
Article

[王喆](#) · Mar 12, 2023 5m read

Tutorial - Develop IRIS using SSH

SSH framework is a relatively representative framework of Java, which was popular many years ago. There are Struts+Spring+hibernate and Spring MVC+Spring+hibernate. Of course, I used the second one in college. If I can connect IRIS to Hibernate as a library, does it also mean that IRIS can be developed using SSH framework in theory?

Tools and environment

JDK 1.8

Maven

Hibernate 5.X.X

IRISHealth-2022.1.3

intellij idea

Windows 10 64

Create database

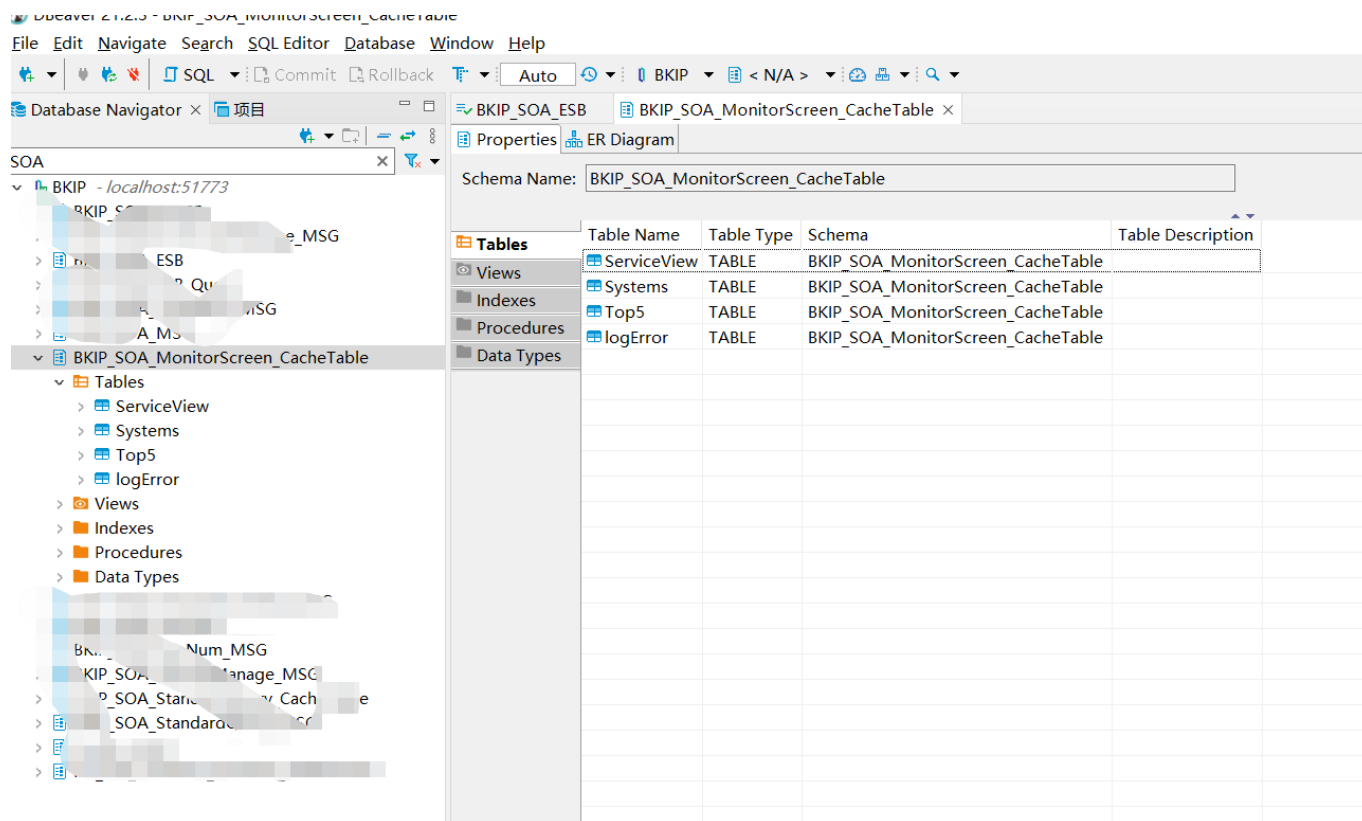
Create several tables in code mode

```
Class BKIP.SOA.MonitorScreen.CacheTable.logError Extends %Persistent
{
  /// Service Overview Cache Table
  Property SucNum As %String(MAXLEN = "");
  Property failNum As %String(MAXLEN = "");
  Property fdatetime As %String(MAXLEN = "");
}
```

As shown in the figure:

Tutorial - Develop IRIS using SSH

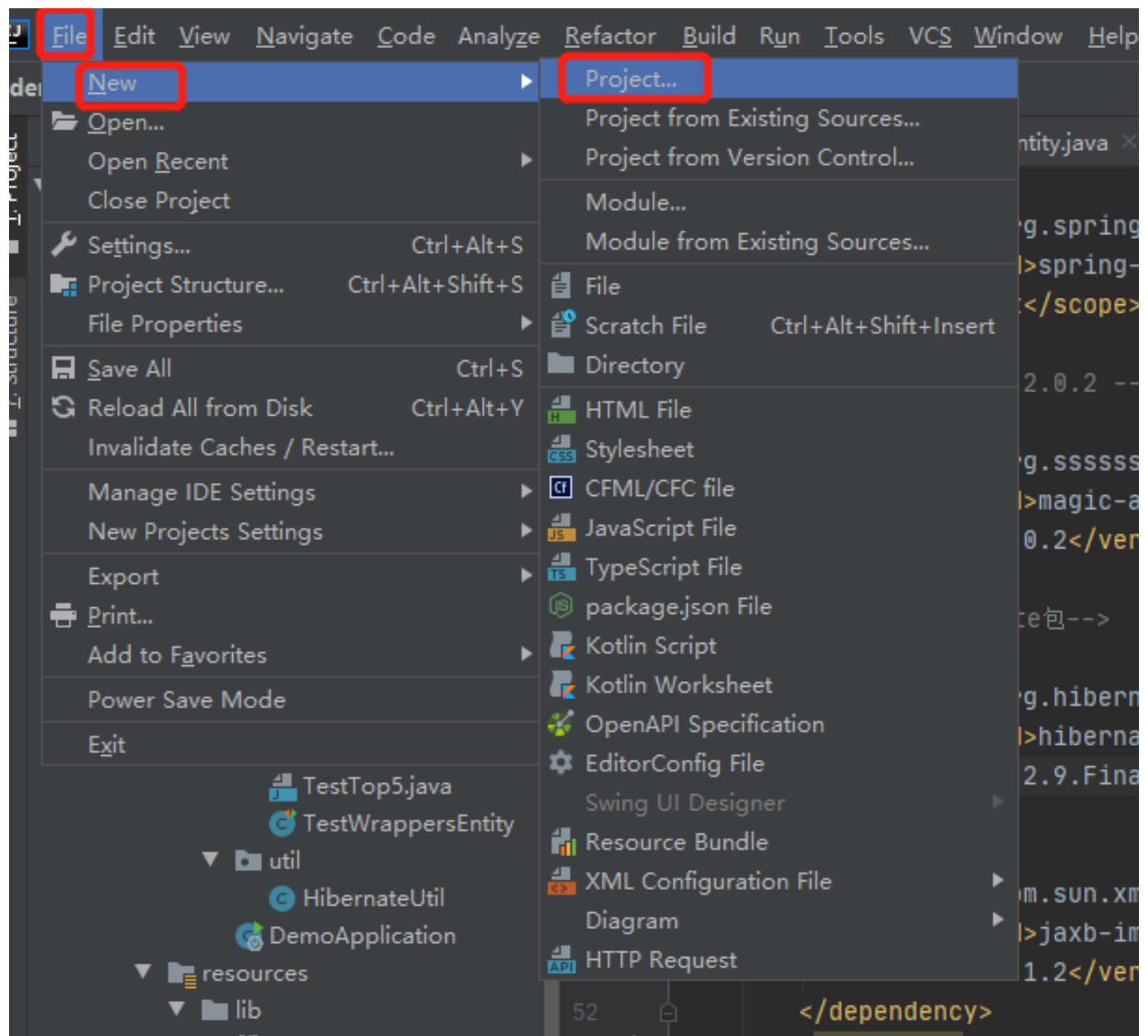
Published on InterSystems Developer Community (<https://community.intersystems.com>)



Create a Spring project

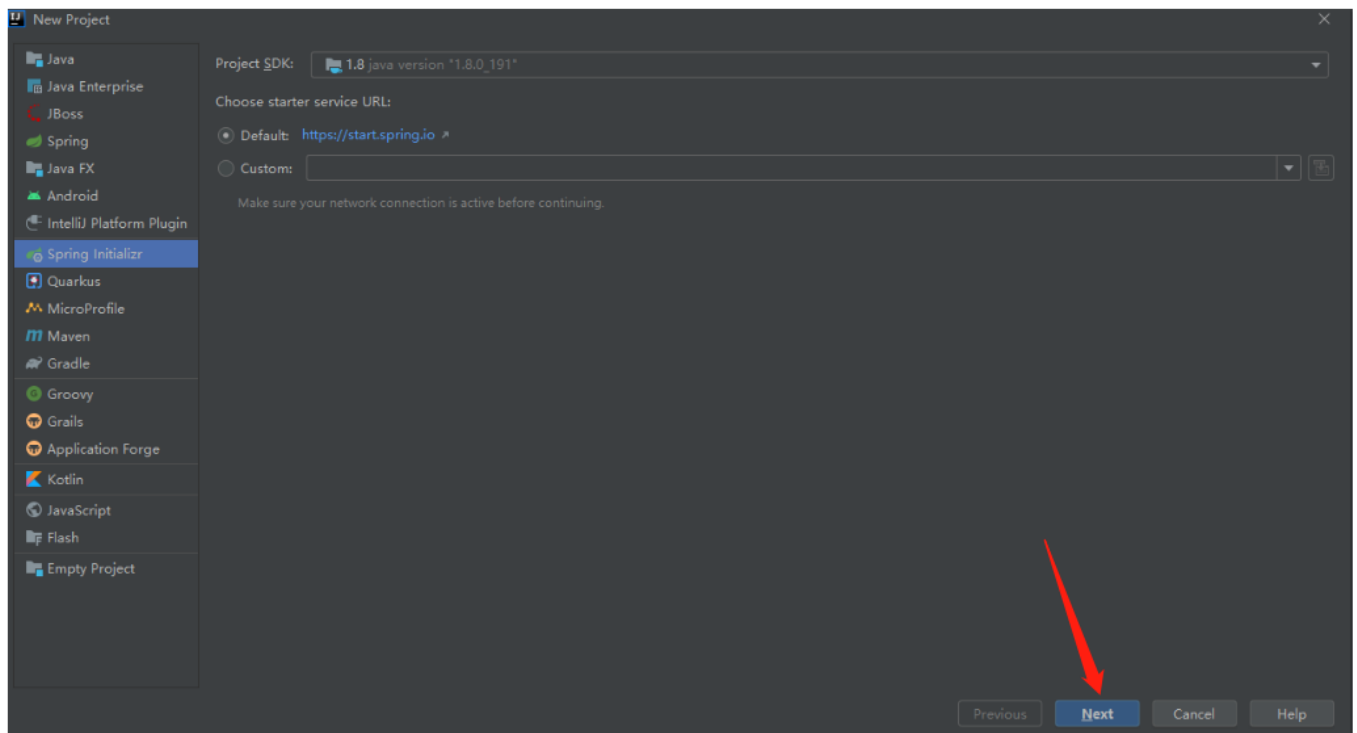
The next step is to use IDEA to create the mapping of the library and create the entity class

File—New—Project....

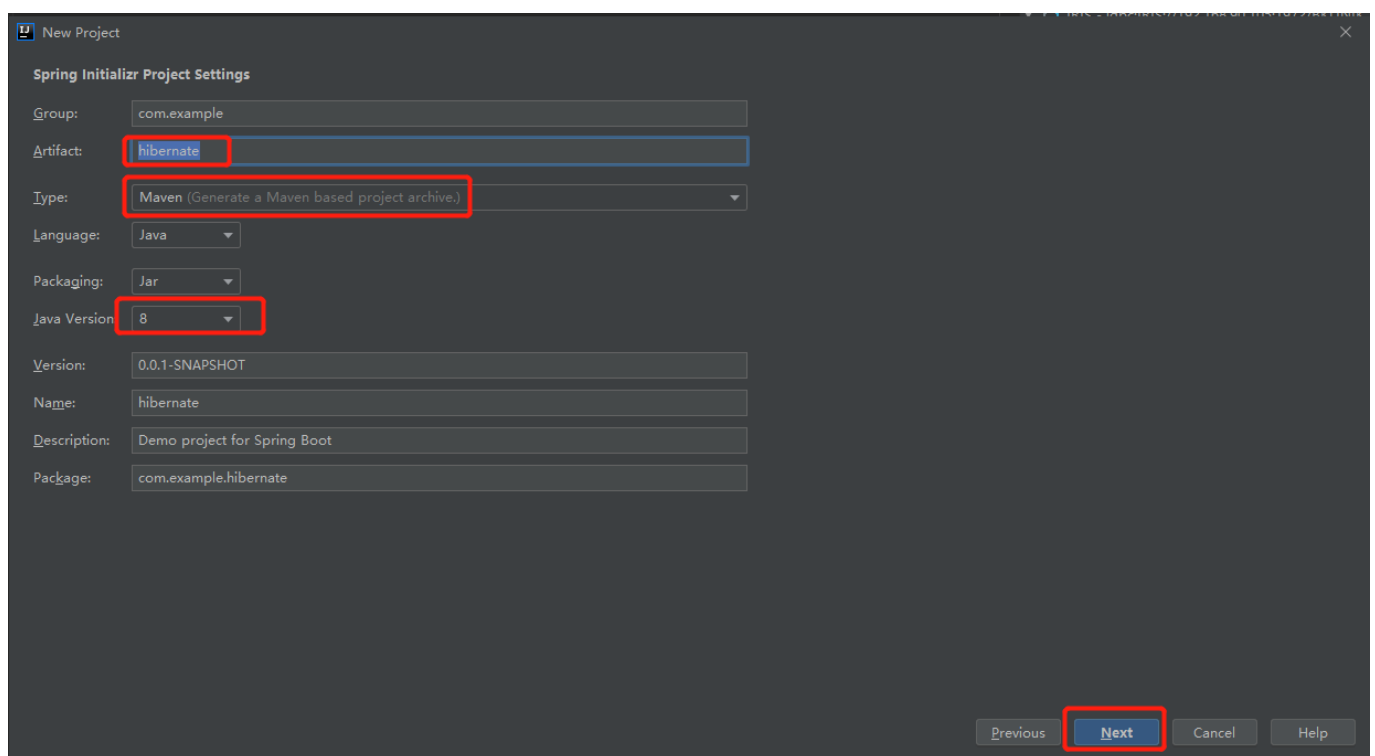


Tutorial - Develop IRIS using SSH

Published on InterSystems Developer Community (<https://community.intersystems.com>)

