Bab 3

Teknologi Injection Of Control (IoC) dengan Spring

3.1 Tujuan

Dalam bab ini, kita akan mendiskusikan bagaimana peran Spring sebagai Injection of Controll (IoC)controller didalam MVC sekaligus mendiskusikan perkembangan beberapa teknologi IoC dan mencoba mengimplementasikan teknik IoC ini kedalam sebuah MVC project.

Pada akhir bab ini, pelajar diharapkan dapat:

- Memahami peran IoC
- Memahami prinsip kerja Spring sebagai IoC
- Mengenal SpringIDE

3.2 Pengenalan IoC

IOC yang singkatan dari Injection of Control atau Dependency Injection, merupakan sebuah mekanisme yang secara awam adalah sebuah mekanisme memanggil sebuah objek tetapi tanpa inisialisasi.

Teknologi IOC sebenarnya telah lama ada di dunia Open Source Java, tetapi balik lagi, hanya beberapa yang bertahan, diantaranya Avalon dari Apache yang discountinue, terus ada JBoss Micro Kernel yang merupakan engine inti dari JBoss, atau yang sedang naik daun adalah Spring dari Interface21.

IOC ini sebenarnya menarik diikuti, dan untuk beberapa kasus sangat bagus digunakan dalam solusi, terutama mengisi interkoneksi antara M, V dan C dalam pemograman berbasis MVC. Yang mana dalam buku ini IOC lebih ditekankan pada integrasi antara WebWork dan Hibernate yang akan dibahas di Bab 5 sampai akhir buku ini.

Definisi IOC, untuk lebih lengkap dapat mengunjungi
http://martinfowler.com/articles/injection.html.

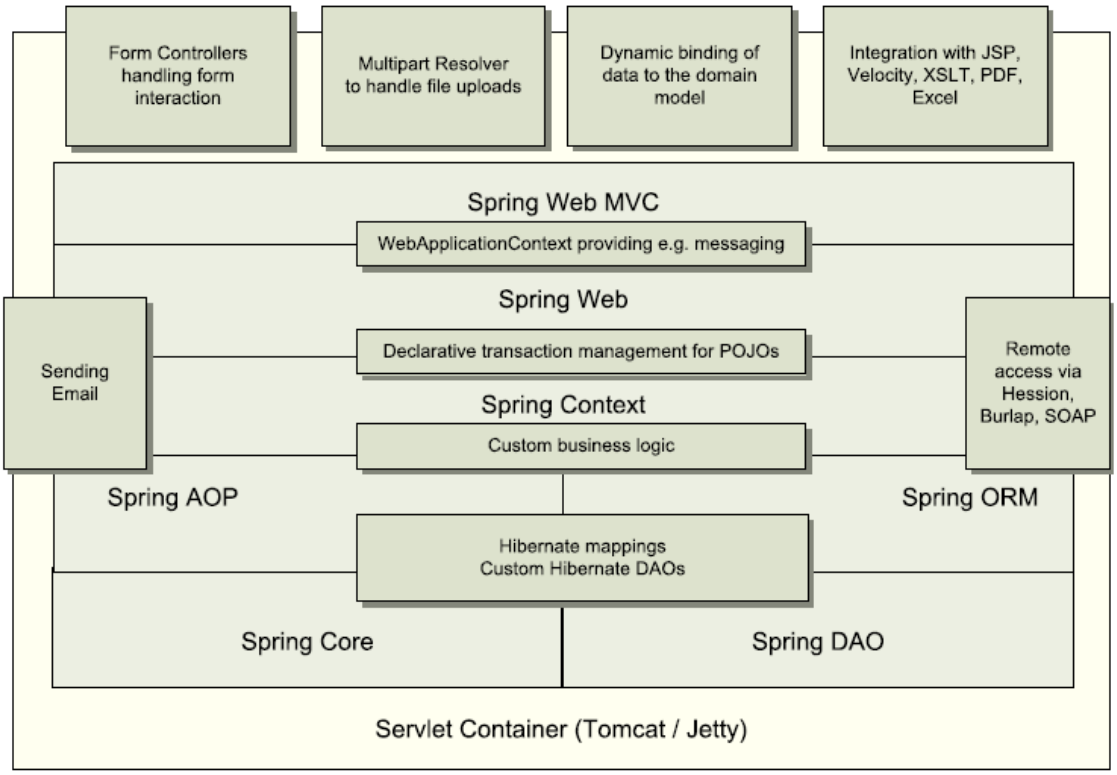
Dalam modul ini IOC diposisikan sebagai sebuah solusi yang menarik, terutama bagi mereka yang menjalankan solusi berbasis cimande pada JBoss. Artinya dapat dikatakan menggunakan dua teknologi IOC yaitu sebagai wrapper antara Controller dan Model dalam MVC, serta

sebagai container engine pada JBoss.

Reposisi ini, apalagi setelah JBoss sangat aktif untuk menjadi pemain utama dalam industri EJB3, yang mana EJB3, terkesan sangat Hibernate, walaupun sebenarnya untuk mengembangkan EJB3 dapat menggunakan saingan Hibernate seperti Toplink dari Oracle, atau Kodo dari Bea. Yang semuanya Open Source, kecuali TopLink comercial.

Dipasaran terjadi persaingan antara Spring melawan JBoss, sebab beberapa berita dari tim Spring, yang berhasil mengembangkan sebuah solusi yang lebih sering disebut dengan light development of Java, telah membuat Spring + Tomcat adalah solusi yang sangat ideal untuk production.

Sedangkan JBoss yang menggunakan JBoss Microkernel sebagai IOC engine, yang merupakan engine didalam JBoss AS dan juga wrapper solusi antara JSF dan Hibernate, dalam produk mereka bernama JBoss Seam. Membuat posisi JBoss AS + JBoss Seam mirip dengan Spring stack.



Arsitektur Spring

Dalam modul ini sebenarnya lebih menekankan integrasi framework MVC antara WebWork/Struts 2 dengan Hibernate menggunakan Spring. Dimana Seam menggunakan JSF dan Hibernate. Sedangkan Alfresco menggunakan JSF, Hibernate dengan Spring.

Semua merupakan kombinasi yang berbeda, dan tinggal Anda menentukan solusi mana yang lebih sesuai, dimana sebenarnya ke-3 pilihan solusi ini dapat diintegrasikan, yaitu dengan merubahnya menjadi solusi berbasis SOA.

