via xml config):

```
@Component
@Scope("session")
public class SessionDataImpl implements SessionData {
    ...
}
```

When you use such object in another class by

```
@Autowired
private SessionData sessionData;
```

spring ensures that each session get's it's own object instance for sessionData.

If you try to use such objects from tapestry with following annotations

```
@Inject
@Service("sessionDataImpl")
```

it will fail because spring needs to create proxy objects for accessing session scoped objects from global objects.

This can be resolved by adding string [RequestContextFilter](#) before [TapestryStringFilter](#) for tapestry urls, like this:

```
<filter>
  <filter-name>app</filter-name>
  <filter-class>org.apache.tapestry5.spring.TapestrySpringFilter</filter-class>
</filter>
<filter>
  <filter-name>requestContextFilter</filter-name>
  <filter-class>org.springframework.web.filter.RequestContextFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>requestContextFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
<filter-mapping>
  <filter-name>app</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```