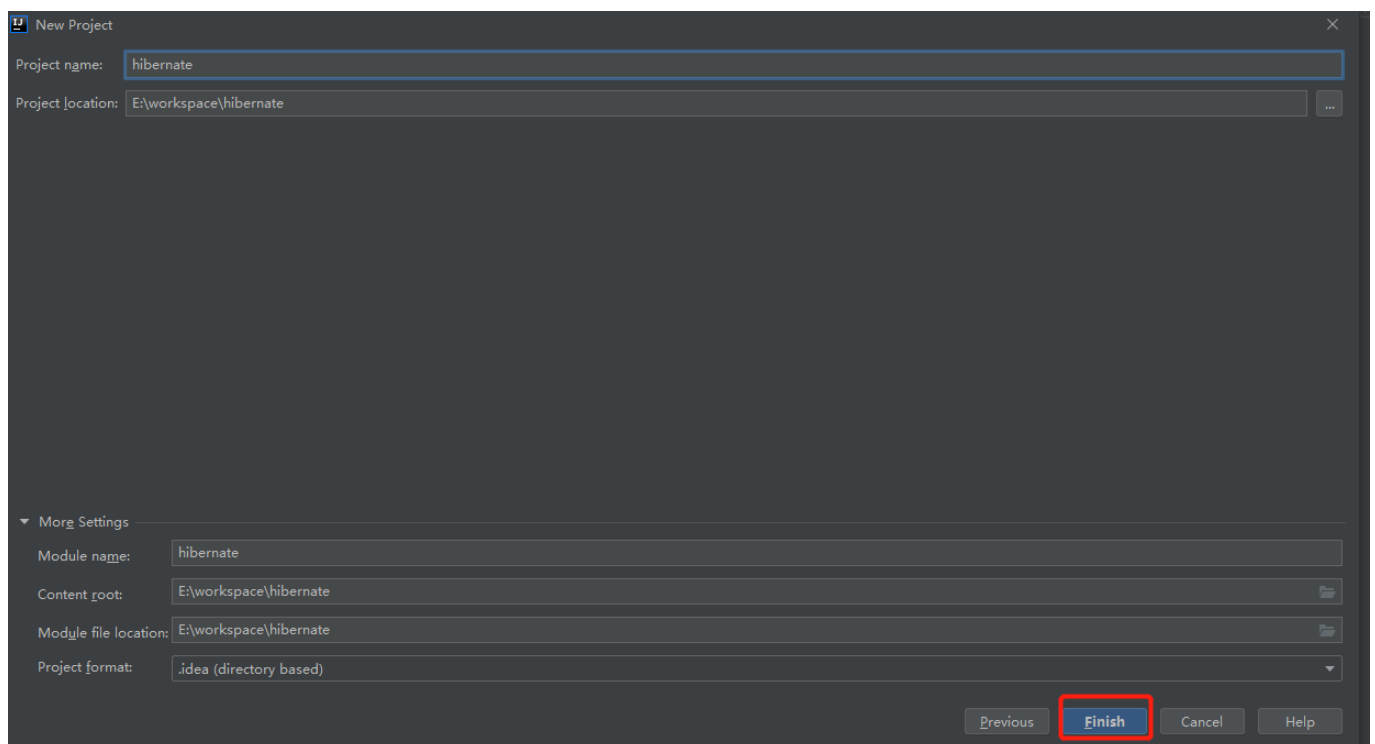
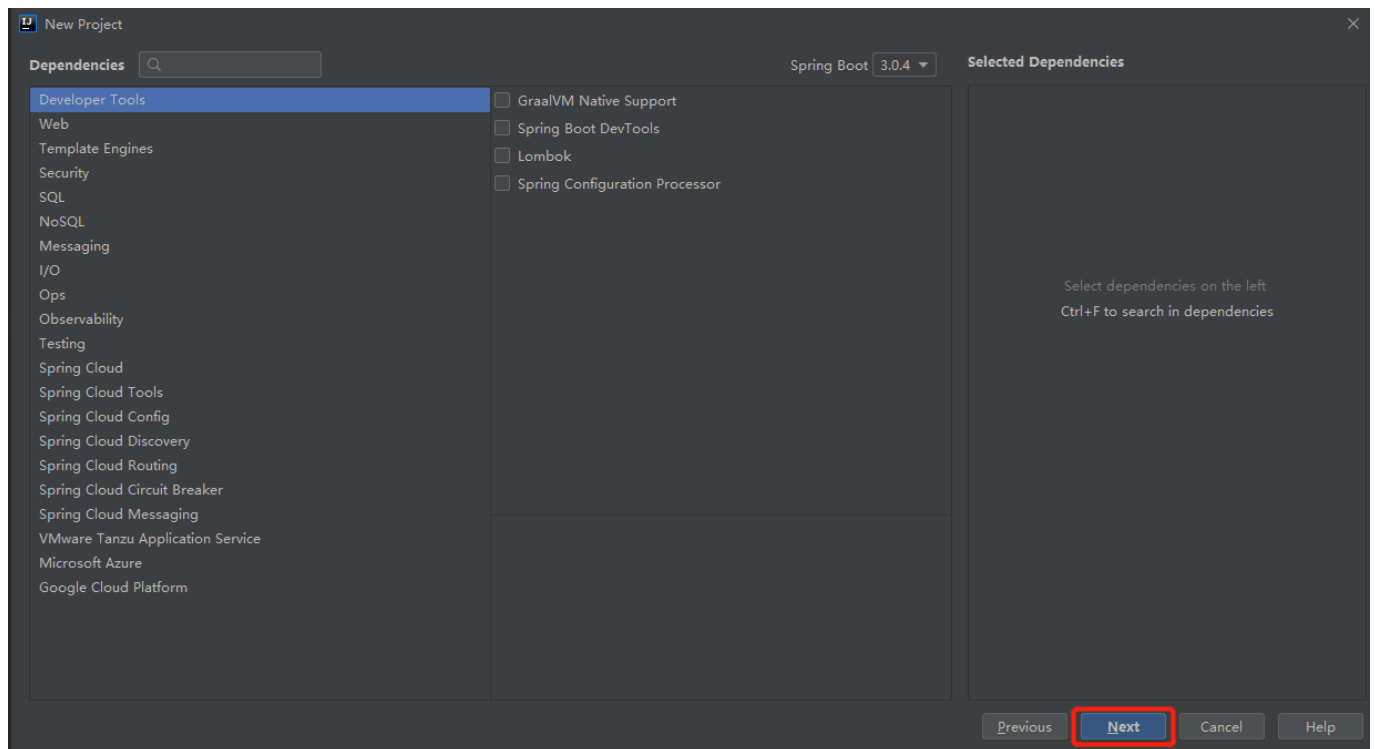


Name the project and select the jdk version, Click "Next"



Tutorial - Develop IRIS using SSH

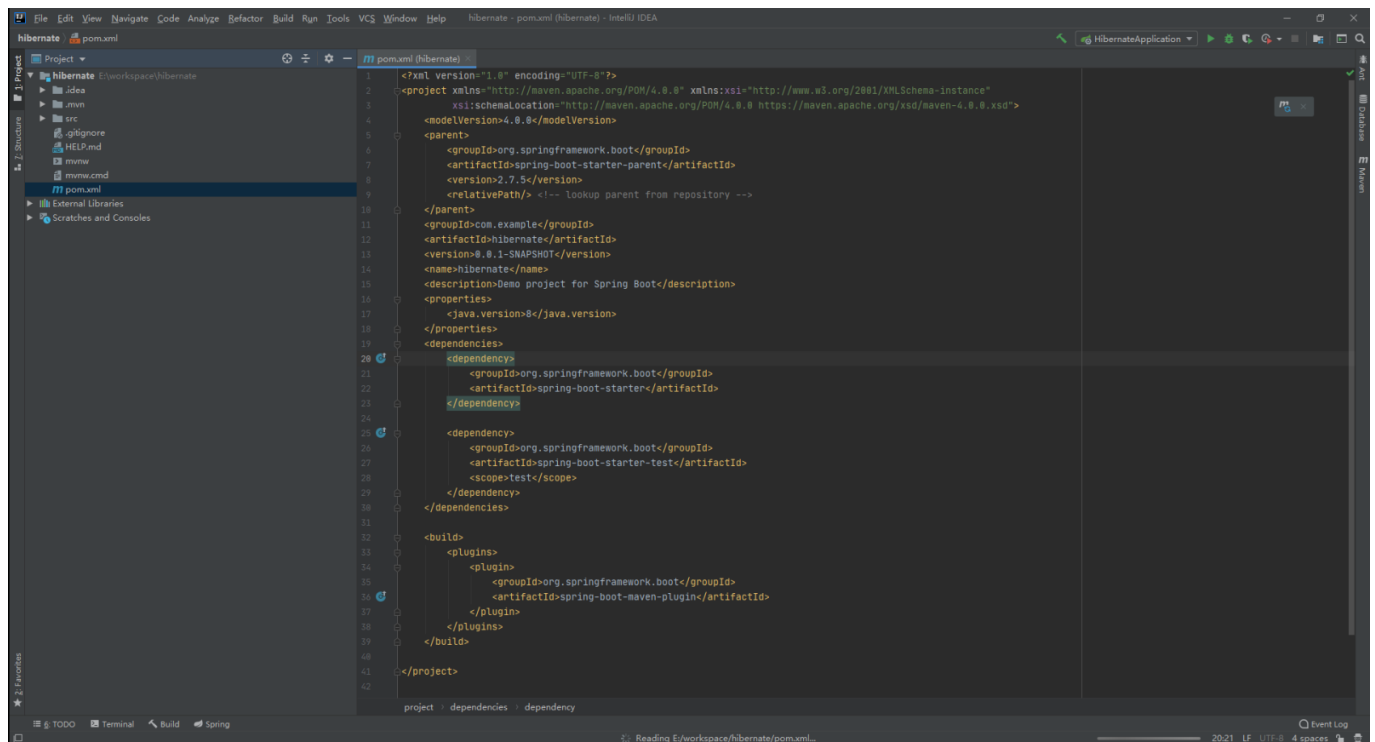
Published on InterSystems Developer Community (<https://community.intersystems.com>)



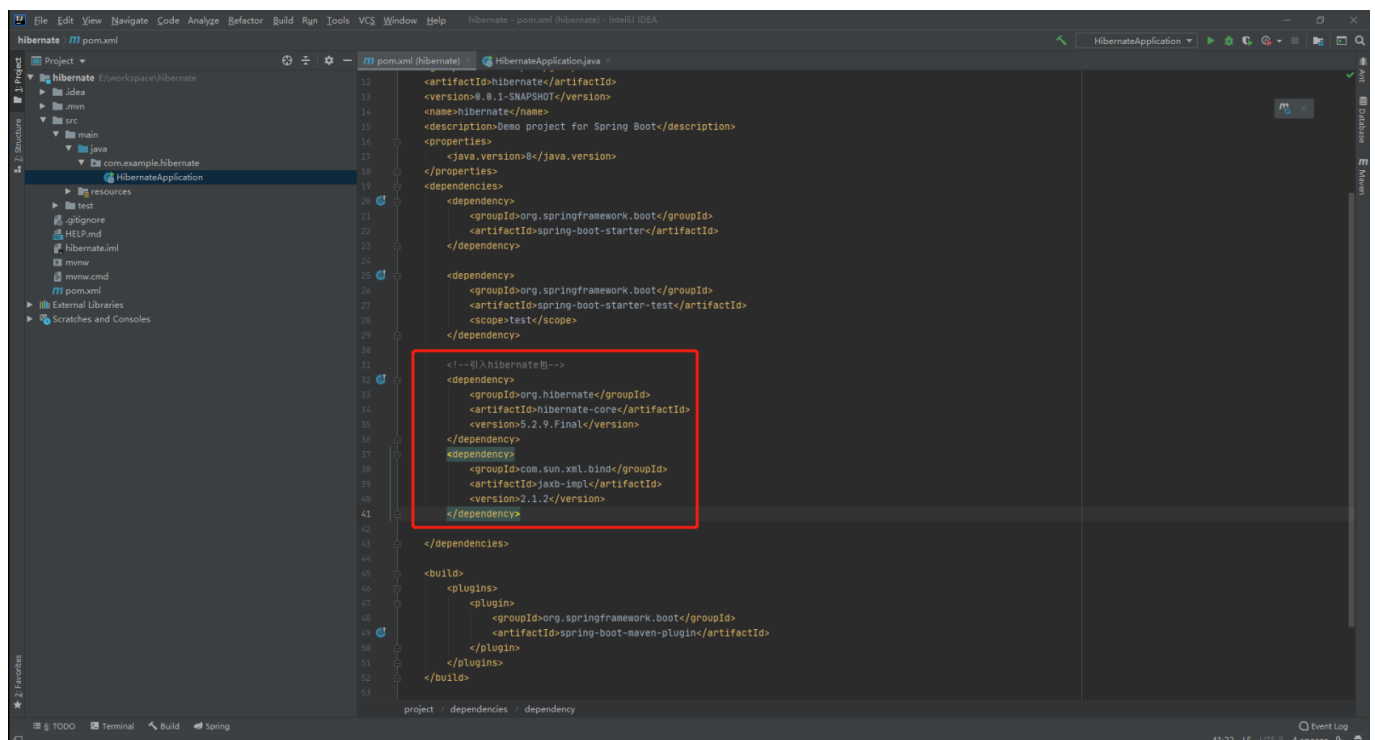
Wait for maven to finish creating the project, As shown in the figure

Tutorial - Develop IRIS using SSH

Published on InterSystems Developer Community (<https://community.intersystems.com>)



Add required packages

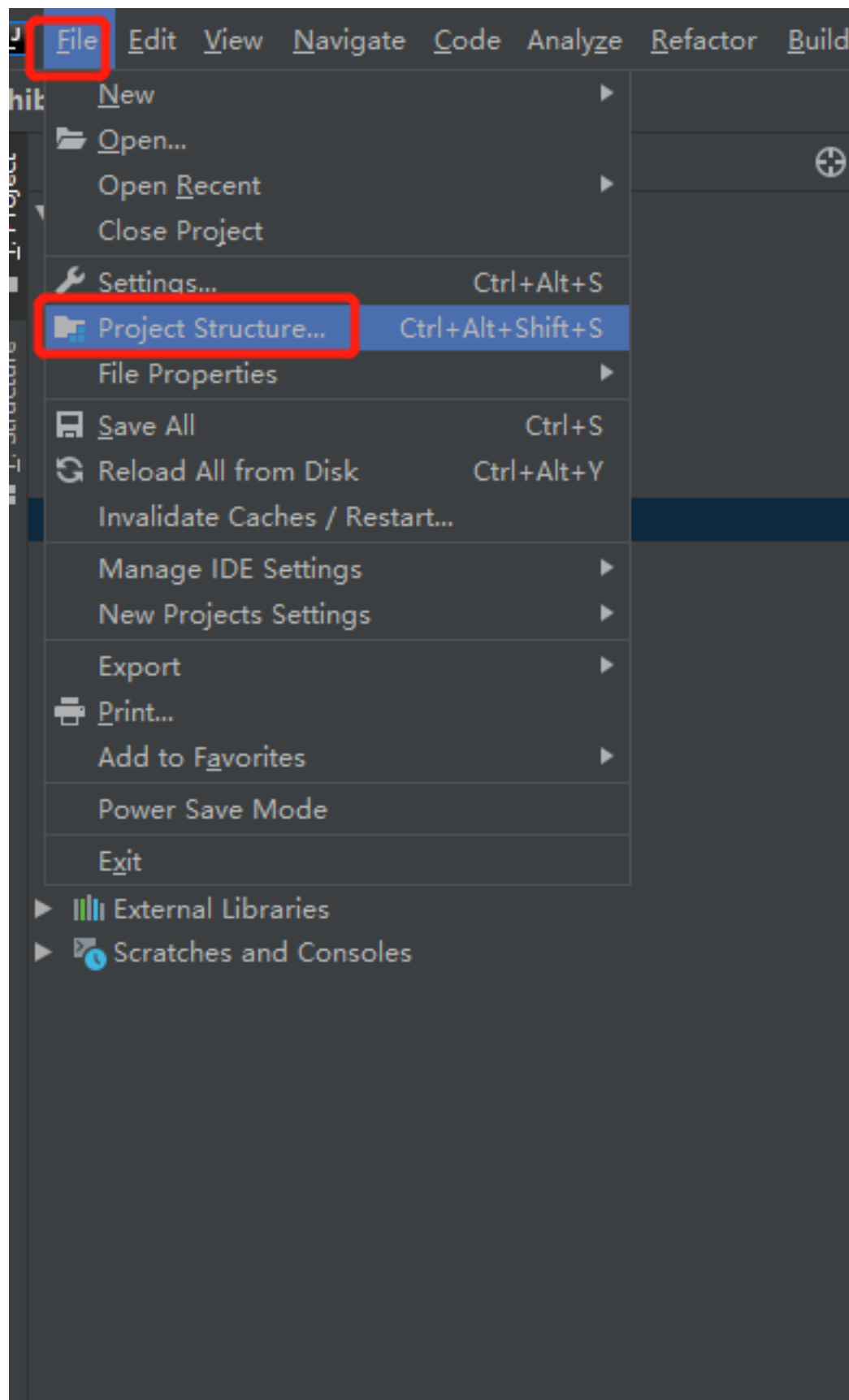


<!-- Introduce hibernate package -->

```
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-core</artifactId>
  <version>5.2.9.Final</version>
</dependency>
<dependency>
  <groupId>com.sun.xml.bind</groupId>
  <artifactId>jaxb-impl</artifactId>
```

```
<version>2.1.2</version>  
</dependency>
```

