http://www.egovframe.go.kr/wiki/doku.php?id=egovframework:rte3:fdl:property_source

Property Source

1. Outline

Property Source is loaded by the keys in the bean configuration on XML, using property place-holder or, in the coding, “Environment”.

Property source file offers additional configurations to load the attribute value on the DB table, in the form of PropertySource. Definition of PropertySource can also be customized by user.

Property-placeholder and PropertySource

Property-placeholder

Definition of bean in ${…} can be replaced by using property placeholder.
Refer to the following for coding example:

```
<context:property-placeholder location="com/bank/config/datasource.properties"/>

<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close">
	<property name="driverClass" value="${database.driver}"/>
	<property name="jdbcUrl" value="${database.url}"/>
	<property name="username" value="${database.username}"/>
	<property name="password" value="${database.password}"/>
</bean>
```

Before Spring 3.1, you need to use PropertyPlaceholderConfigurer by defining <context:property-placeholder>.
Spring 3.1 intrinsically comprise PropertySourcesPlaceholderConfigurer and allows ${database.*} to be found on the attribute of Environment when the datasource.properties cannot be used.

PropertySource is accessibly by way of Environment, signifying that the user-defined PropertySource can be used from Spring 3.1 on, using property-placeholder.

User-defined PropertySource

In Spring 3.1, users may come into play to define PropertySource. The definition can be made by working the ApplicationContextInitializer interface, web.xml contextInitializerClasses, servlet context and Environment.

Implementation of ApplicationContextInitializer interface may cause the initialization logic of ApplicationContext realized, by registering the implementation class in the servlet context parameter.

Refer to the following for how to define web.xml:

```
<context-param>
	<param-name>contextInitializerClasses</param-name>
	<param-value>com.bank.MyInitializer</param-value>
</context-param>
```

Example of MyInitializer for implementation class of ApplicationContextInitializer is as follows:

```
public class MyInitializer implements ApplicationContextInitializer<ConfigurableWebApplicationContext> {
	public void initialize(ConfigurableWebApplicationContext ctx) {
		PropertySource ps = new MyPropertySource();
		ctx.getEnvironment().getPropertySources().addFirst(ps);
```

```
        // perform any other initialization of the context ...
    }
}
```

With the foregoing registration, you can find the implementor of ApplicationContextInitializer is initizliaed when ApplicationContext is loaded or refreshed. The user-defined PropertySource(where PropertySource is inherited) is to be added onto the attributeSources in Environment (using getPropertySources).

The user defined PropertySource loads the attribute by way of Environment or property-placeholder.

DB PropertySource

In eGovFramework 3.0, you can take advantage of DBPropertySource to load the attribute value out of DB Table.

Configurations for using DB PropertySource

Configurations for using DB PropertySource are as follows:

- Generating DB TABLE and data: Generate DB TABLE with the title of columns being PKEY and PVALUE
- Defining BEAN : Configure XML where dataSource and dbPropertySource are defined as beans.
- Configuring web.xml : Configure xml path and DBPropertySourceInitializer as defined above, in web.xml.

Configuring DB

Connect DB when WAS is in operation, by loading XML to load the attribute values off the table. First thing you need to do so is to generate a table for DB property, with the title of column being PKEY and PVALUE:

```
CREATE TABLE PROPERTY (
   PKEY VARCHAR(20) NOT NULL    PRIMARY KEY ,
   PVALUE VARCHAR(20) NOT NULL
);

commit;
INSERT INTO PROPERTY (PKEY, PVALUE) VALUES ('egov.test.sample01', 'db-property-sample01');
INSERT INTO PROPERTY (PKEY, PVALUE) VALUES ('egov.test.sample02', 'db-property-sample02');

...

commit;
```

Refer to the following example for how to configure db for xml:

```
<beans xmlns="http://www.springframework.org/schema/beans" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                        xmlns:p="http://www.springframework.org/schema/p"
                        xmlns:jdbc="http://www.springframework.org/schema/jdbc"
                        xmlns:context="http://www.springframework.org/schema/context"
                        xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.2.xsd
                                http://www.springframework.org/schema/jdbc
http://www.springframework.org/schema/jdbc/spring-jdbc-3.2.xsd">

        <jdbc:embedded-database id="dataSource" type="HSQL">
                <jdbc:script location="classpath:db/ddl.sql" />
                <jdbc:script location="classpath:db/dml.sql" />
        </jdbc:embedded-database>
```

```
        <bean id="dbPropertySource" class="egovframework.rte.fdl.property.db.DbPropertySource">
                <constructor-arg value="dbPropertySource"/>
                <constructor-arg ref="dataSource"/>
                <constructor-arg value="SELECT PKEY, PVALUE FROM PROPERTY"/>
        </bean>
</beans>
```

Configuring web.xml

You'll then need to configure loading of DB value when WAS is in operation. Add DBPropertySourceInitializer as provided by egov and configure the path of xml like you previously did.

```
<context-param>
    <param-name>contextInitializerClasses</param-name>
    <param-value>egovframework.rte.fdl.property.db.initializer.DBPropertySourceInitializer</param-value>
  </context-param>
  <context-param>
    <param-name>propertySourceConfigLocation</param-name>
    <param-value>classpath:/initial/propertysource-context.xml</param-value>
  </context-param>
```

Accessing DB PropertySource

Using DBPropertySource in xml

In order to use PropertySource defined in xml, you need to configure property-placeholder as follows:

```
...
<context:property-placeholder/>

<!-- Configuring Message Source Bean -->
<bean id="propertyTest" class="egov.sample.property.PropertyTest">
  <property name="sample01" value="${egov.test.sample01}"/>
  <property name="sample02" value="${egov.test.sample02}"/>
</bean>
...
```

Using DBPropertySource in Code

You'll need to use Environment Abstraction in eGovFramework 3.0 to be granted an access to PropertySource. See "Environment" section for more information.

4. References

- Spring 3.1 M1: Unified Property Management