**StudentLife Documentation**

**Build And Deploy Application**

**Update: I am using Wildfly:**

**Please implement add this datasource to your standalone.xml:**

**add to standalone.xml in the datasources section:**

**<datasource jndi-name="java:/jdbc/StudentLife" pool-name="StudentLife" enabled="false" use-java-context="true">**

**<connection-url>jdbc:mysql://den1.mysql1.gear.host:3306/studentlifedb</connection-url>**

**<driver>mysql</driver>**

**<security>**

**<user-name>studentlifedb</user-name>**

**<password>Dd6Z5w\_0S4~s</password>**

**</security>**

**</datasource>**

I am using Tomcat Plume EE server: apache-tomee-plume-7.0.4 to test my app locally.

**1)** Under GlobalNamingResources within Tomcats server.xml, paste the following:

<Resource name="jdbc/StudentLife"

auth="Container"

type="javax.sql.DataSource"

username="studentlifedb"

password="Dd6Z5w\_0S4~s"

url="jdbc:mysql://den1.mysql1.gear.host:3306/studentlifedb"

driverClassName="com.mysql.cj.jdbc.Driver"

initialSize="20"

maxWaitMillis="15000"

maxTotal="75"

maxIdle="20"

maxAge="7200000"

testOnBorrow="true"

validationQuery="select 1"

/>

**2)** Within tomcat's context.xml, paste the following:

<ResourceLink name="jdbc/StudentLife"

global="jdbc/StudentLife"

type="javax.sql.DataSource"/>

**3)** Copy mysql-connector-java-6.0.5.jar

from \StudentLifeWebBuild\target\StudentLife\WEB-INF\lib

to :\Users\final\Documents\tomcatserverhome\lib

**4)** Use Maven goals “clean package” on the StudentLifeCommonBuild’s pom(this is the parent pom).

The end result should be a StudentLifauioe.war file. This file can be deployed to Tomcat.

**5)** Pull up the test page:

<http://localhost:8080/test>

http://localhost:8080/getStudents