

Dependency Injection



This section is out of date. BigSense has moved off of Spring for Dependency Injection and AOP. It now has Dependency Injection built into standard Scala code and AOP via AspectJ at compile time. This section will be updated soon to reflect these changes. For now, the source code for the Dependency Injection can be found here:

<https://github.com/BigSense/BigSense/blob/master/src/main/scala/io/bigsense/spring/MySpring.scala>

Spring Dependency Injection is used to tie all the individual [components](#) together. To learn more about Spring Dependency Injection, please read the chapter on Dependency Injection in *Spring in Action* published by Manning. In our Application Context, the *name* and *id* fields are identical for every bean. The following examples are taken from the Application Context in the current version found in [BigSense](#) at the time of this writing. They may not be up to date. Please view the latest version at the following URL:

<https://github.com/sumdog/BigSense/blob/master/src/io/bigsense/spring/spring.xml>

Actions

```
...
<!-- Actions -->

<bean id="BaseAction" name="BaseAction" class="io.bigsense.action.ActionTrait" abstract="true">
  <property name="dbHandler" ref="databaseHandler" />
</bean>

<bean id="ActionSensor" name="ActionSensor" class="io.bigsense.action.SensorAction" parent="BaseAction">
  <property name="validator" ref="SensorActionValidator" />
</bean>
<bean id="ActionQuery" name="ActionQuery" class="io.bigsense.action.QueryAction" parent="BaseAction">
  <property name="validator" ref="QueryActionValidator" />
</bean>
...
```

Actions are called based on their *id*. The very first argument to the service is post-fixed with the word **Action**, so it is impossible for the user to inject anything and call a bean that isn't explicitly defined as an Action within Spring. Each Action must have a **ValidationTrait** implementation. This is not optional. If an action requires no validation, a blank validation trait must be made to pass all arguments unfiltered to the action.

```
...
<!-- Validators -->

<bean id="QueryActionValidator" name="QueryActionValidator" class="io.bigsense.validation.
QueryActionValidator" />
<bean id="AggregateActionValidator" name="AggregateActionValidator" class="io.bigsense.validation.
AggregateActionValidator" />
<bean id="SensorActionValidator" name="SensorActionValidator" class="io.bigsense.validation.
SensorActionValidator" />
<bean id="StatusActionValidator" name="StatusActionValidator" class="io.bigsense.validation.
StatusActionValidator" />
<bean id="ImageActionValidator" name="ImageActionValidator" class="io.bigsense.validation.
ImageActionValidator" />
...
```

Validators are fairly straight forward. They have no dependencies. The base **ValidatorTrait** contains several helper functions to assist in validating arguments.

TODO: Finish this section

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:util="http://www.springframework.org/schema/util"
  xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans
/spring-beans-3.0.xsd
  http://www.springframework.org/schema/util http://www.springframework.org/schema/util/spring-util-3.0.
```

```

xsd">

    <!-- Actions -->

    <bean id="BaseAction" name="BaseAction" class="io.bigsense.action.ActionTrait" abstract="true">
        <property name="dbHandler" ref="databaseHandler" />
    </bean>

    <bean id="ActionSensor" name="ActionSensor" class="io.bigsense.action.SensorAction" parent="BaseAction">
        <property name="validator" ref="SensorActionValidator" />
    </bean>
    <bean id="ActionQuery" name="ActionQuery" class="io.bigsense.action.QueryAction" parent="BaseAction">
        <property name="validator" ref="QueryActionValidator" />
    </bean>

    <bean id="ActionStatus" name="ActionStatus" class="io.bigsense.action.StatusAction" parent="BaseAction">
        <property name="validator" ref="StatusActionValidator" />
    </bean>

    <bean id="ActionAggregate" name="ActionAggregate" class="io.bigsense.action.AggregateAction" parent="
BaseAction">
        <property name="validator" ref="AggregateActionValidator" />
    </bean>

    <bean id="ActionImage" name="ActionImage" class="io.bigsense.action.ImageAction" parent="BaseAction">
        <property name="validator" ref="ImageActionValidator" />
    </bean>

    <bean id="ActionReport" name="ActionReport" class="io.bigsense.action.ReportAction" parent="BaseAction">
        <property name="validator" ref="StatusActionValidator" />
        <property name="converters" ref="convertersAsScalaMap" />
    </bean>