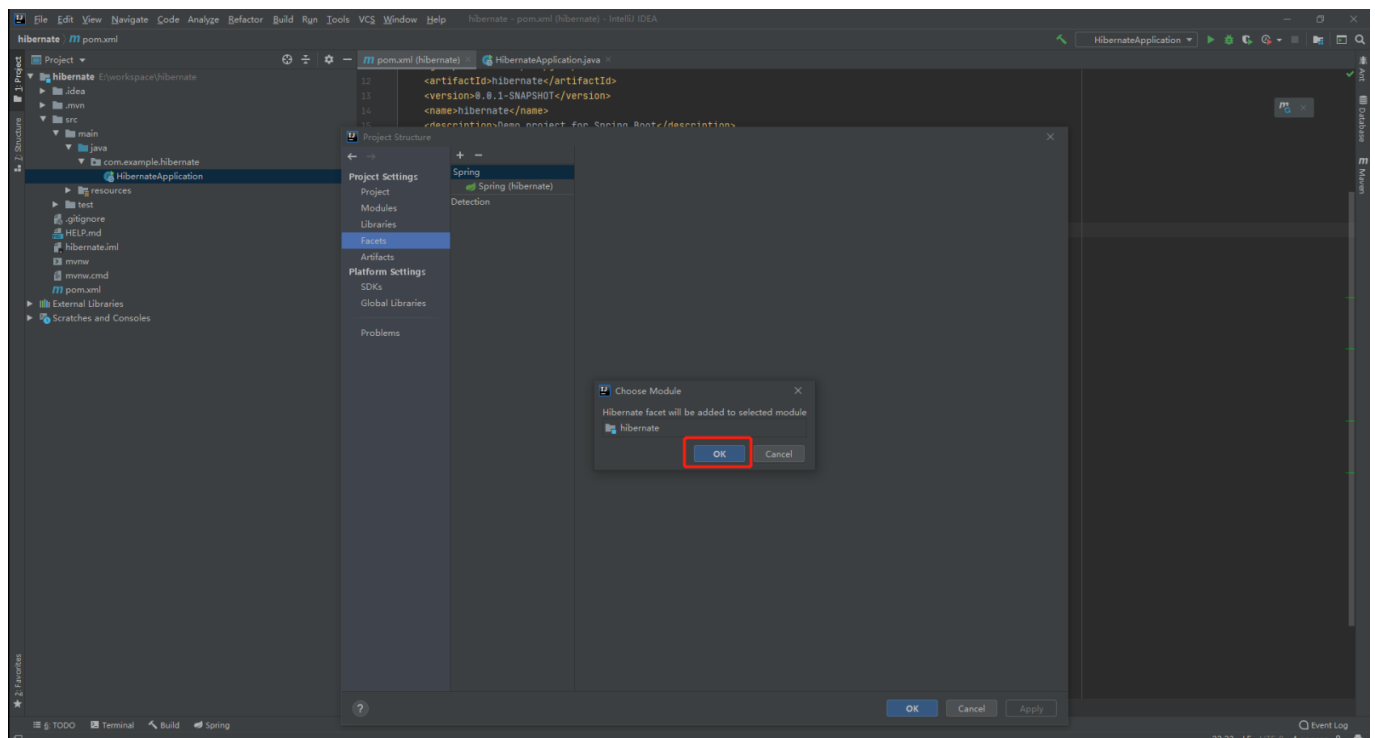
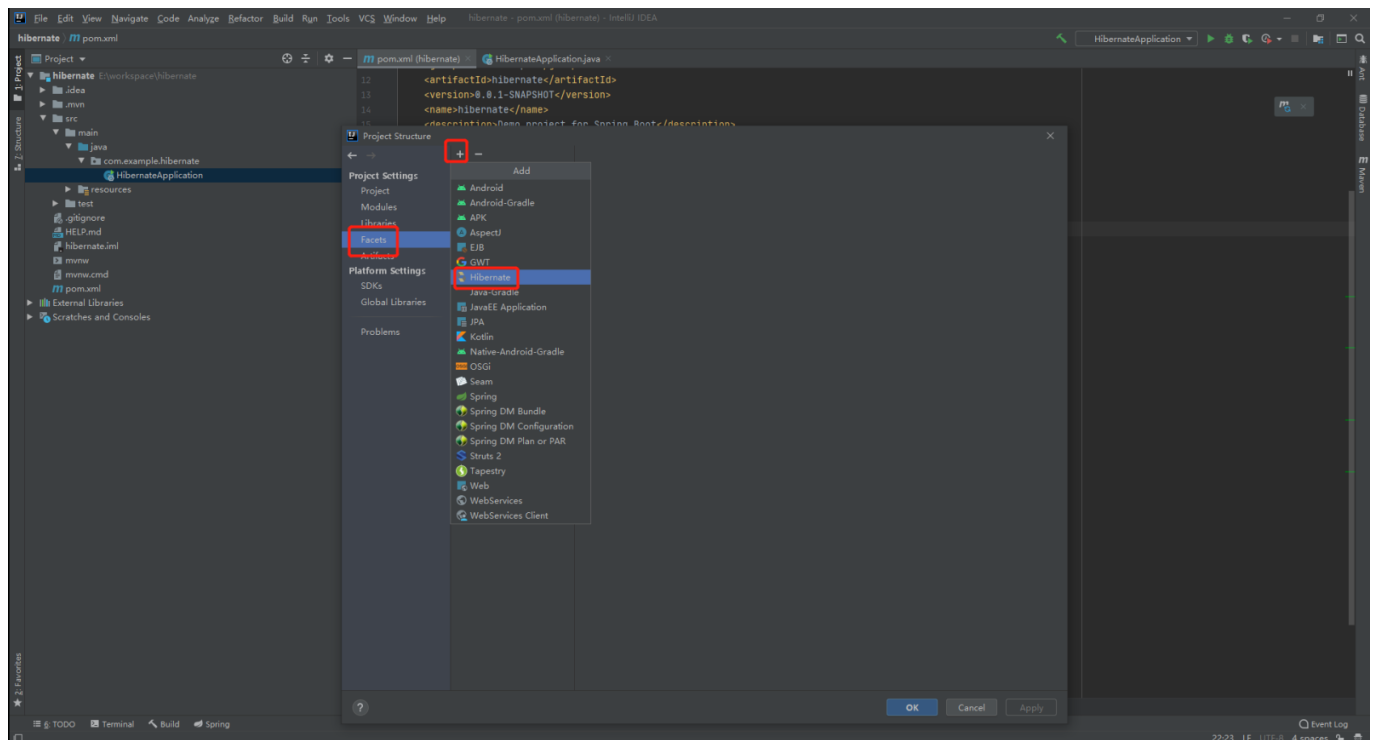
File—Project Structure...



Select Facets - add Hibernate

Tutorial - Develop IRIS using SSH

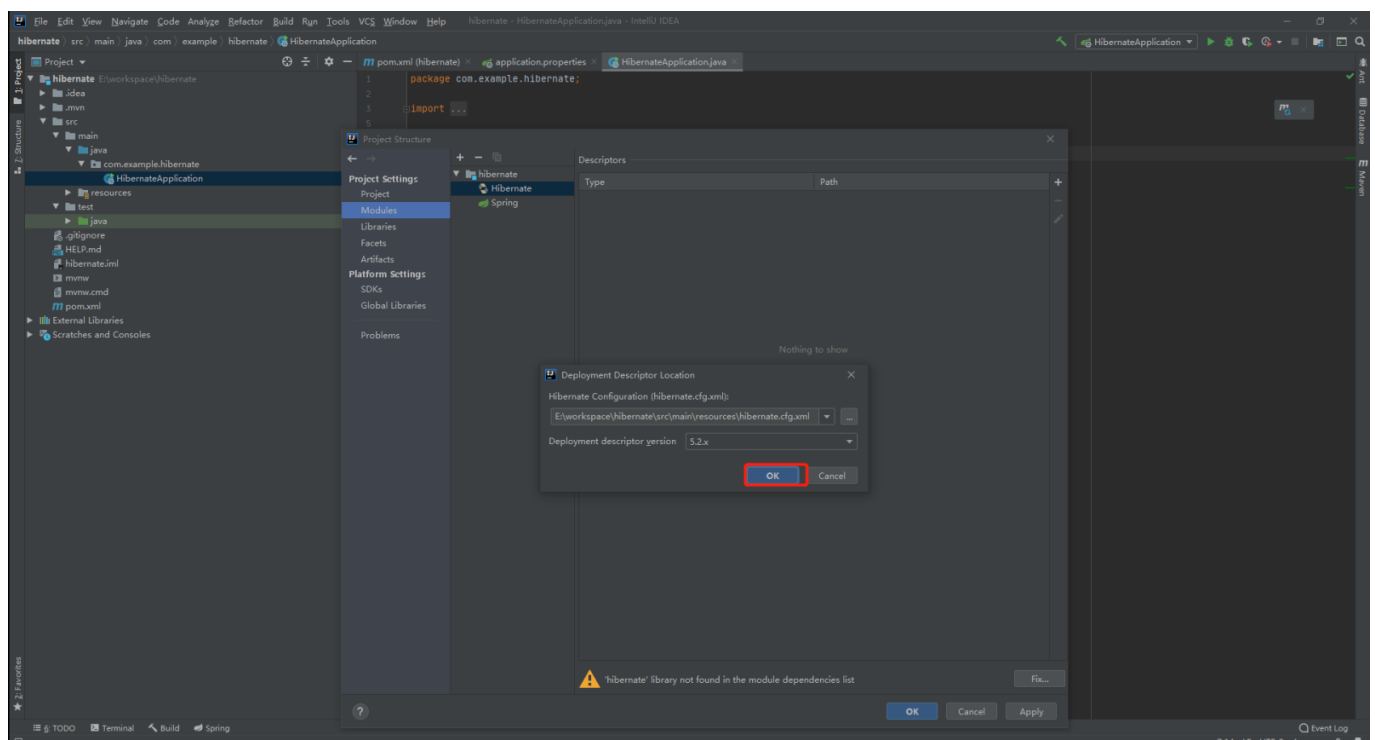
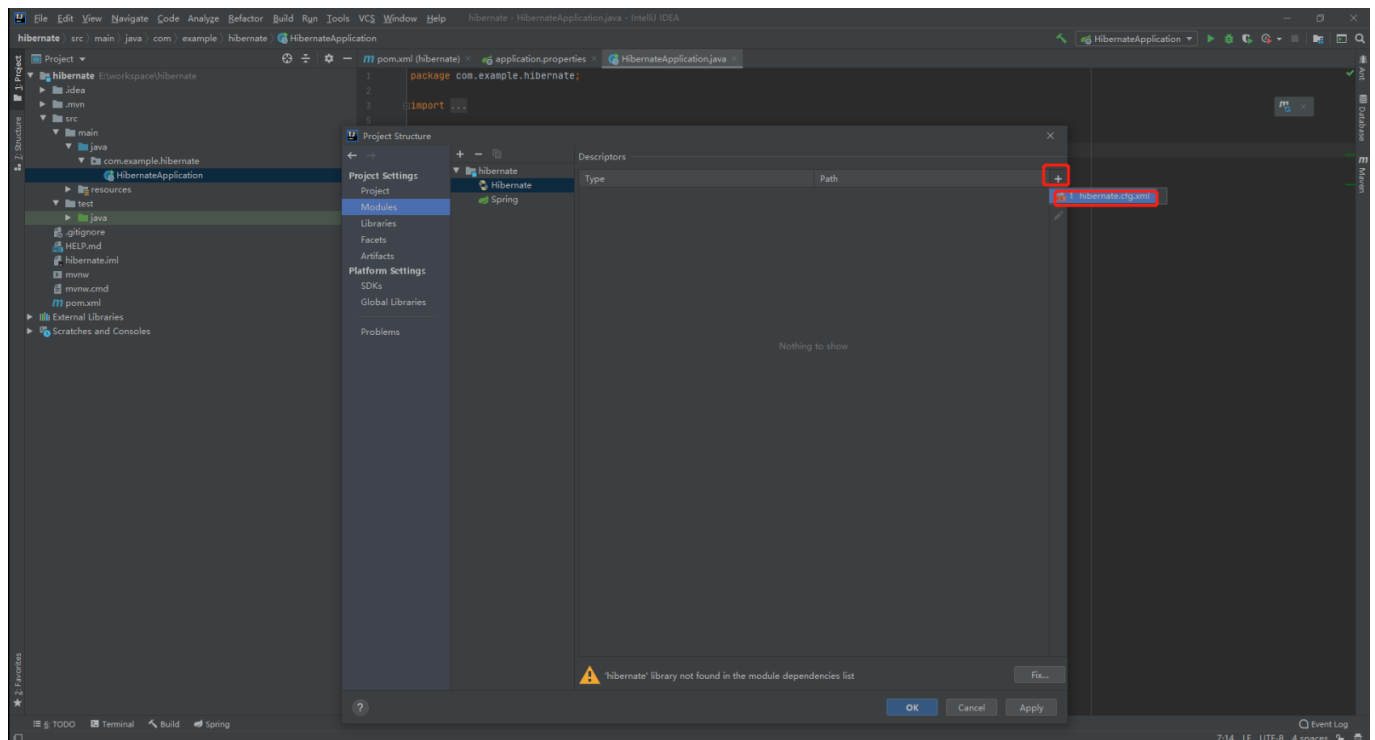
Published on InterSystems Developer Community (<https://community.intersystems.com>)