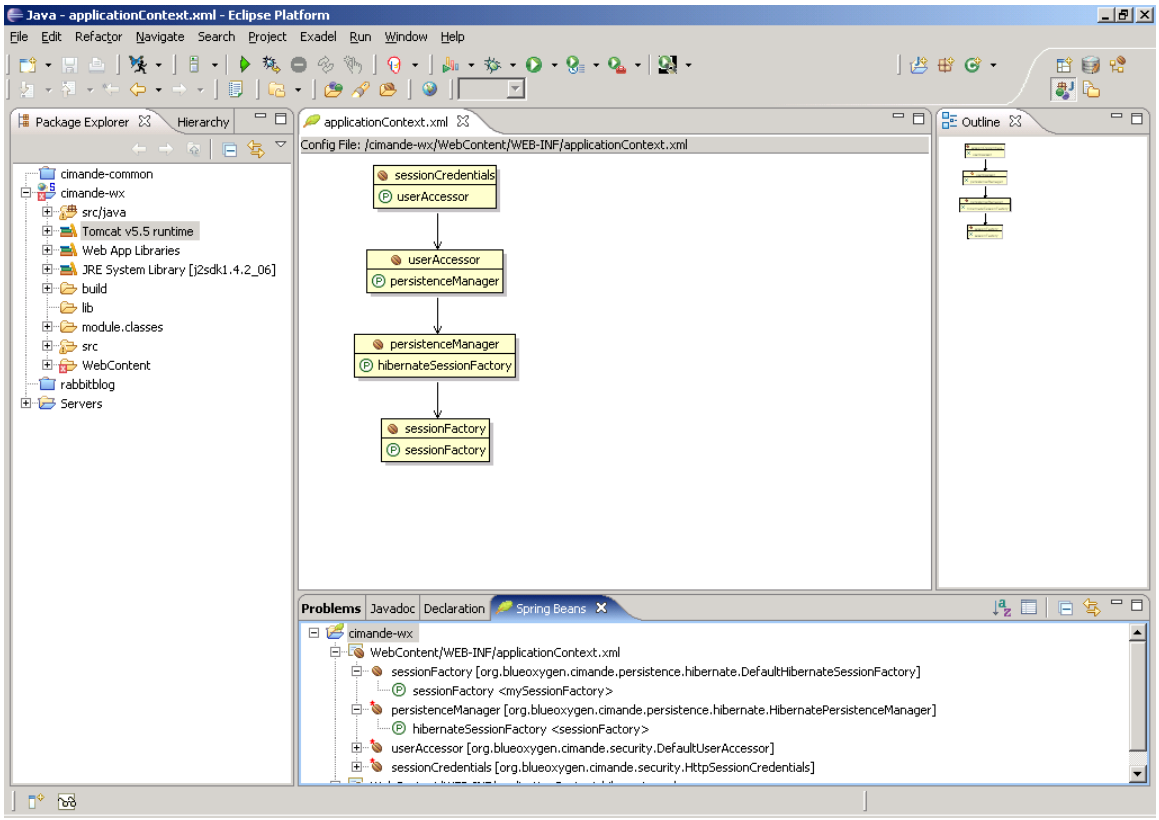
Reposisi Spring yang berawal dari sebuah IOC (Spring Bean) menjadi sebuah solusi yang lengkap membuat Tomcat + Spring memiliki teknologi yang sama dengan JBoss AS. Padahal teknologi Spring ini berjalan diatas JBoss.

Bentuk persaingan ini mirip dengan persaingan Windows dengan Java, yang saat ini terjadi. Kita tahu Java berjalan diatas Windows.

3.3 Sekilas tentang SpringIDE

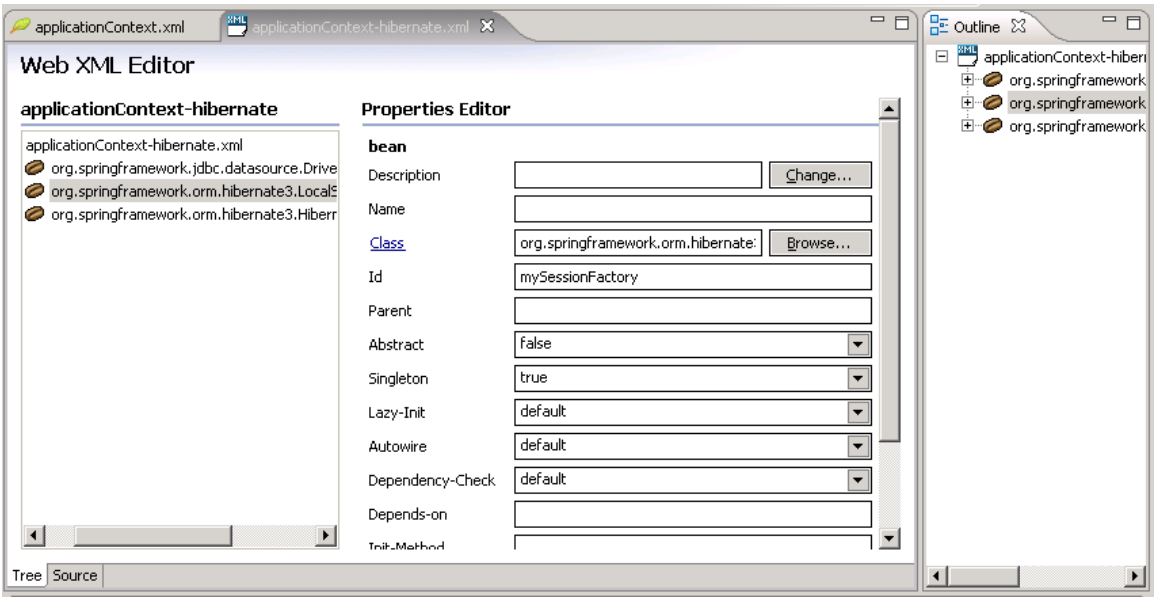
SpringIDE adalah sebuah plugins dari Eclipse. SpringIDE memungkinkan applicationContext.xml divisualisasikan dengan baik.

Fitur yang diperlukan dalam pengembangan IOC adalah adanya grafik visualisasi dari application contextnya spring yang dapat dilihat dibawah ini. Grafik ini membantu sekali untuk mengetahu hubungan antara bean didalam Spring Bean.



Eclipse Calisto yang telah terinstall SpringIDE

Penulis menambahkan Exadel 4.0 kedalam Eclipse Calisto, untuk menggunakan XML editor dari Exadel yang lebih baik daripada XML Editornya Calisto.



ApplicationContext Spring yang dibuka dengan XML Editor dari Exadel.

3.3.1 Menginstall SpringIDE

Untuk menginstal SpringIDE, terdapat sedikit trik untuk diikuti terutama untuk menginstall GEF 1.0 SDK secara manual. Setelahnya hanya memerlukan menambahkan lokasi dari SpringIDE untuk didownload secara kedalam Eclipse.

3.3.2 Bekerja dengan SpringIDE

Berikut adalah application Context projek Cimande, yang mengimplementasikan IOC untuk Hibernate Session Factory dan SessionCredential untuk implementasi pada Security Filter.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN"
"http://www.springframework.org/dtd/spring-beans.dtd">

<beans >

        <bean id="sessionFactory"
class="org.blueoxygen.cimande.persistence.hibernate.DefaultHibernateSessionFactory"
        init-method="init"
        destroy-method="destroy"
        singleton="true">
                <property name="sessionFactory" ref="mySessionFactory"/>
    </bean>

