

ARTIGO: 11485

Tomee + Spring + IoC + Jersey

Tive alguns problemas ao portar uma aplicação que rodava perfeitamente em um **Glassfish** para um **Tomee**. Alguns problemas relacionados a configurações de **WebServices REST FULL** usando **Jersey** e de injeção de dependência **CDI X Spring IoC**.

Quanto aos **WebServices**, apenas adicionei as seguintes linhas no **noconf/system.properties** do **Tomee**

```
openejb.api.javax.ws.rs.Path.validation=false  
com.sun.jersey.server.impl.cdi.lookupExtensionInBeanManager = true
```

Quanto a injeção de dependência do **Spring**, a anotação **@Autowired** parou de funcionar. Para resolver esse problema, precisei desabilitar a injeção do container (**CDI**), pois parecia estar conflitado com a injeção do **Spring IoC**. Para isso bastou remover o arquivo **beans.xml** da aplicação que os problemas relacionados a isso acabaram.

As exceção gerada, além de **NullPointerException** por conta do não funcionamento do **@Autowired**, era essa:

```
org.apache.webbeans.exception.inject.DeploymentException: javax.enterprise.inject.UnsatisfiedResolutionException: Api type [org.hibernate.SessionFactory] is not found with the qualifiers
```

Pelo que pude observar, o container, usando o **beans.xml** tentava configurar as classes anotadas com **@Inject** antes do **Spring** carregar o seu container, ou seja, antes de ler a configuração do **applicationContext.xml**. E como as classes dependiam de coisas configuradas no **Spring**, como a **SessionFactory** do **hibernate** dava erro de **UnsatisfiedResolutionException**;

E lembrando que a aplicação funcionava perfeitamente quando implantada no **Glassfish**.

Depois de várias horas quebrando a cabeça parece que consegui resolver!