Add the configuration file as shown in the figure

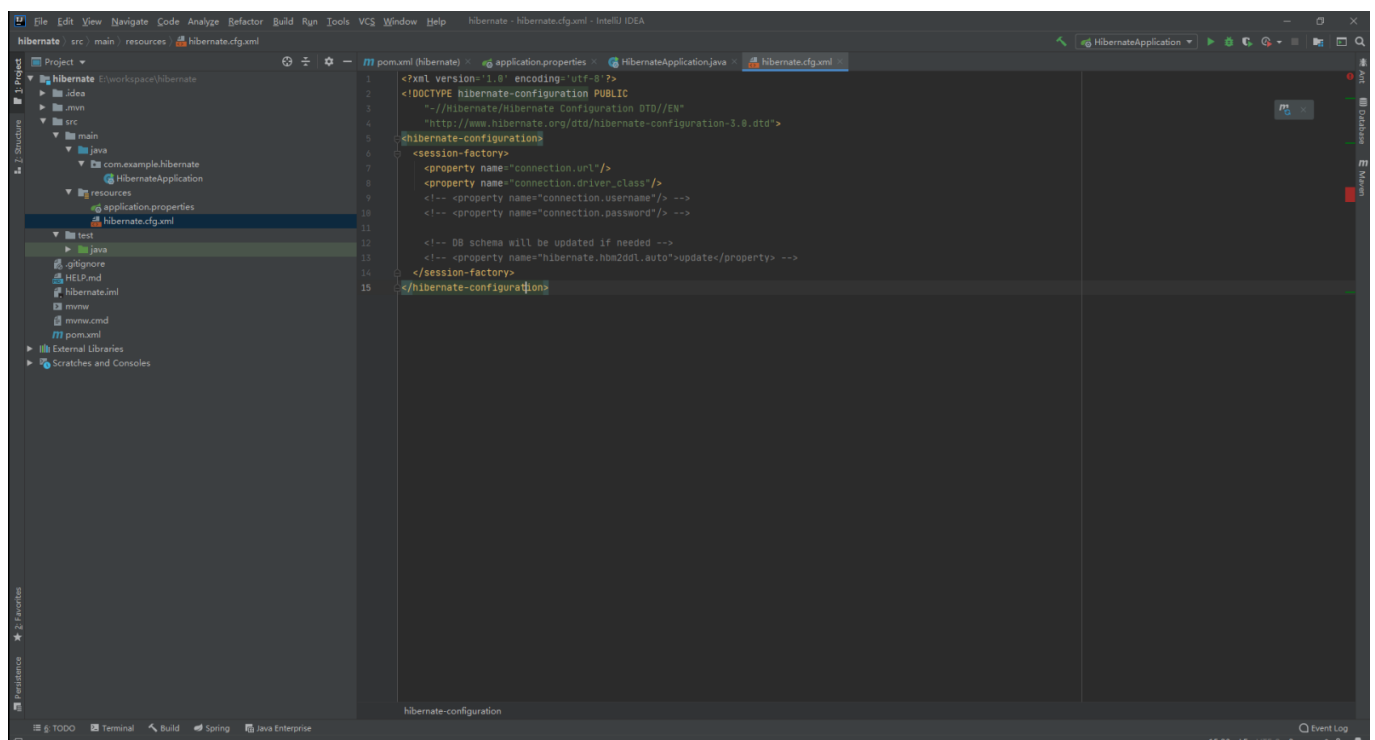
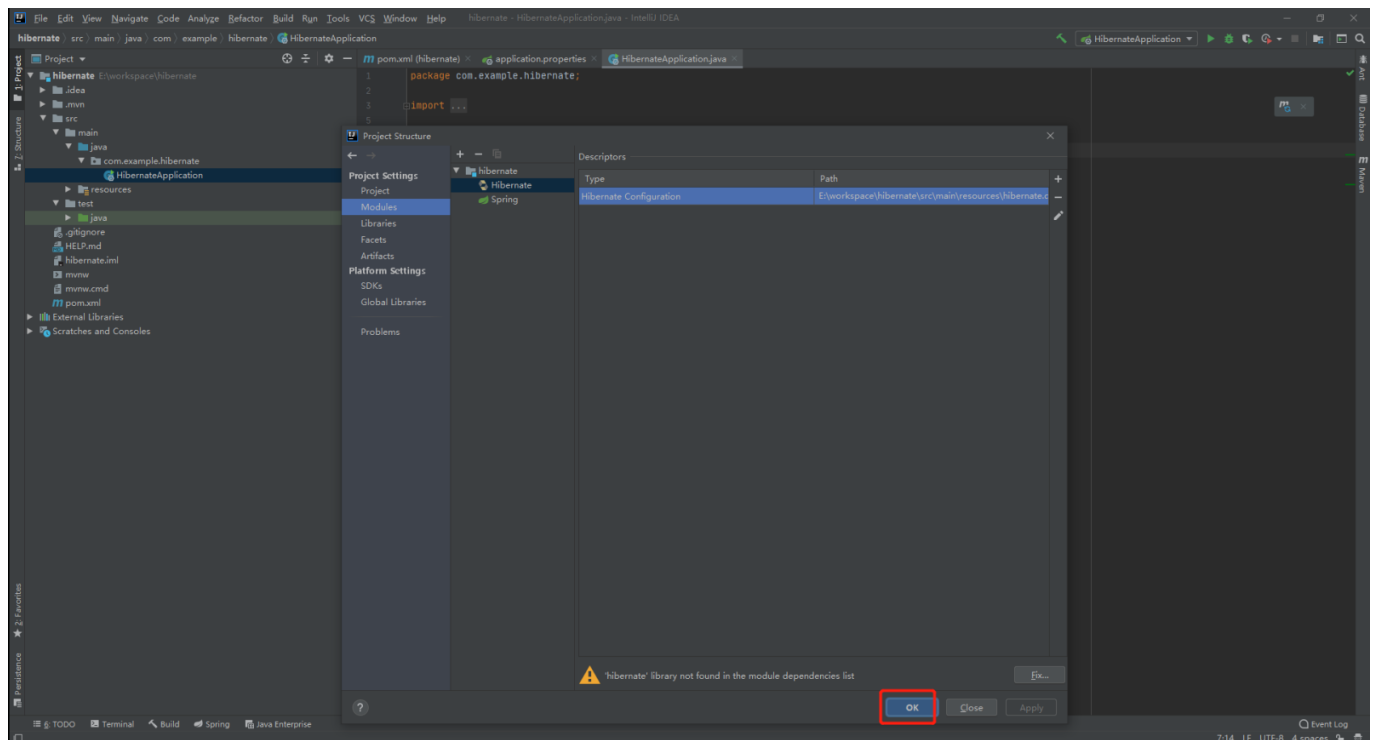
Tutorial - Develop IRIS using SSH

Published on InterSystems Developer Community (<https://community.intersystems.com>)

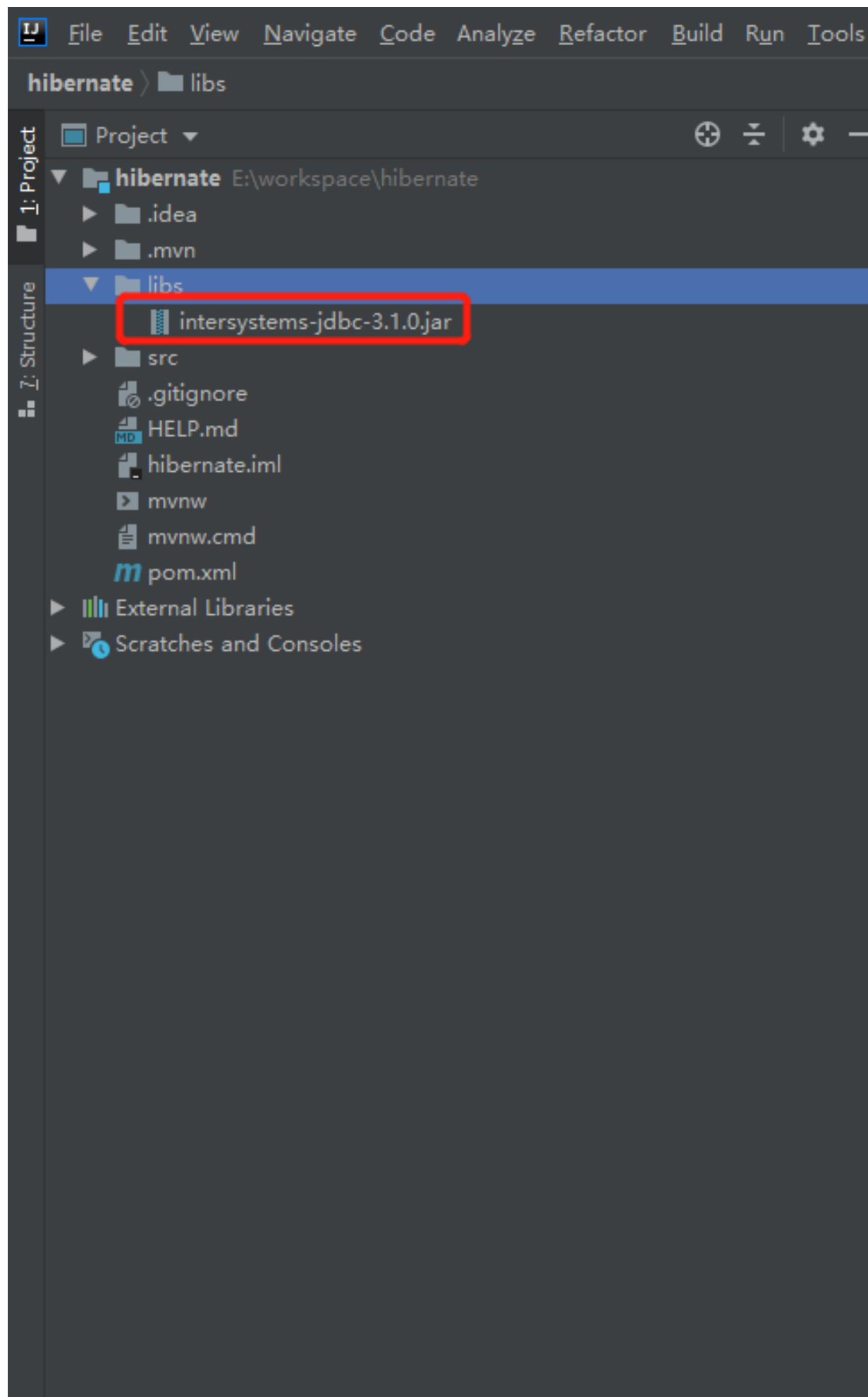


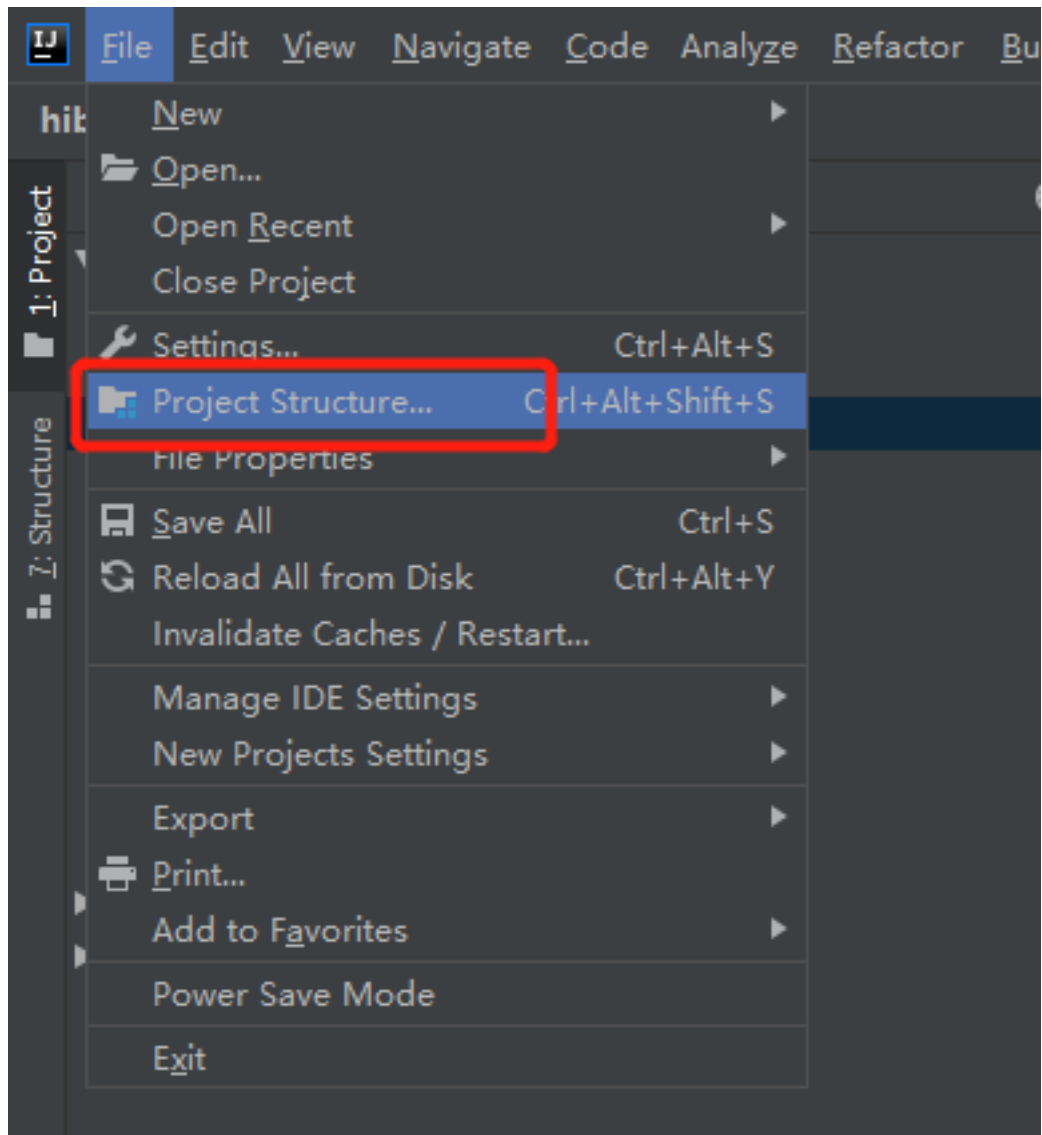
Tutorial - Develop IRIS using SSH

Published on InterSystems Developer Community (<https://community.intersystems.com>)