        <bean          id="persistenceManager"
class="org.blueoxygen.cimande.persistence.hibernate.HibernatePersistenceManager"
                init-method="init"
                destroy-method="dispose"
                singleton="false">
        <property name="hibernateSessionFactory" ref="sessionFactory"/>
    </bean>
```

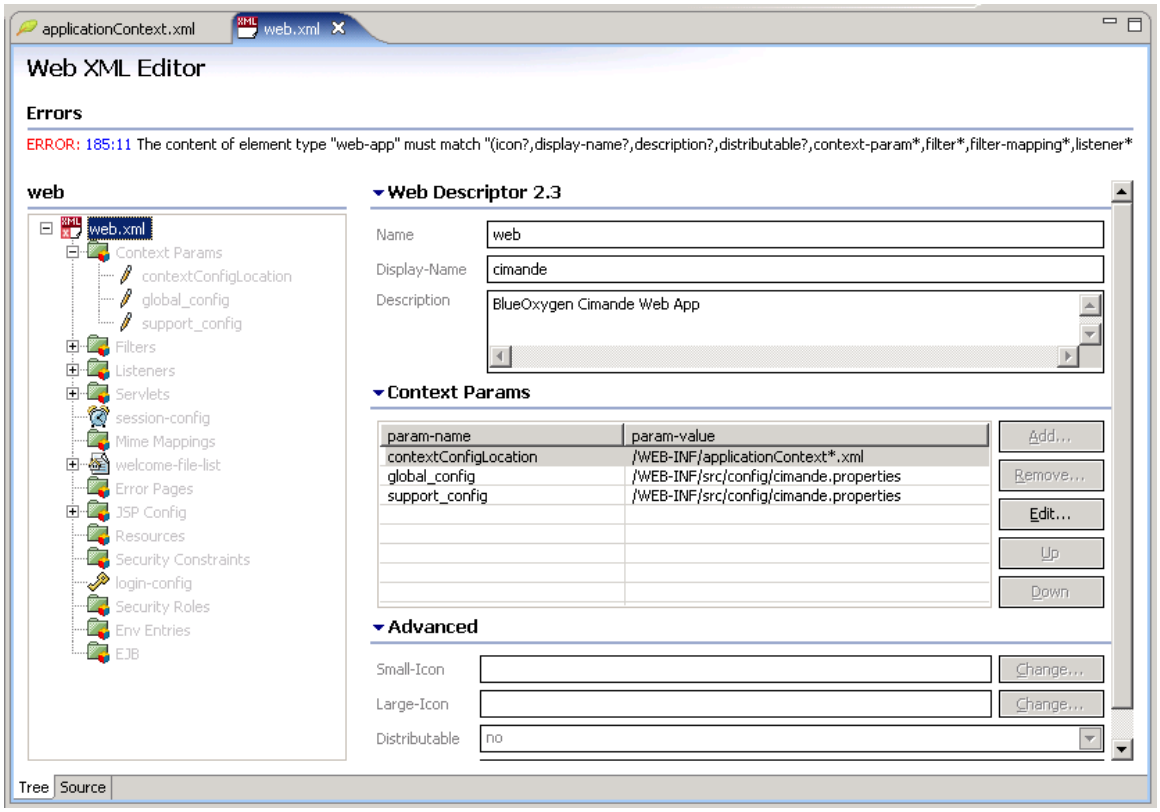
```
        <bean        id="userAccessor"
                     class="org.blueoxygen.cimande.security.DefaultUserAccessor"
                     singleton="false">
                <property name="persistenceManager"><ref
bean="persistenceManager"/></property>
        </bean>

    <bean       id="sessionCredentials"
                class="org.blueoxygen.cimande.security.HttpSessionCredentials"
                singleton="false">
                <property name="userAccessor"><ref bean="userAccessor"/></property>
    </bean>


</beans>
```

Lokasi dari applicationContext.xml diatas berada pada folder WEB-INF. ApplicationContext.xml ini yang akan menjadi kasus yang akan aktif secara otomatis saat Java EE container diaktifkan.

Dimana settingnya terdapat pada file web.xml pada folder WEB-INF juga. Inilah tag xmlnya untuk mengaktifkan applicationContext.xml didalam sebuah projek Java.