    <!-- Security -->
    <bean id="SecurityManager" name="SecurityManager" class="io.bigsense.security.SignatureSecurityManager" />

    <!-- Processors -->
    <bean id="BaseProcessor" name="BaseProcessor" class="io.bigsense.processor.ProcessorTrait" abstract="true">
        <property name="dbHandler" ref="databaseHandler" />
    </bean>

    <bean id="CounterProcessor" name="ProcessorCounter" class="io.bigsense.processor.CounterProcessor" parent="
BaseProcessor" />
    <bean id="ImageProcessor" name="ProcessorImage" class="io.bigsense.processor.ImageProcessor" parent="
BaseProcessor" />

    <!-- Smart Vision Analyzers -->
    <bean id="AnalyzerWaterLevel" name="AnalyzerWaterLevel" class="io.bigsense.smartvision.WaterLevelAnalyzer"
/>

    <!-- Validators -->

    <bean id="QueryActionValidator" name="QueryActionValidator" class="io.bigsense.validation.
QueryActionValidator" />
    <bean id="AggregateActionValidator" name="AggregateActionValidator" class="io.bigsense.validation.
AggregateActionValidator" />
    <bean id="SensorActionValidator" name="SensorActionValidator" class="io.bigsense.validation.
SensorActionValidator" />
    <bean id="StatusActionValidator" name="StatusActionValidator" class="io.bigsense.validation.
StatusActionValidator"/>
    <bean id="ImageActionValidator" name="ImageActionValidator" class="io.bigsense.validation.
ImageActionValidator"/>

    <!-- Convertors -->
    <bean id="convertersAsScalaMap" class="scala.collection.JavaConversions" factory-method="mapAsScalaMap">
        <constructor-arg ref="converters"/>
    </bean>

    <util:map id="converters">
        <entry key="Units">

```

```

        <bean class="io.bigsense.conversion.UnitsConverter" />
    </entry>
    <entry key="Timezone">
        <bean class="io.bigsense.conversion.TimezoneConverter" />
    </entry>
</util:map>

<!-- Formats -->

<bean id="FormatAGRA.XML" class="io.bigsense.format.SenseDataXMLFormat" />
<bean id="FormatTXT" class="io.bigsense.format.TabDelimitedFormat" />
<bean id="FormatCSV" class="io.bigsense.format.CSVFormat" />
<bean id="FormatFLAT.XML" class="io.bigsense.format.FlatXMLFormat" />
<bean id="FormatTABLE.HTML" class="io.bigsense.format.TableHTMLFormat" />

<!-- logging -->
<bean name="logger" class="io.bigsense.spring.LoggingInterceptor" />
<bean id="loggingAdvisor" class="org.springframework.aop.support.RegexpMethodPointcutAdvisor">
    <property name="advice" ref="logger" />
    <property name="pattern" value="io.bigsense.*" />
</bean>
<bean class="org.springframework.aop.framework.autoproxy.DefaultAdvisorAutoProxyCreator" />

<!-- Database Handler -->
<bean id="databaseHandler" class="io.bigsense.db.DatabaseHandler">
    <property name="ds" ref="mainDataSource" />
    <property name="converters" ref="convertersAsScalaMap" />
    <property name="sqlCommands" ref="sqlMicrosoftCommands" />
</bean>

<bean id="sqlMicrosoftCommands" class="io.bigsense.util.SQLProperties">
    <constructor-arg name="sqlCommandFile" value="/io/bigsense/db/mssql.commands"/>
</bean>

<!-- Database configuration -->
<bean id="mainDataSource" class="com.jolbox.bonecp.BoneCPDataSource" destroy-method="close" >
    <property name="driverClass" value="net.sourceforge.jtds.jdbc.Driver" />
    <property name="jdbcUrl" value="connectionString" />
    <property name="username" value="dbUser"/>
    <property name="password" value="dbPass"/>
    <property name="idleConnectionTestPeriod" value="60"/>
    <property name="idleMaxAge" value="240"/>
    <property name="maxConnectionsPerPartition" value="30"/>
    <property name="minConnectionsPerPartition" value="10"/>
    <property name="partitionCount" value="3"/>
    <property name="acquireIncrement" value="5"/>
    <property name="statementsCacheSize" value="100"/>
    <property name="releaseHelperThreads" value="3"/>
</bean>
</beans>

```