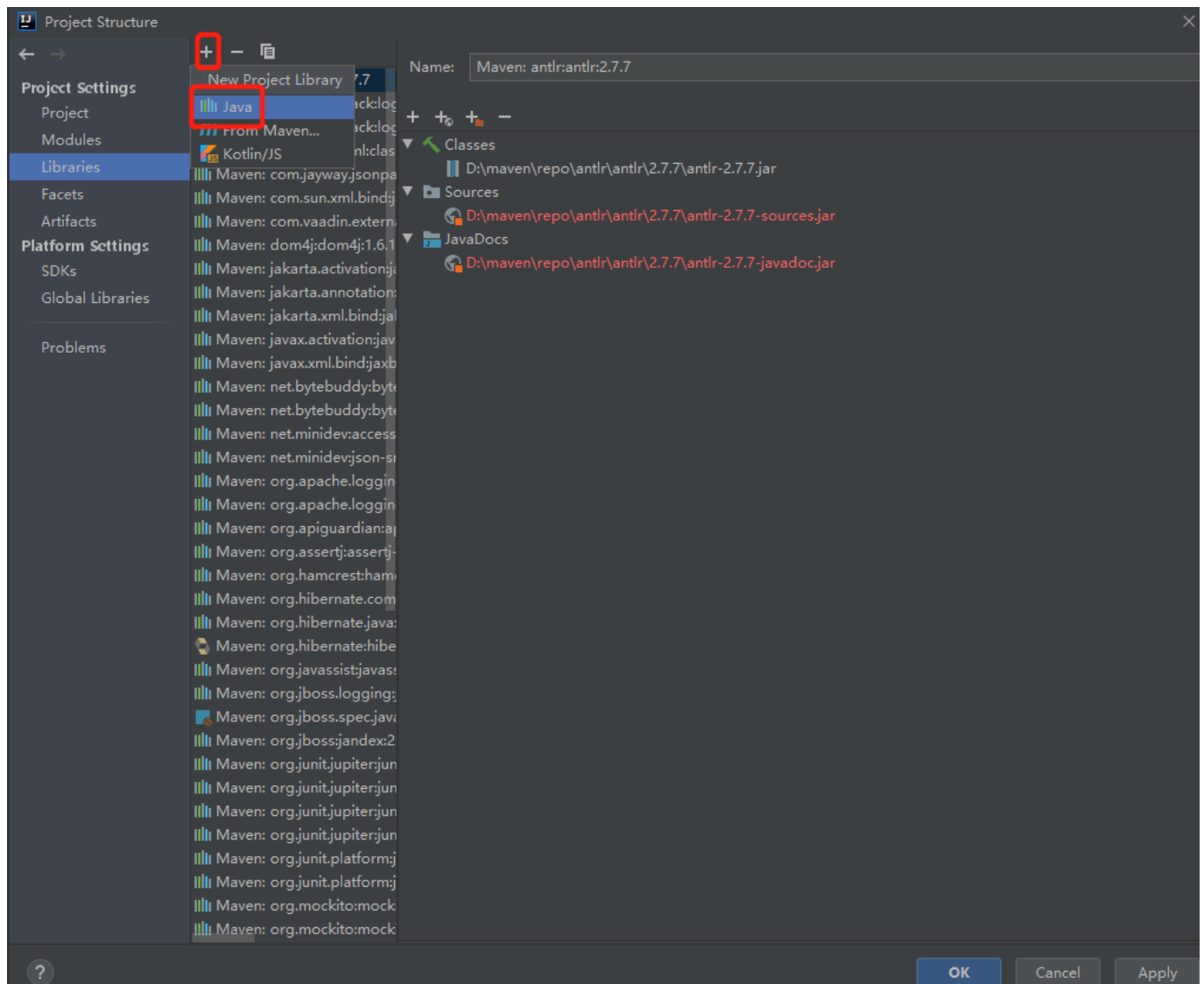
Jdbc package introducing iris





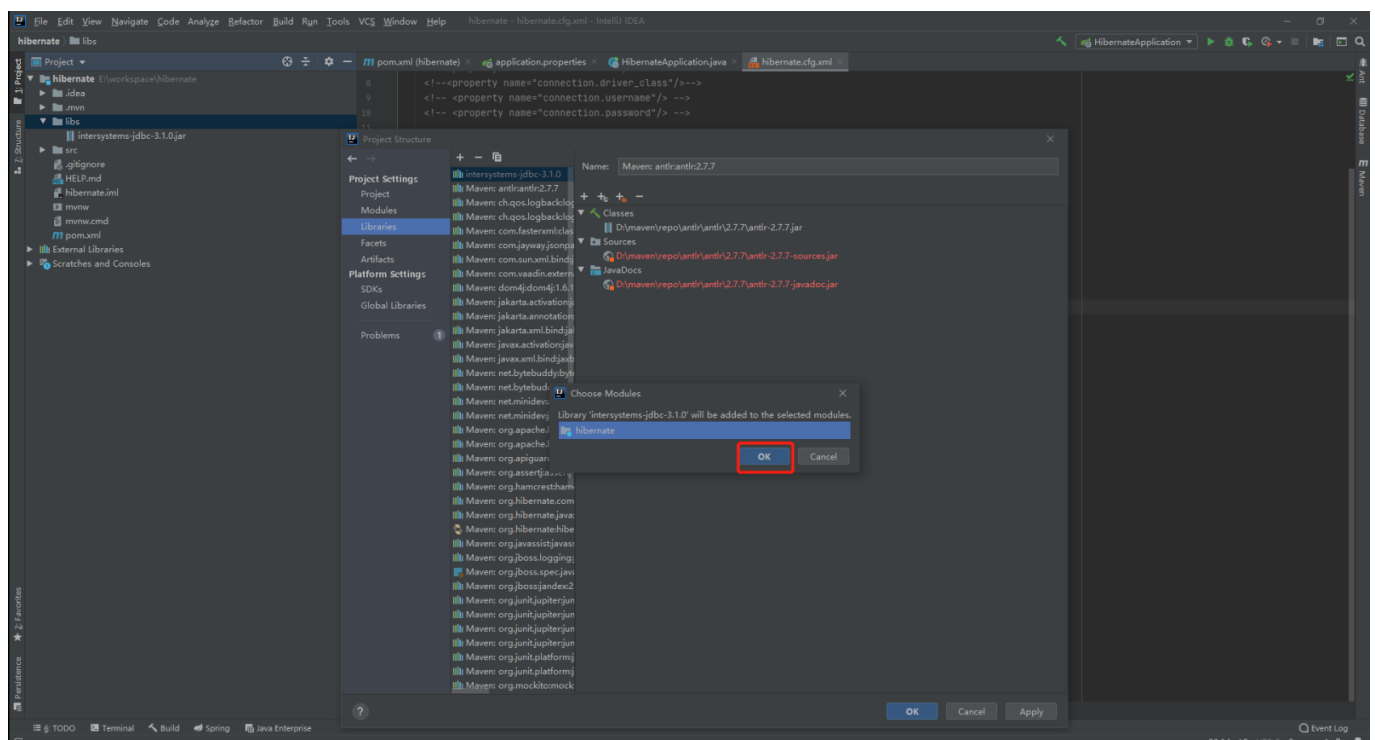
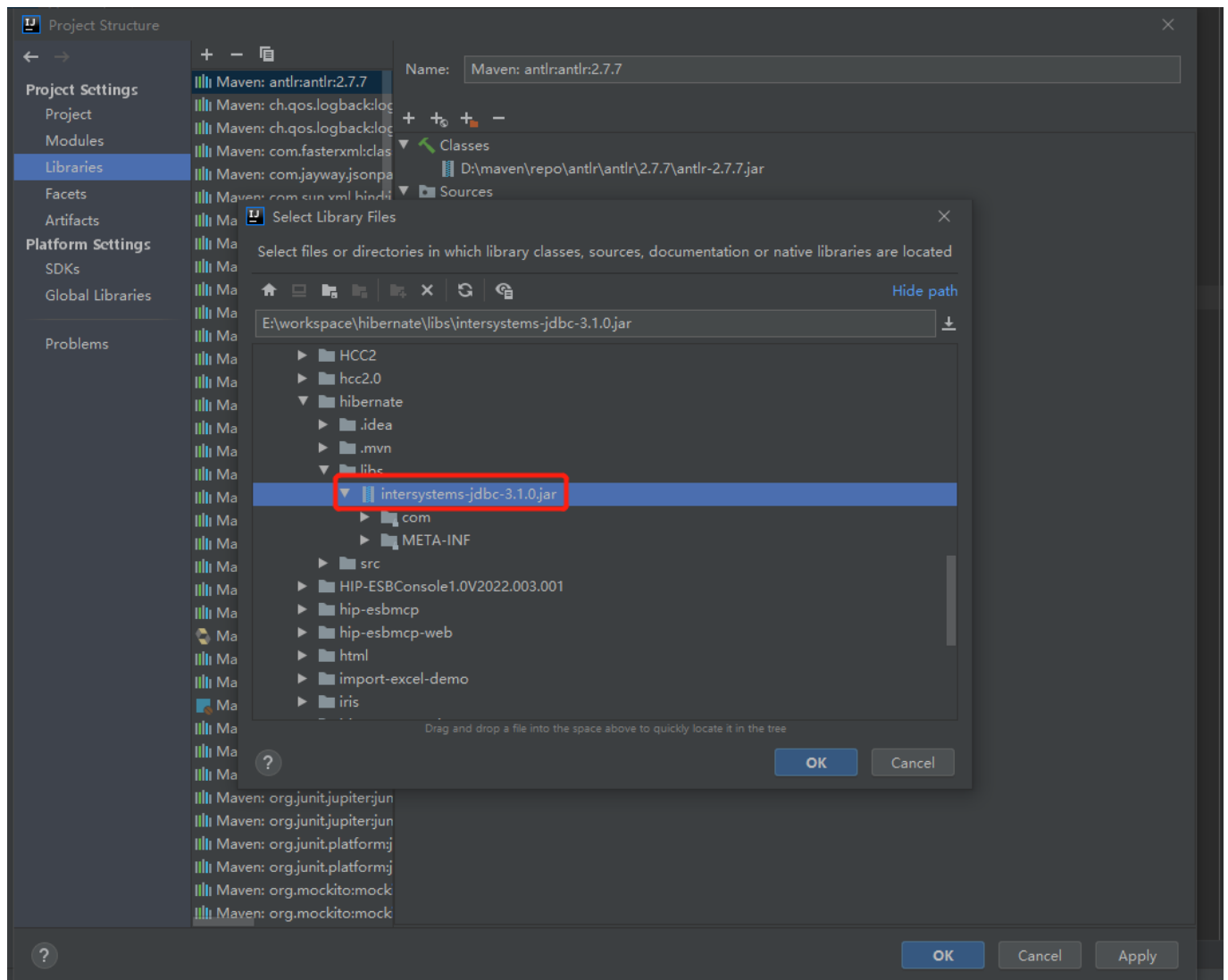
Tutorial - Develop IRIS using SSH

Published on InterSystems Developer Community (<https://community.intersystems.com>)



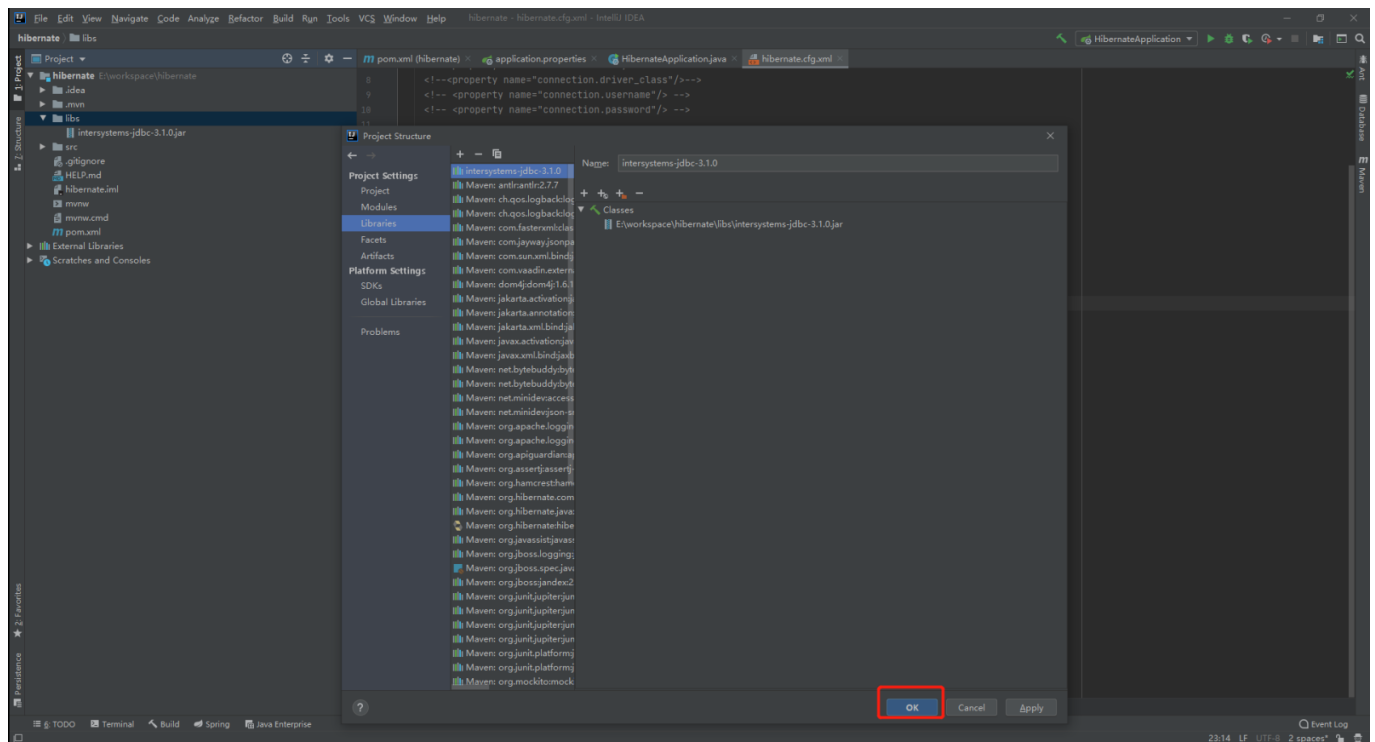
Tutorial - Develop IRIS using SSH

Published on InterSystems Developer Community (<https://community.intersystems.com>)