```
<context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/applicationContext*.xml</param-value>
</context-param>
```

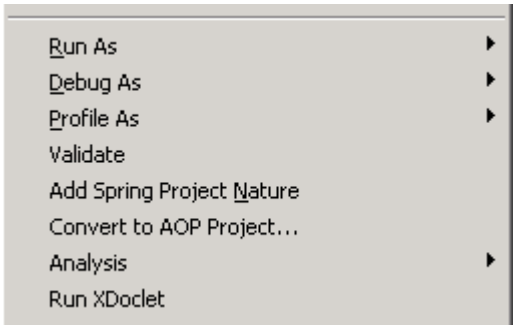
Atau bilamana melihat melalui tree viewnya XML Editor adalah sebagai berikut



Jadi jangan kaget bilamana setelah bab ini, semua kegiatan persistence yang digunakan oleh buku ini yaitu Hibernate secara otomatis aktif, dan beberapa keajaiban muncul. Inilah kehebatan Injection of Control.

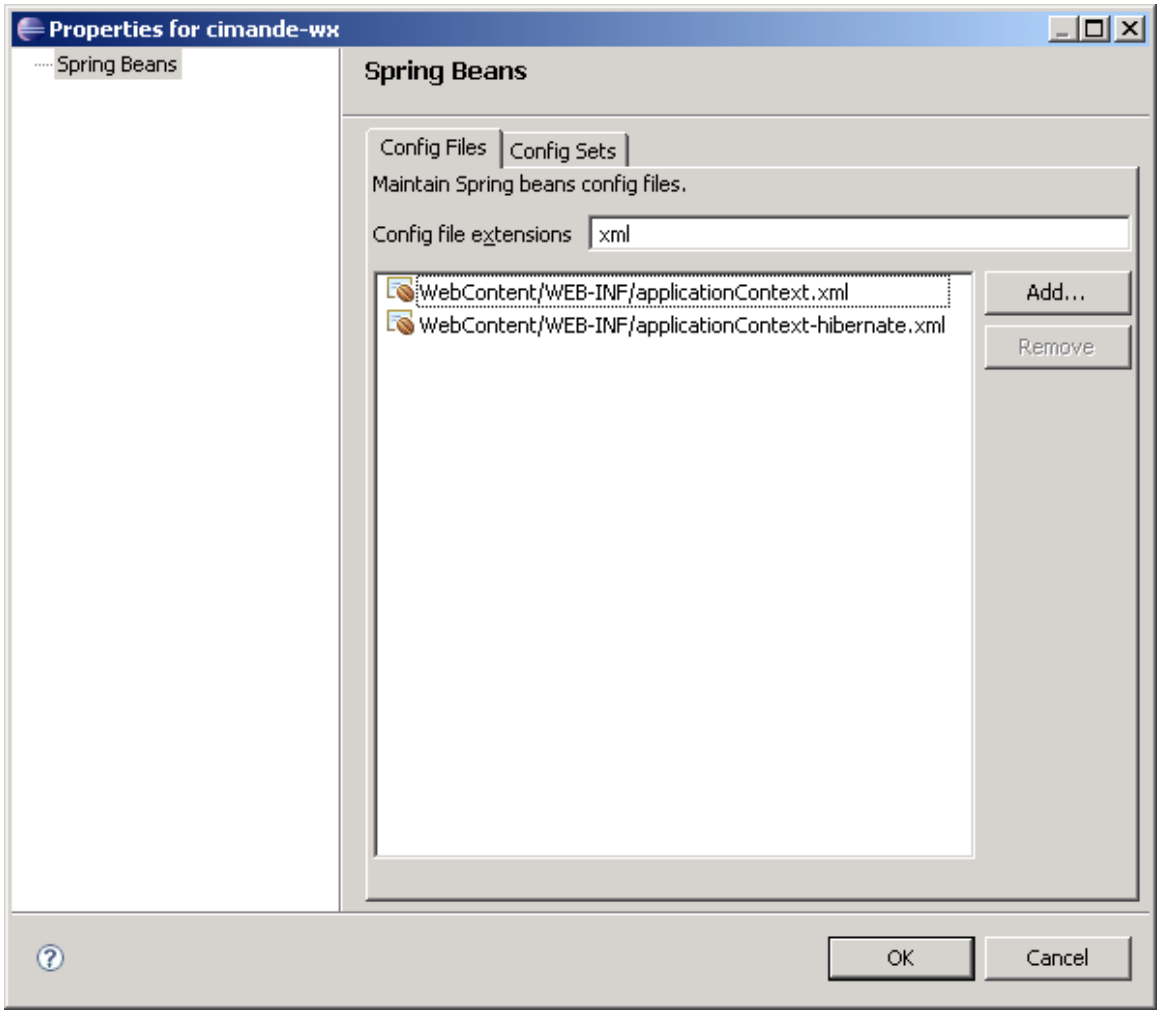
Untuk membuat sebuah projek Eclipse mendukung Spring adalah dengan membuat projek yang dibuka diakui sebagai projek berbasis Spring. Tanpa setting ini, view Spring Beans tidak dapat digunakan.

Caranya dengan meklik kanan nama projek yang akan disetting, lalu menambahkan Spring Project Nature kedalam projek tersebut.



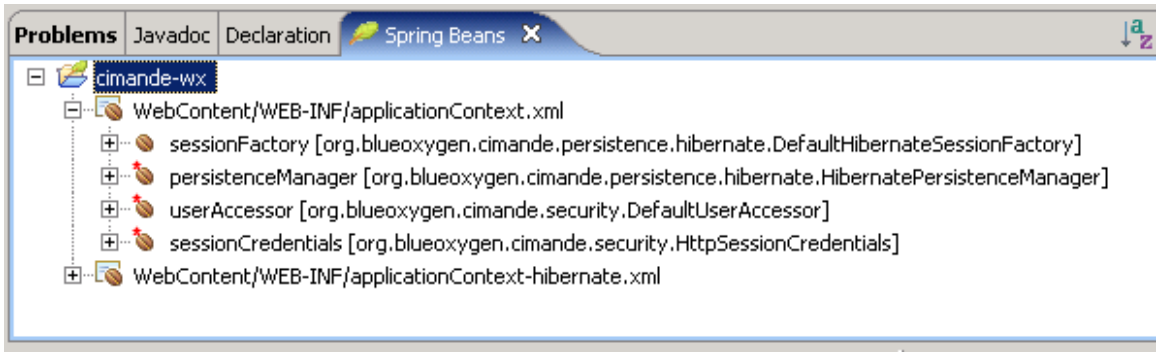
Bilamana telah diset sebagai Spring Project Nature, maka projek Java tersebut otomatis akan dianggap projek Spring.

Untuk membuat view Spring Beans dapat digunakan, tambahkan applicationContext.xml kedalam dialog boxnya.



XML ApplicationContext yang dimasukan kedalam Spring Beans

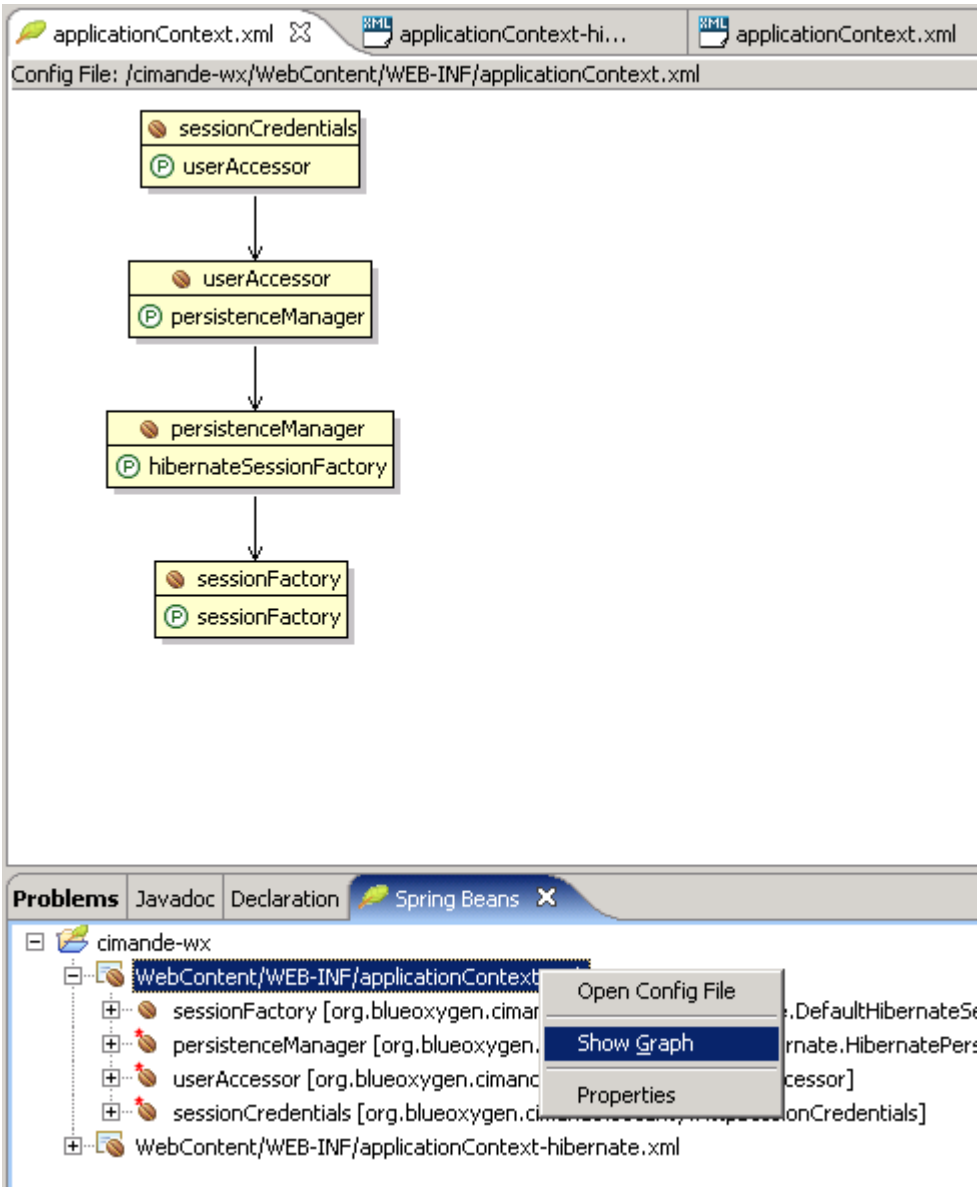
