

# DatasourceService

## Creating a [DataSource](#) Service without writing code

One thing that I did not see in HiveMind was any examples on how to easily implement a DataSource into my project using HiveMind. I then took it upon myself "again" to fix this so called deficiency. Looking thru the docs and getting real excited about contributing to a new and kewl framework, I tried to use every neat little keyword, factory and builder to create a totally reusable service that of course everyone should have. I go to the HiveMind JIRA and submit my request for an enhancement to HiveMind library, sitting back and patting myself on the back I realize that I really over thought the process, I retracted my request and went into hiding until now.

What I realized is that I was recoding all the things that HiveMind already does for me. I went back to the code and deleted it and went to my hivemodule.sdl now .xml and did this

```
<service-point id="DefaultDataSource" interface="javax.sql.DataSource">
  <invoke-factory service-id="hivemind.BuilderFactory">
    <construct class="org.apache.commons.dbcp.BasicDataSource">
      <set property="driverClassName" value="org.firebirdsql.jdbc.FBDriver"/>
      <set property="url" value="jdbc:firebirdsql:localhost/3050:D:/Firebird/data/etstore.gdb"/>
      <set property="username" value="SYSDBA"/>
      <set property="password" value="masterkey"/>
    </construct>
  </invoke-factory>
</service-point>
```

What I like about this is that the construct class can contain any class that implements javax.sql.DataSource and I can configure it thru its own setters or I can include this into a service that will do it for me or whatever.

Here is an example implementation:

```
Registry r = getRegistry();
DataSource datasource = (DataSource) r.getService( "datasource.DefaultDataSource", javax.sql.DataSource.
class );
Connection conn = datasource.getConnection();
```

In conclusion what I learned is there is a lot of power built into HiveMind and the other is not to junk up the core with what I thought was neat, instead write a Wiki and hope someone reads it.

---

Contributed by [KurtHoehn](#)