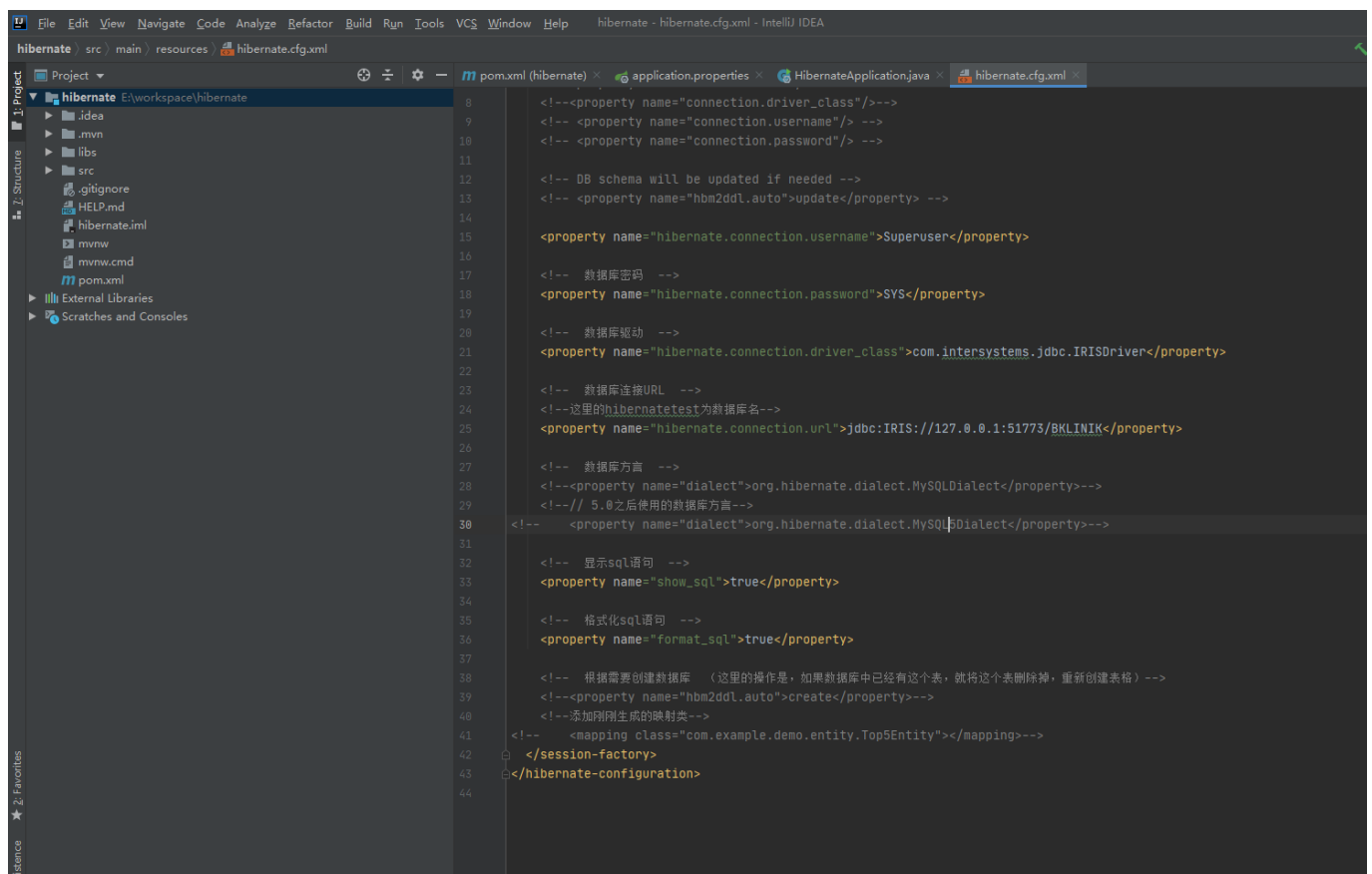


Tutorial - Develop IRIS using SSH

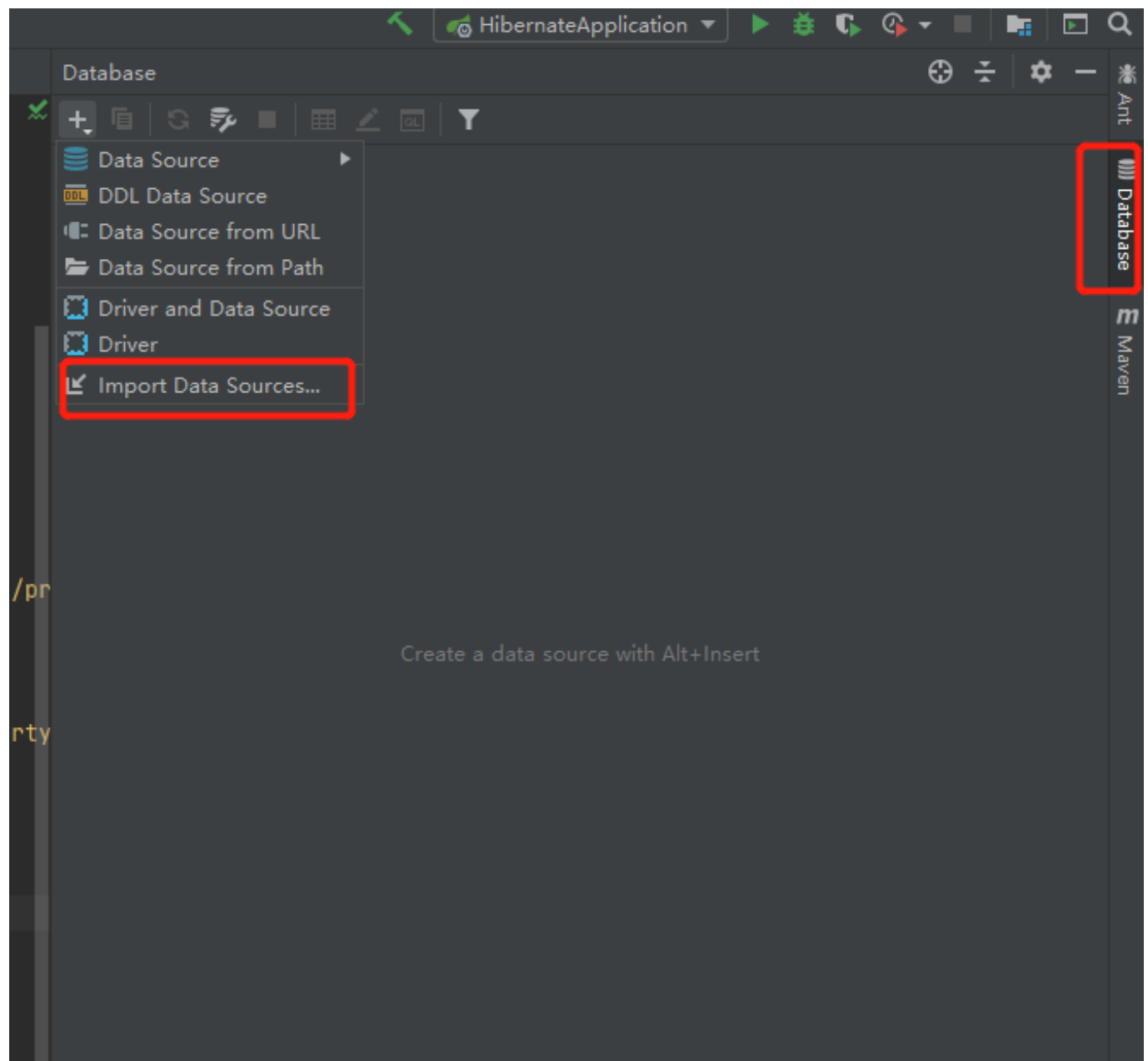
Published on InterSystems Developer Community (<https://community.intersystems.com>)



Enter connection information in Hibernate configuration



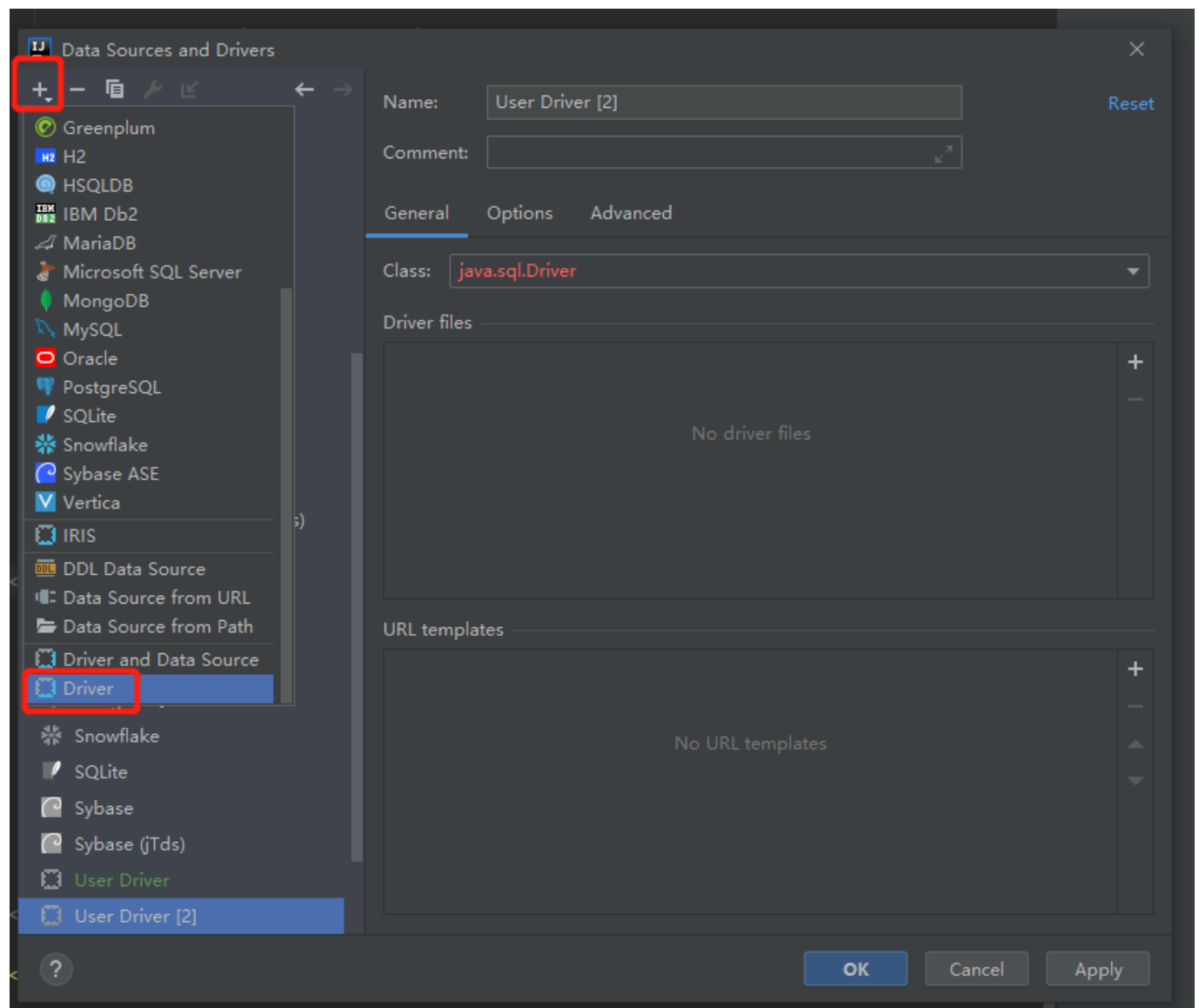
IDEA's database management tool connects to IRIS



Introduction of drive package

Tutorial - Develop IRIS using SSH

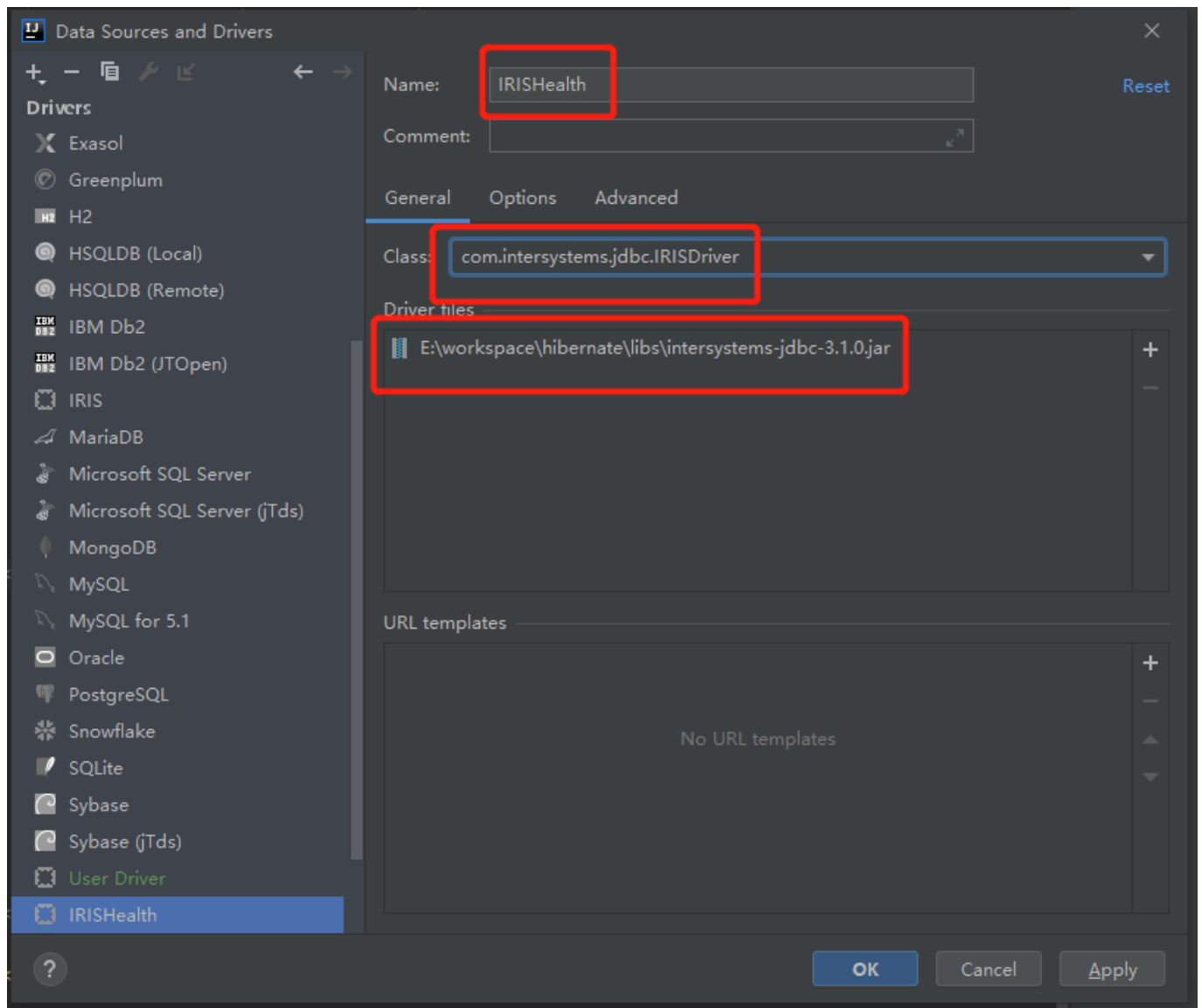
Published on InterSystems Developer Community (<https://community.intersystems.com>)