Spring Bean mendukung lebih dari satu XML. Bilamana telah selesai maka didalam view Spring Beans, semua bean akan tervisualisasi.



ApplicationContext dalam view Spring Beans.

Untuk melihat hubungan antara bean didalam application contextnya Spring dapat mengklik kanan terhadap applicationContext yang diinginkan.

Maka grafik hubungan bean akan muncul



Harap diperhatikan, bilamana melakukan klik 2x, maka yang muncul adalah XML Editor, sedangkan bilamana klik kanan, dapat memilih antara Show Graph untuk melihat diagram hubungan, dan membuka dengan XML Editor.

Berikut adalah diagram hubungan untuk applictionContext-hibernate.xml, yang digunakan projek Cimande untuk mengaktifkan persistence Hibernate.

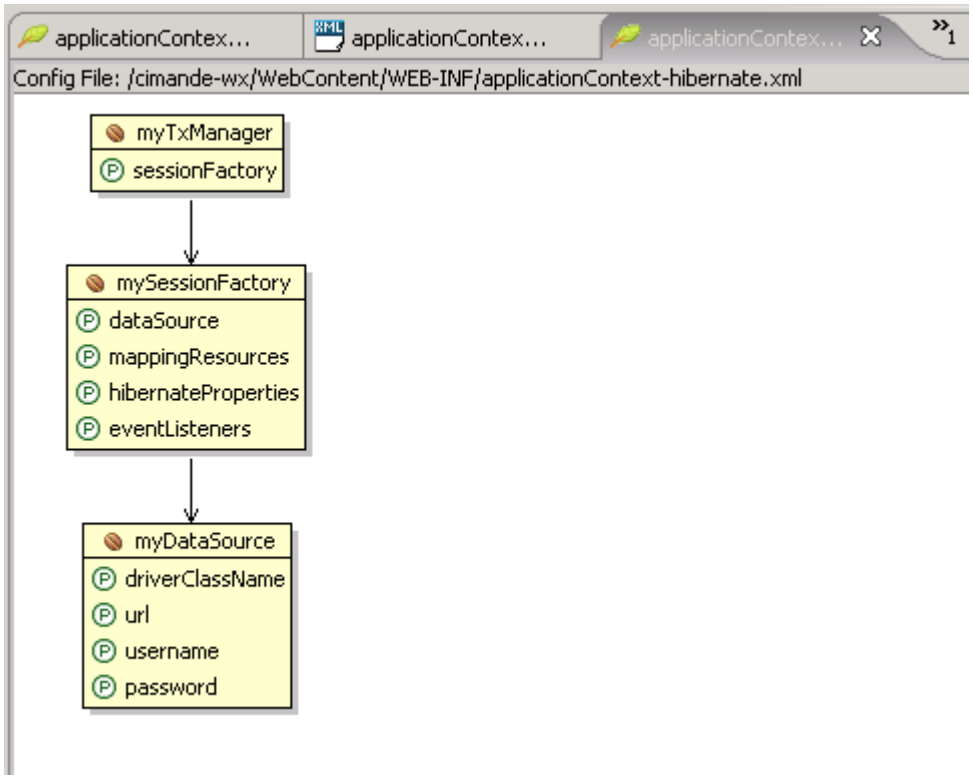
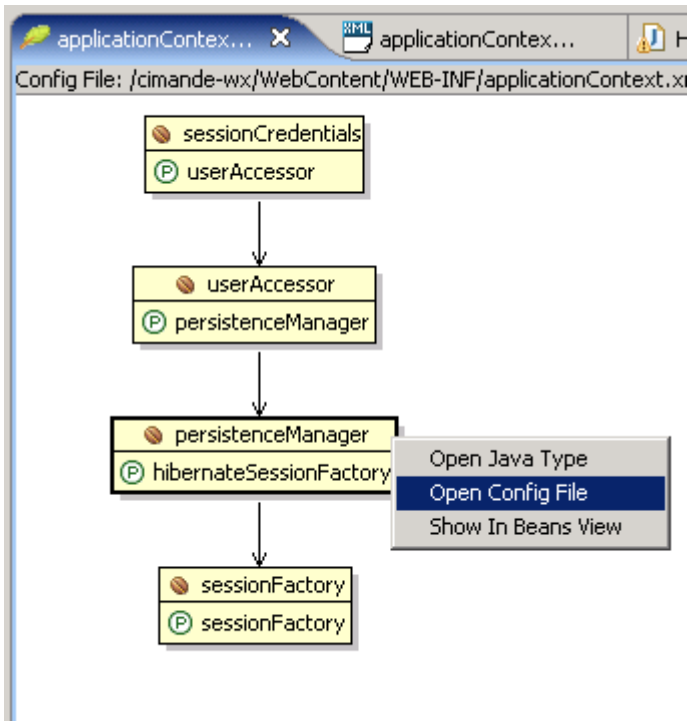


Diagram hubungan pada SpringIDE, bilamana diklik 2x, maka akan mengaktifkan XML Editor, sedangkan untuk mengetahui dimana sebuah bean didalam applicationContext didaftarkan, harus mengklik kanan.



Bilamana klik kanan telah dilakukan ada 3 pilihan yaitu Open Java Type, Open Config File, dan Show In Beans View.

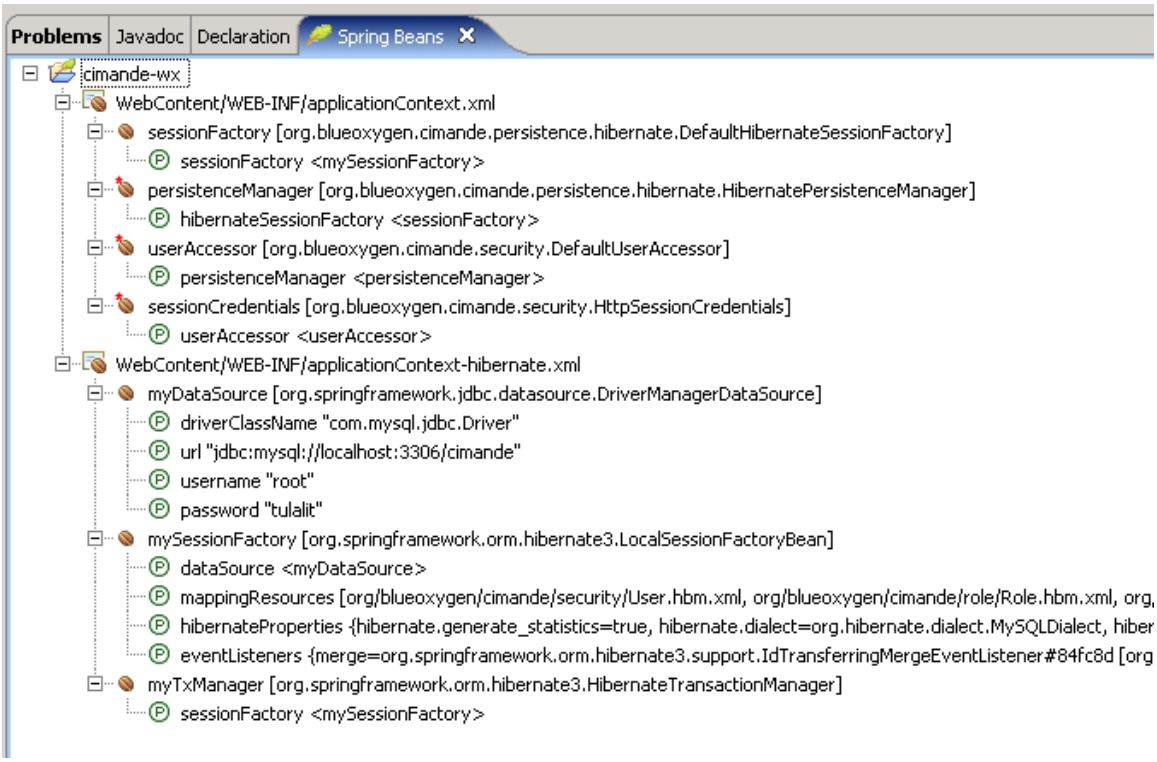
Open Java Type adalah kegiatan yang sama dengan mengaktifkan Java Editor, sedangkan Open Config File adalah membukanya dengan XML Editor, sedangkan Show In Beans View, adalah melink dengan view Spring Beans.