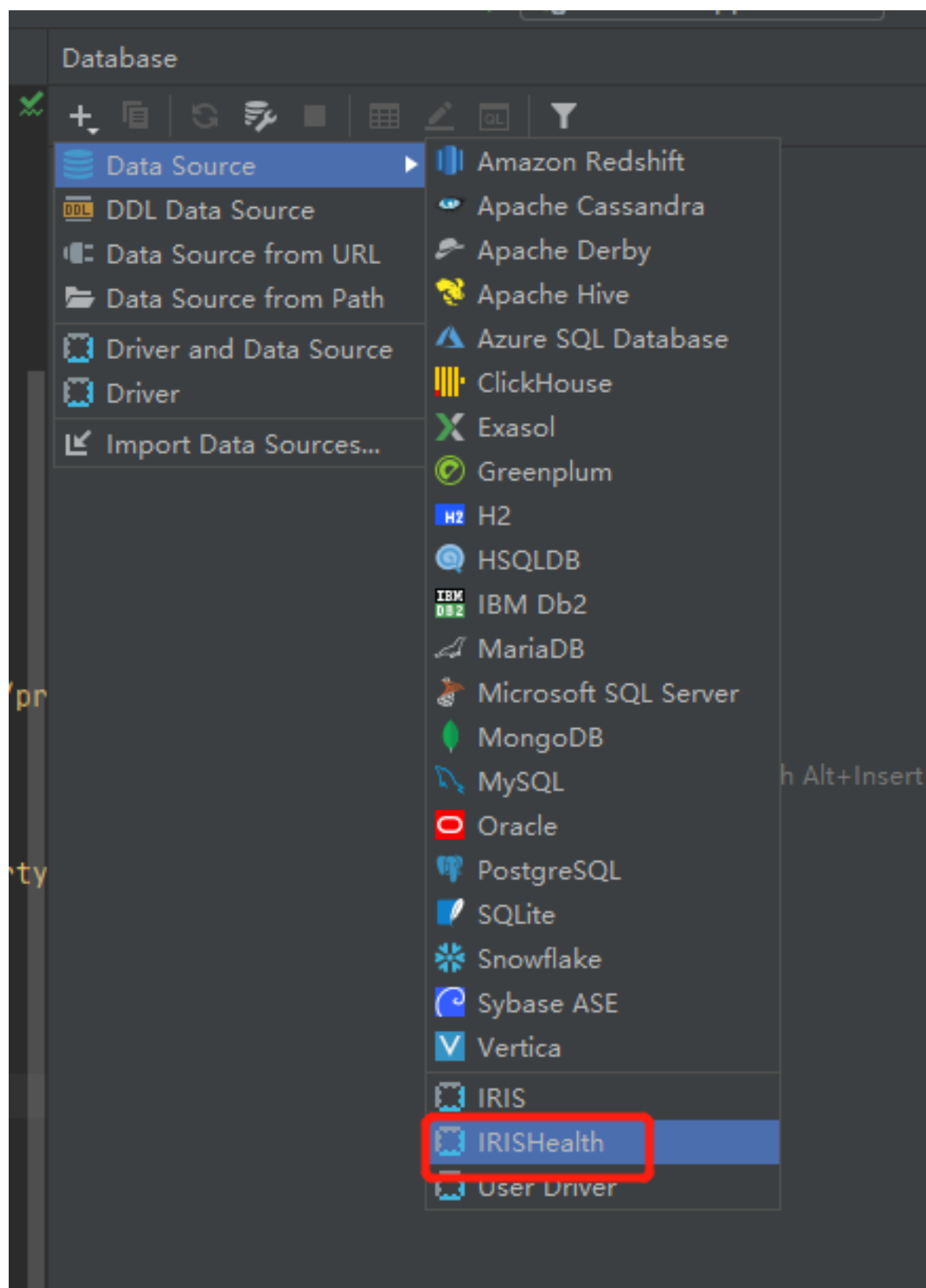
Published on InterSystems Developer Community (<https://community.intersystems.com>)

Tutorial - Develop IRIS using SSH

Published on InterSystems Developer Community (<https://community.intersystems.com>)

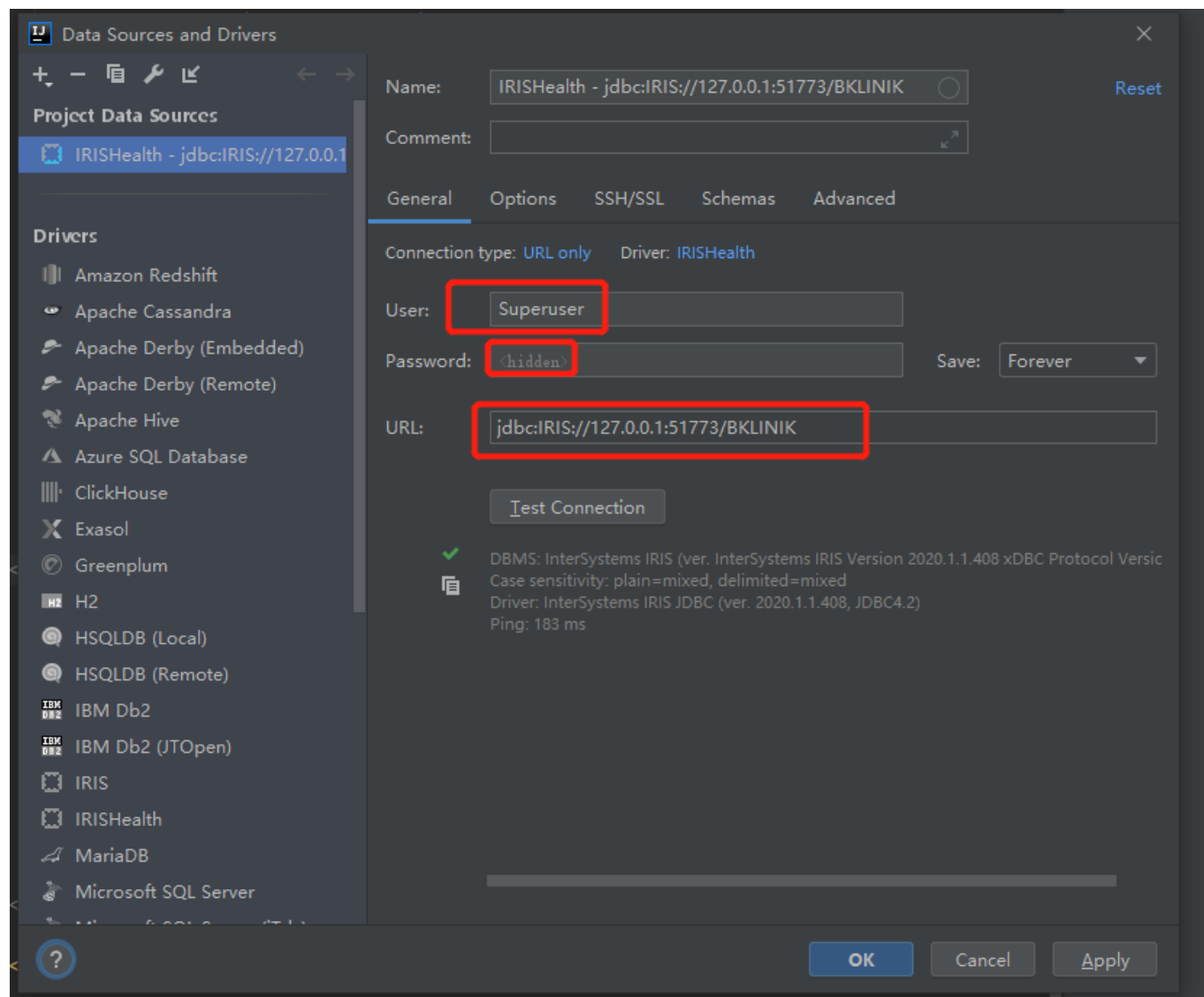


Connect to IRIS database



Tutorial - Develop IRIS using SSH

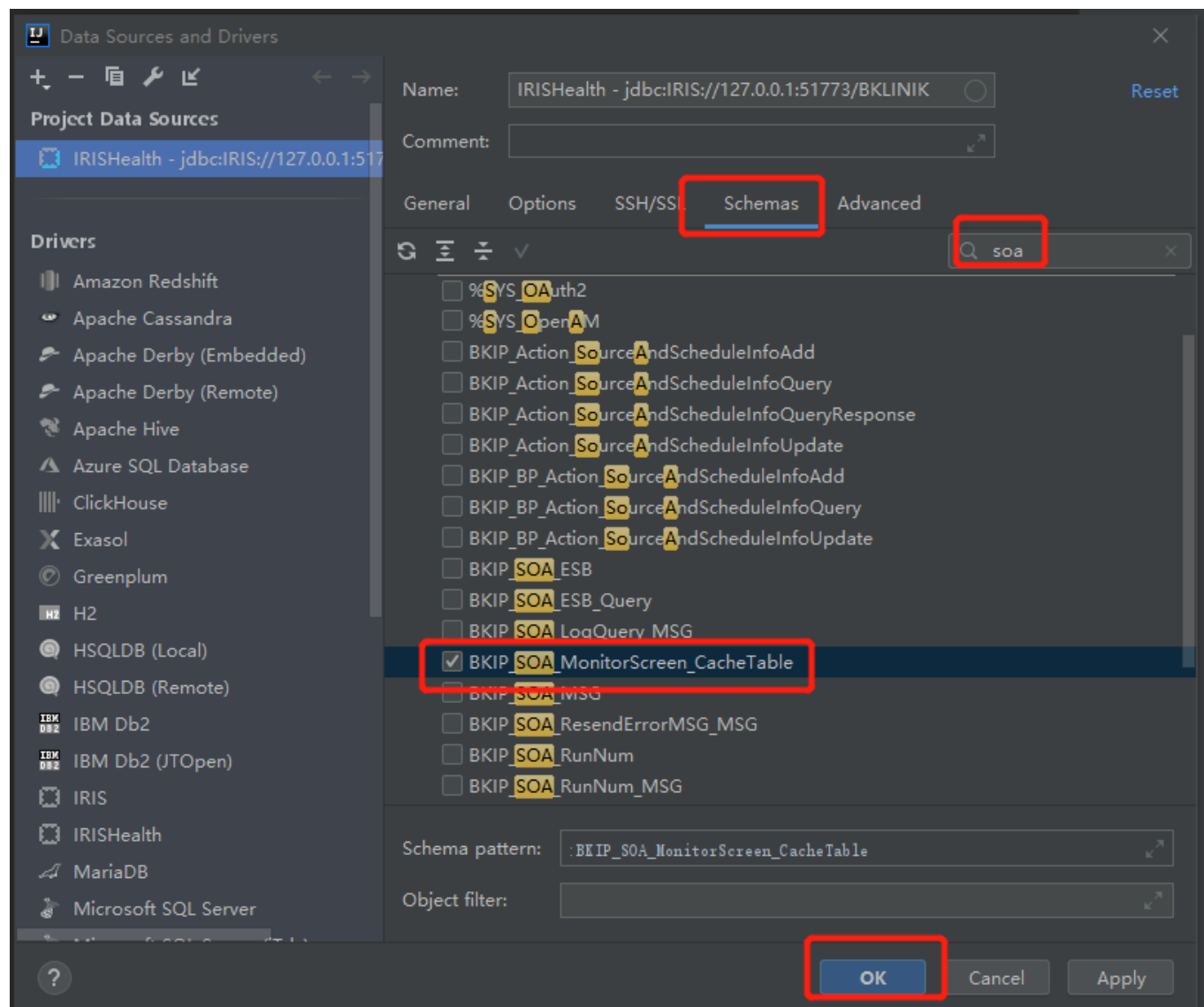
Published on InterSystems Developer Community (<https://community.intersystems.com>)



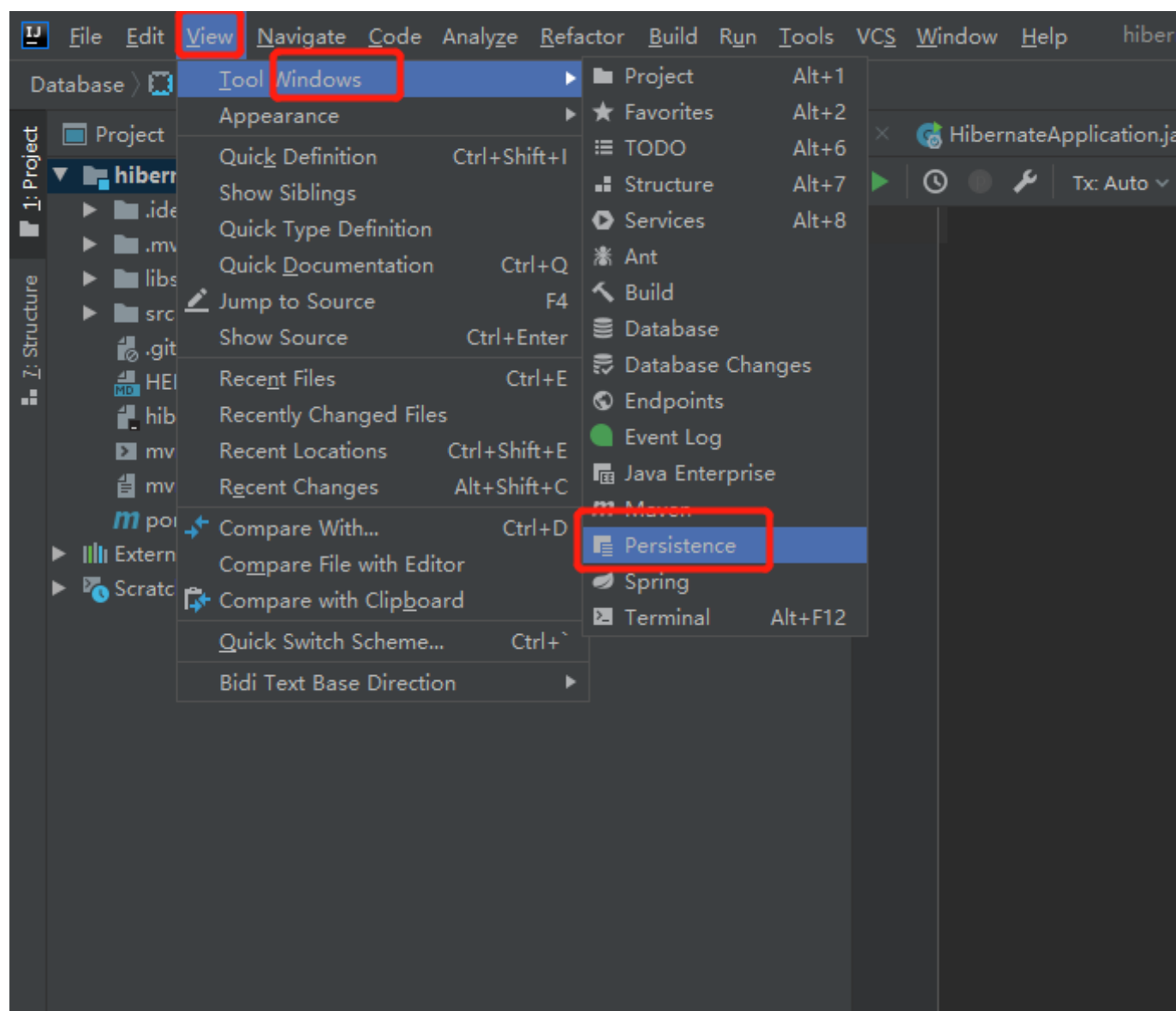
Select the required database

Tutorial - Develop IRIS using SSH

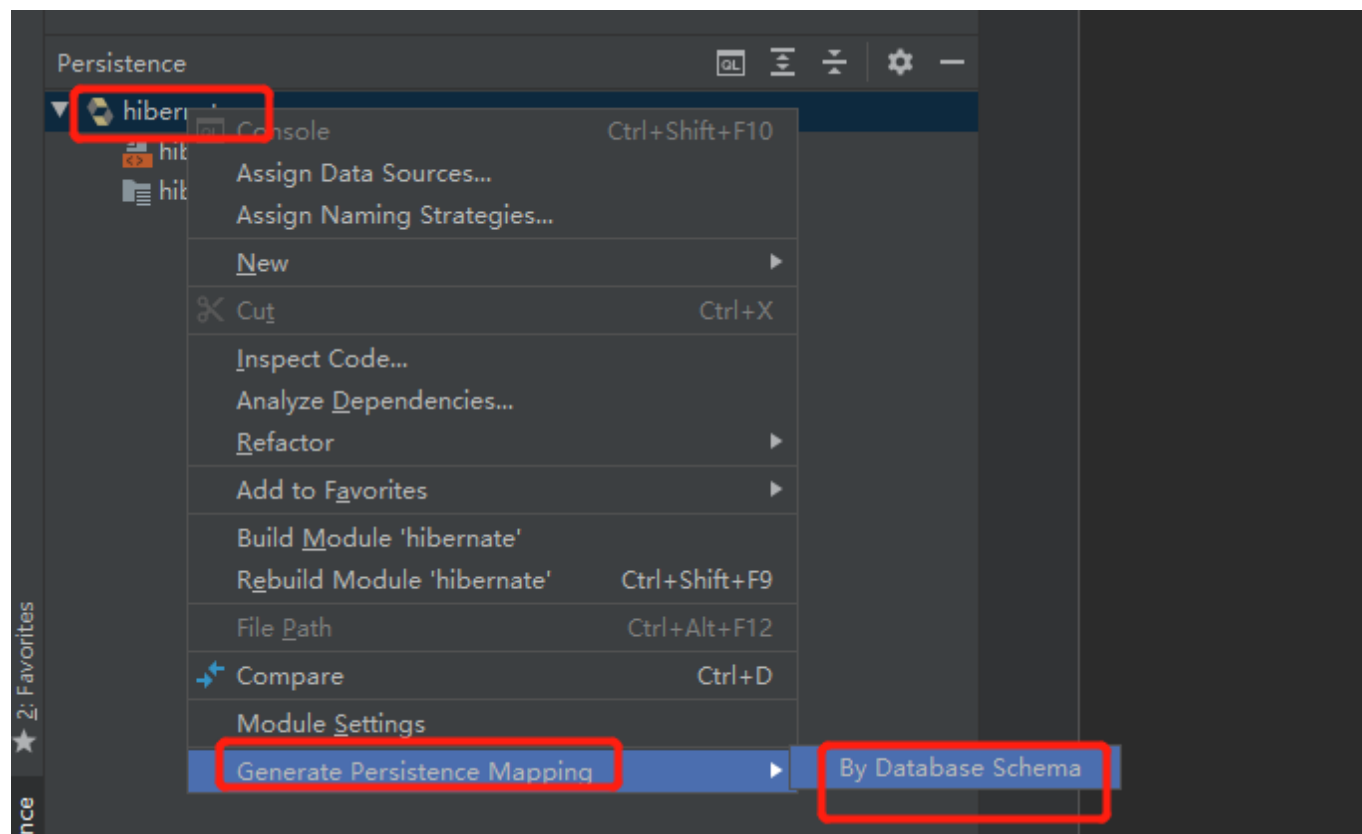
Published on InterSystems Developer Community (<https://community.intersystems.com>)