3.3.3 Hubungan Hierarki dalam Spring Beans.

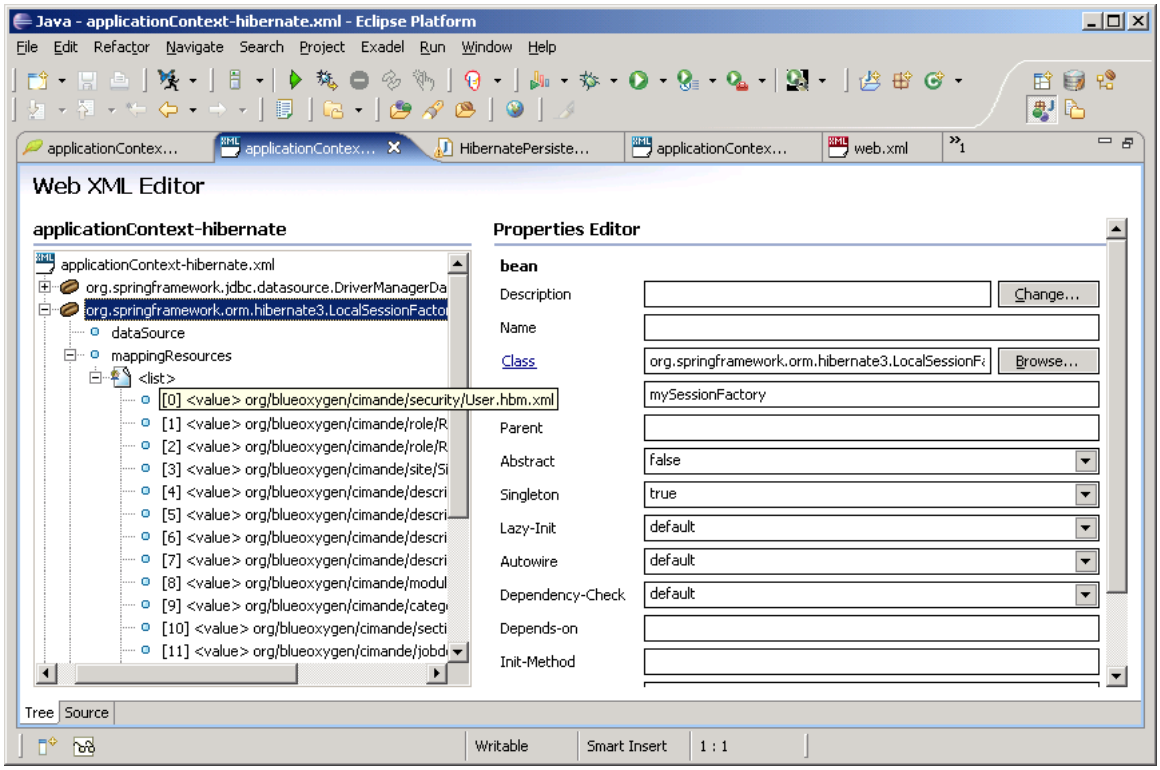
Application context Spring adalah sebuah XML, sehingga dalam implementasinya memungkinkan dilakukan hubungan antara satu bean dengan bean lain didalamnya.

Sayang sekali, belum ada editor untuk propertis didalam applicationContext, sehingga untuk melihat lebih jelas isi dari sebuah beans ini harus menggunakan XML Editor.

Berikut adalah drill down semua applicationContext yang digunakan oleh projek Cimande.



Sedangkan bilamana org.springframework.orm.hibernate3.LocalSessionFactoryBean dipilih maka didalam XML Editor kita dapat melihat apa saja yang ada didalamnya.



Cobalah geser mouse kedalam <list> yang ada, maka kita dapat melihat isi dari propertiesnya.

<list> sebenarnya adalah kumpulan parameter didalam propertiesnya applicationContext.

Berikut adalah isi dari list didalam applicationContext-hibernate.xml

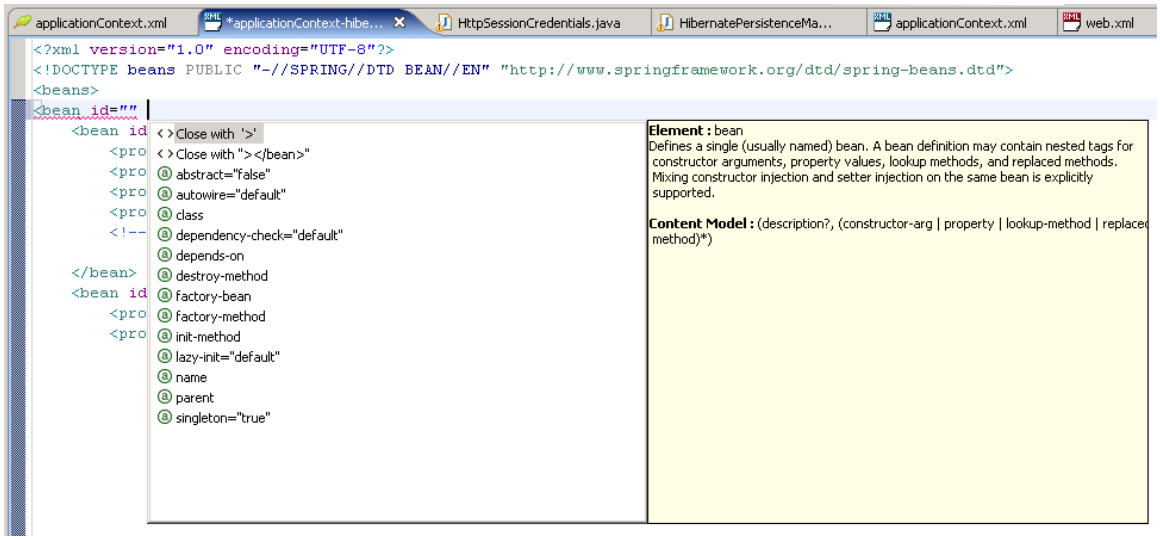
```
<list>
        <value>org/blueoxygen/cimande/security/User.hbm.xml</value>
        <value>org/blueoxygen/cimande/template/TemplateObjectDetail.hbm.xml</value>

        ...
        <value>org/blueoxygen/cimande/template/Template.hbm.xml</value>
</list>
```

Terdapat 17 mapping Hibernate didalamnya. Mekanisme applicationContext-hibernate sebenarnya adalah pilihan didalam implementasi pada Cimande, apakah hendak menggunakan hibernate.cfg.xml, atau menginject hibernate.cfg.xml kedalam applicationContext.

3.3.4 Bekerja dengan XML Editor

Spring IDE yang terintegrasi dengan baik dengan XML Editor didalam Eclipse, walaupun XML Editor yang dipakai penulis adalah XML Editor dari Exadel, ternyata tag-tag xml didalam SpringIDE telah terintegrasi.

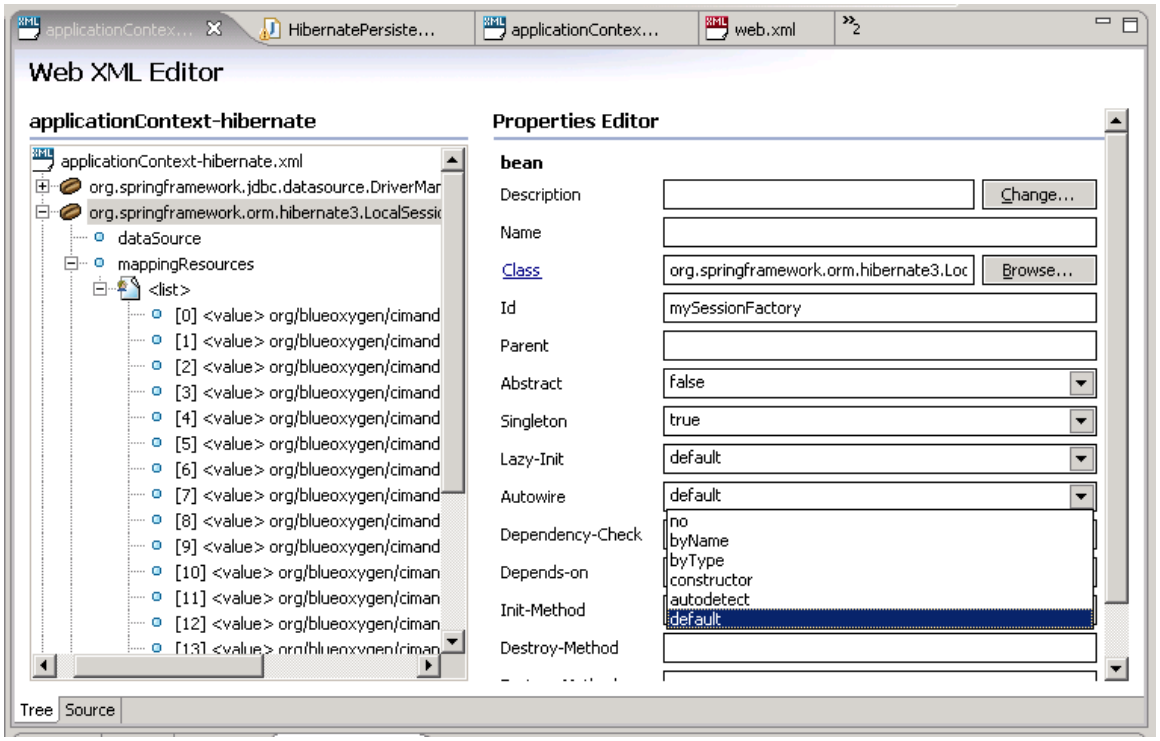
Berikut adalah sebuah dialog yang muncul bilamana kita hendak membuat sebuah <bean> baru.



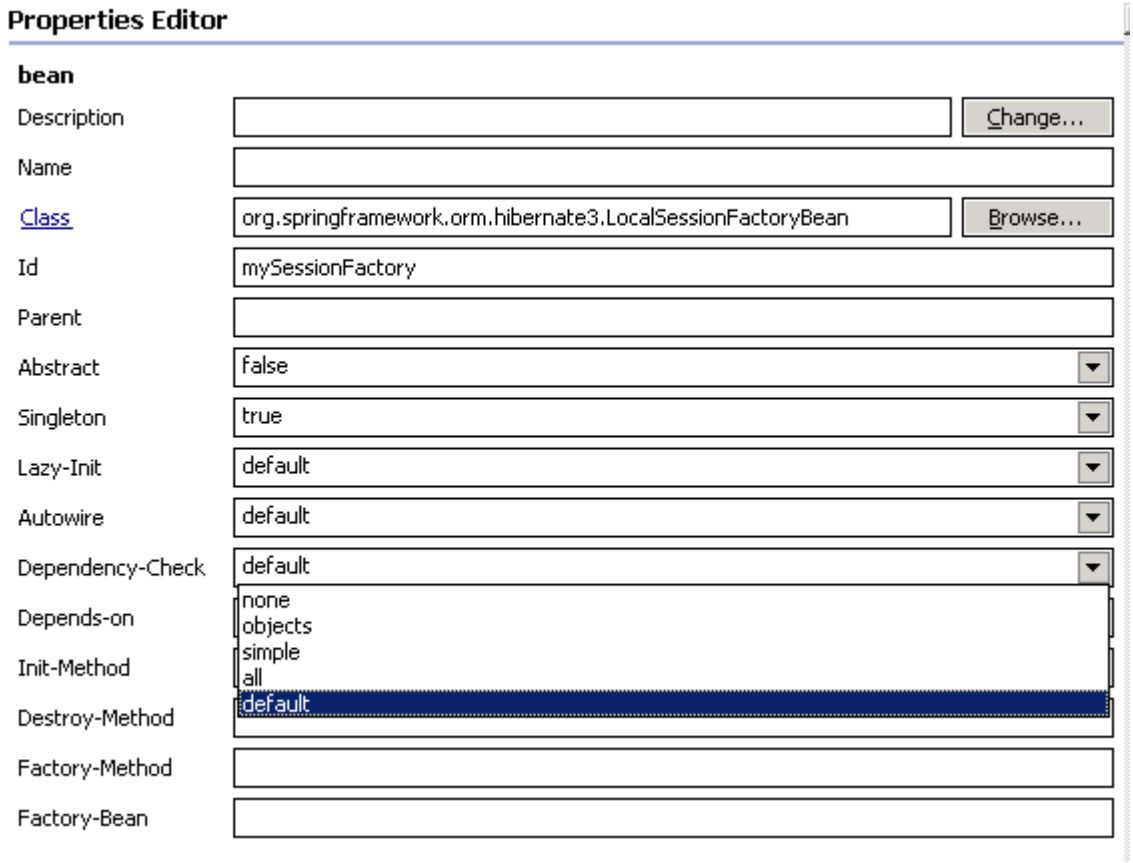
Sebenarnya rahasia semua ini terdapat pada satu tag yang tertulis diatas application context.