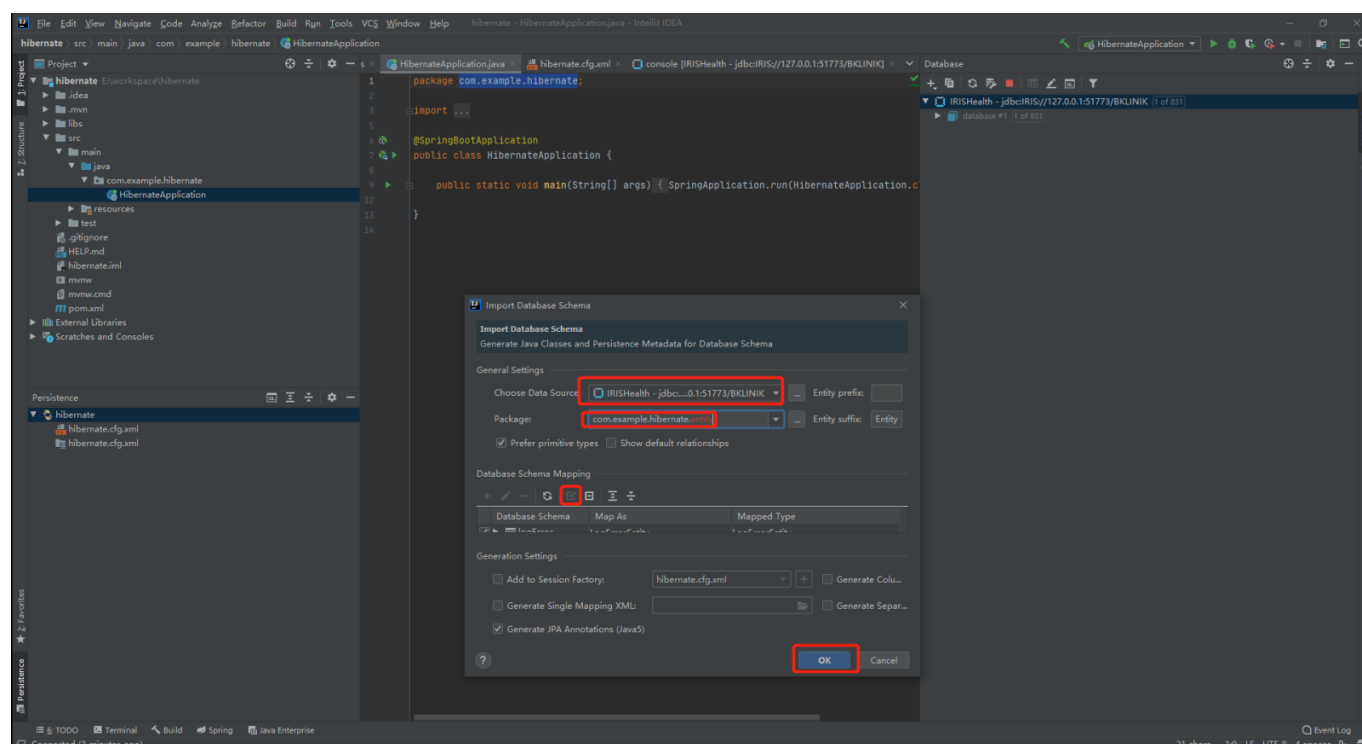
Open Hibernate tool, View -- Persistence



Open the mapping, as shown:



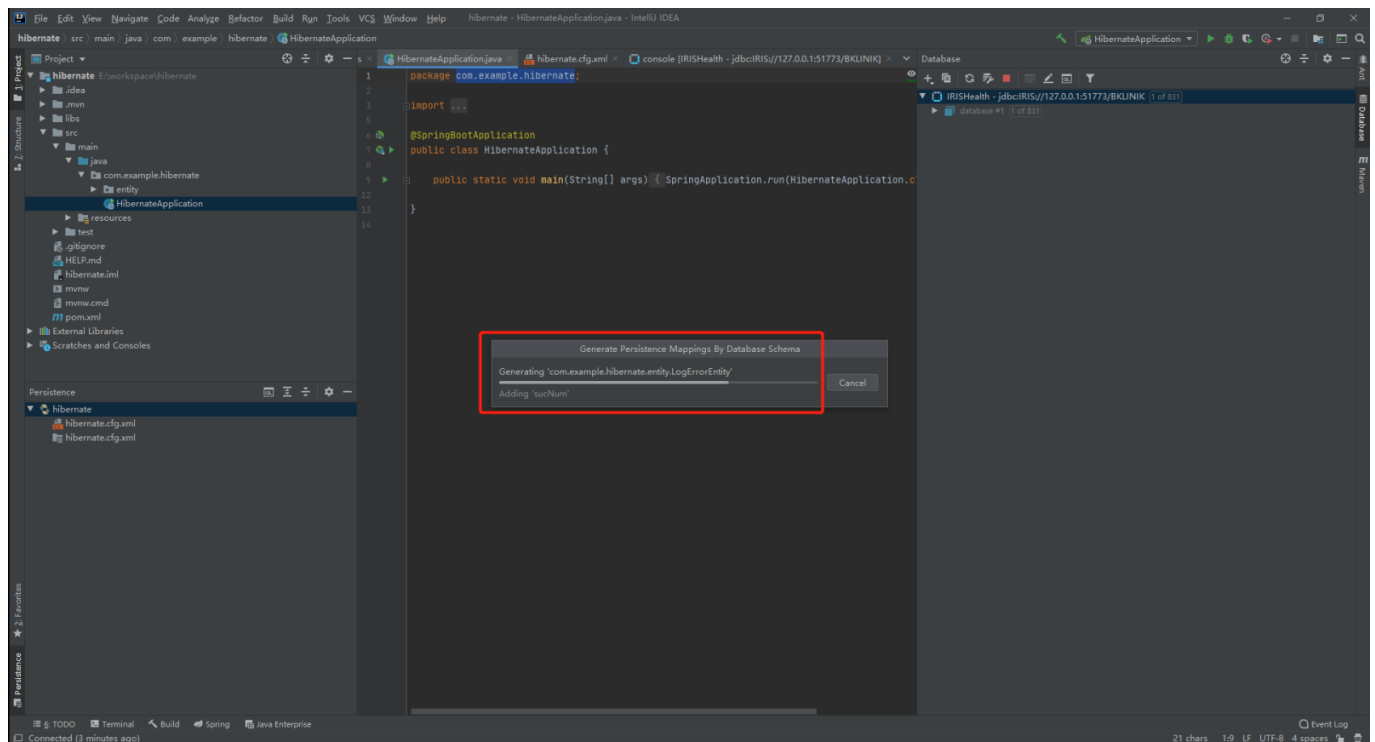
Select Connect and enter the registration. Because the computer display is incomplete, select all.



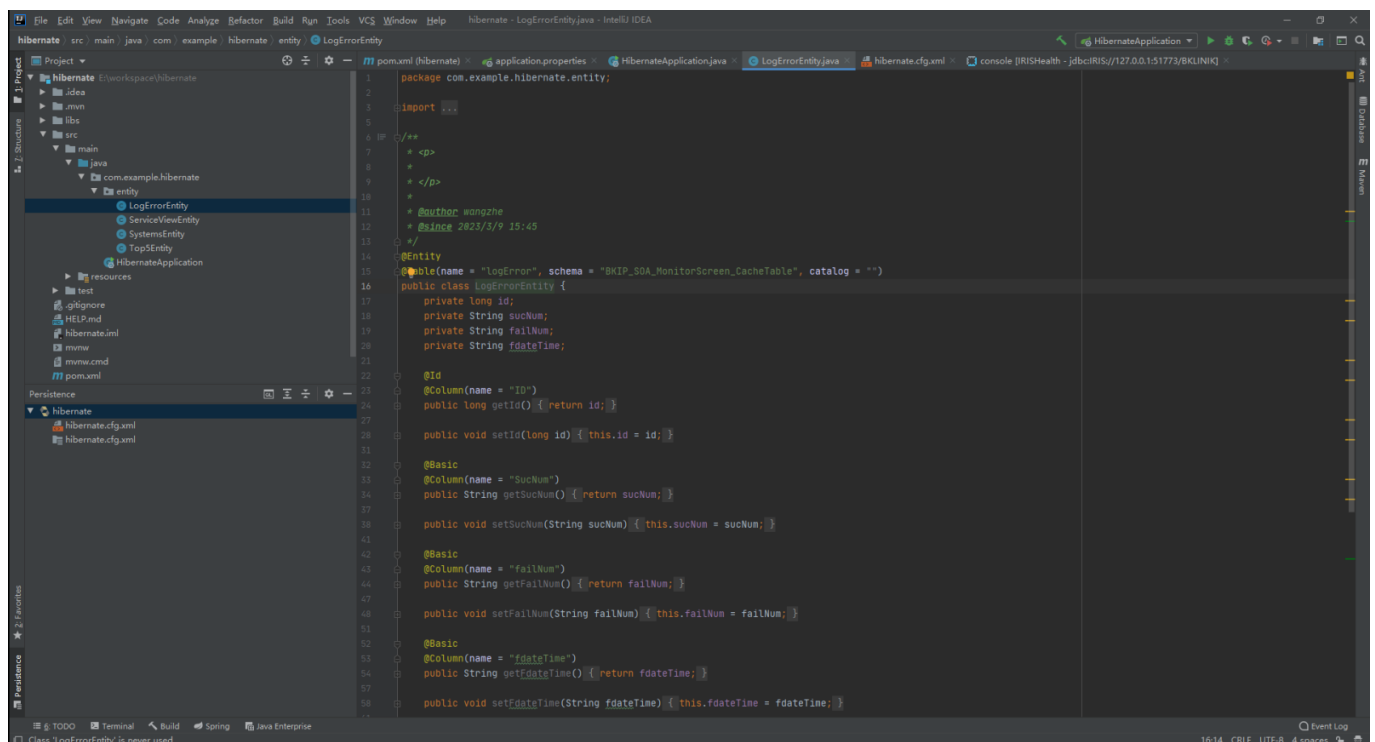
Next, click OK until you succeed:

Tutorial - Develop IRIS using SSH

Published on InterSystems Developer Community (<https://community.intersystems.com>)



You get a bunch of entity classes:



Now that the entity class has been created, the rest of the logic of creating a session factory to add, delete, and check is not shown here. Share a tool class I used for reference only

```
package com.example.hibernate.utils;

import org.hibernate.HibernateException;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.boot.MetadataSources;
```

```
import org.hibernate.cfg.Configuration;
import org.hibernate.service.ServiceRegistry;

/**
 * <p>
 * Hibernate Tools
 * </p>
 *
 * @author wangzhe
 * @since 2017/3/9 14:42
 */
public class HibernateUtil {

    private static final
    ThreadLocal<Session> threadLocal = new ThreadLocal<Session>();
    private static SessionFactory sessionFactory = null;

    static {
        try {
            Configuration cfg = new Configuration().configure();
            ServiceRegistry serviceRegistry = cfg.getStandardServiceRegistryBuilder().bu
            ild();
            sessionFactory = new
            MetadataSources(serviceRegistry).buildMetadata().buildSessionFactory();
        } catch (Exception e) {
            System.err.println("Failed to create session factory");
            e.printStackTrace();
        }
    }

    /**
     * Get Session
     *
     * @return Session
     * @throws HibernateException
     */
    public static Session getSession() throws HibernateException {
        Session session = (Session) threadLocal.get();
        if (session == null || !session.isOpen()) {
            if (sessionFactory == null) {
                rebuildSessionFactory();
            }
            session = (sessionFactory != null) ? sessionFactory.openSession() : null;
            threadLocal.set(session);
        }

        return session;
    }

    /**
     * Rebuild session factory
     */
    public static void rebuildSessionFactory() {
        try {
            Configuration cfg = new Configuration().configure();
            ServiceRegistry serviceRegistry = cfg.getStandardServiceRegistryBuilder().bu
            ild();
            sessionFactory = new
            MetadataSources(serviceRegistry).buildMetadata().buildSessionFactory();
        }
    }
}
```

```
        } catch (Exception e) {
            System.err.println("Failed to create session factory");
            e.printStackTrace();
        }
    }

    /**
     * Get SessionFactory object
     *
     * @return SessionFactory object
     */
    public static SessionFactory getSessionFactory() {
        return sessionFactory;
    }

    /**
     * Close Session
     *
     * @throws HibernateException
     */
    public static void closeSession() throws HibernateException {
        Session session = (Session) threadLocal.get();
        threadLocal.set(null);
        if (session != null) {
            session.close();
        }
    }
}
```

Conclusion:

The above is a step that I showed how to build an environment and continue to develop IRIS as a data source in combination with the SSH framework. Although the content shown is relatively simple, if I follow this way, in theory, I can use Java to develop and use IRIS, and I can use a series of native Java methods to make it more convenient for Java developers to use the powerful performance of IRIS and complete more businesses, At the same time, if you want to use SpringBoot or even SpringCloud to develop IRIS, it is not impossible. The key is the role of IRIS in this architecture.

[#Java](#) [#JDBC](#) [#Tutorial](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)

Source URL: <https://community.intersystems.com/post/tutorial-develop-iris-using-ssh>