```
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN"
"http://www.springframework.org/dtd/spring-beans.dtd">
```

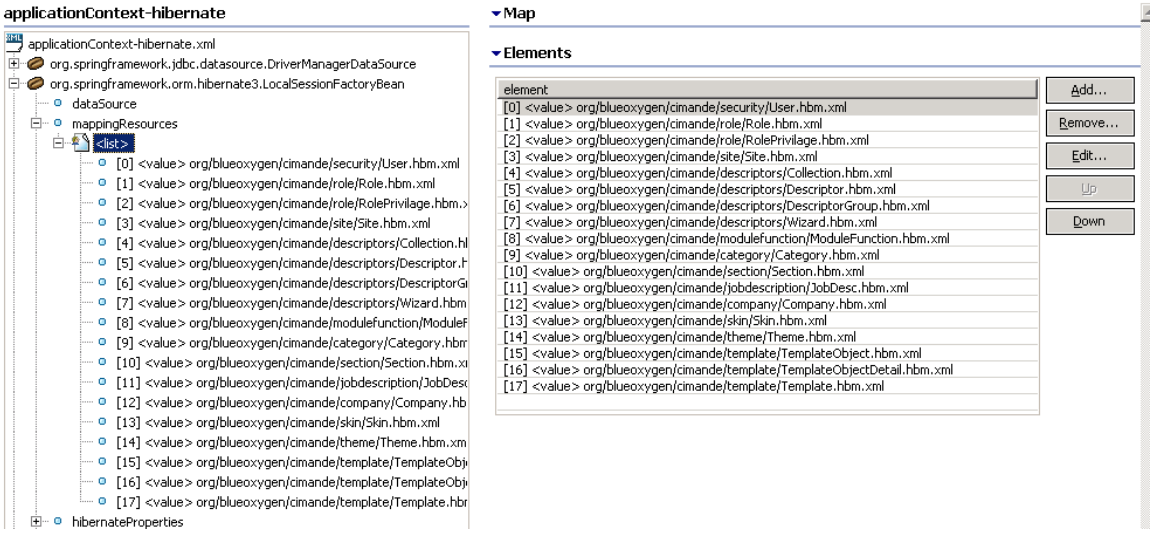
Bilamana telah membuat sebuah bean didalam application context yang terbuka, dapat pindah ke modus “tree”, dan secara otomatis semua isian dari tag-tag xml aktif. Kita dapat mengganti semua isian dari namespace didalam XML yang ada.



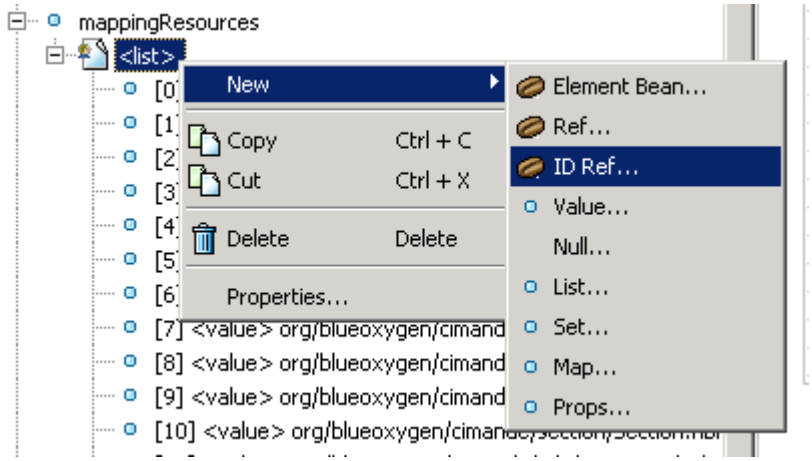
Berikut adalah properties editor yang ada didalam XML Editor.

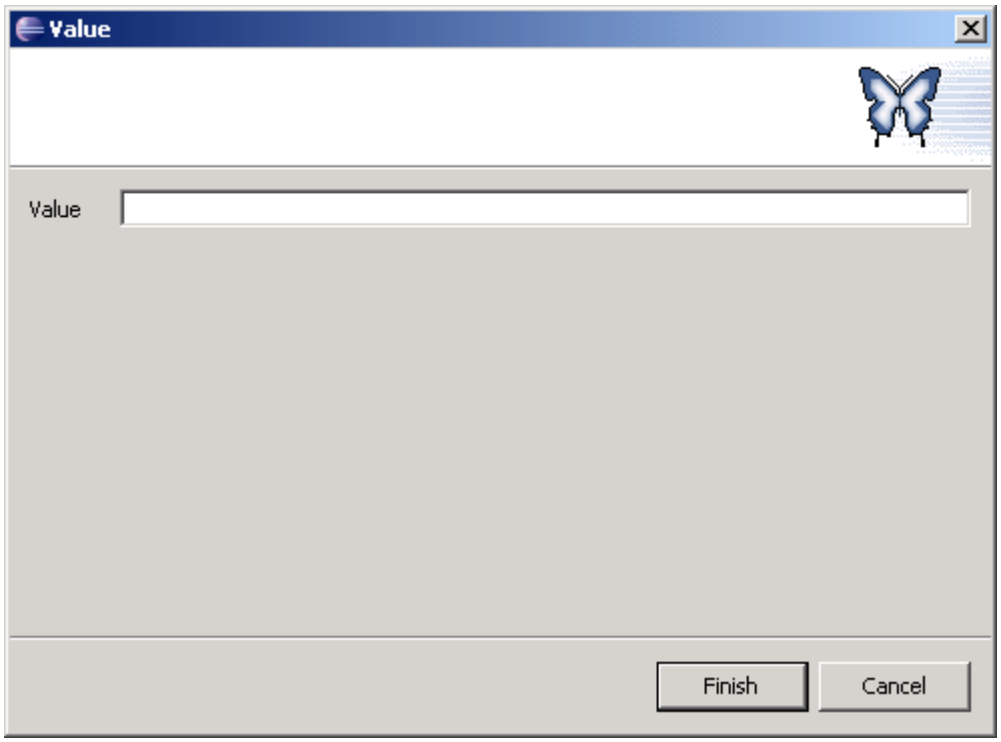


Bilamana implementasinya adalah berbentuk list, maka akan muncul sebuah element grid yang berisikan value didalam list tersebut.

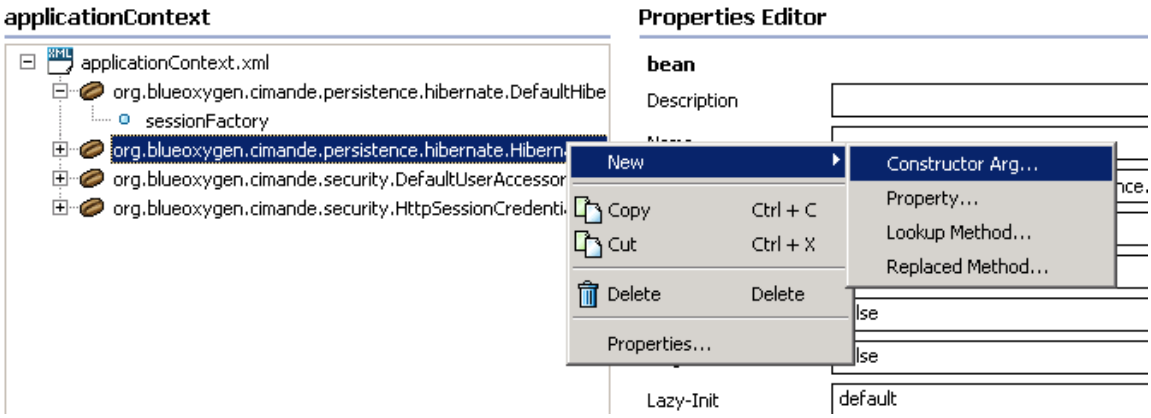


Bilamana kita hendak menambahkan value didalam list tersebut tinggal klik kanan, maka akan muncul dialog box.





Sedangkan bilamana kita mengklik kanan langsung dari beannya akan muncul dialog untuk membuat Constructor, Property, Lookup Method atau Replaced Method.



Sayang sekali semua implementasi dari New didalam Spring IDE ini hanya seperti isian value.

Untuk lebih jelas bagaimana cara kerja semua isian ini, dapat melihat pada DTD dari Spring.