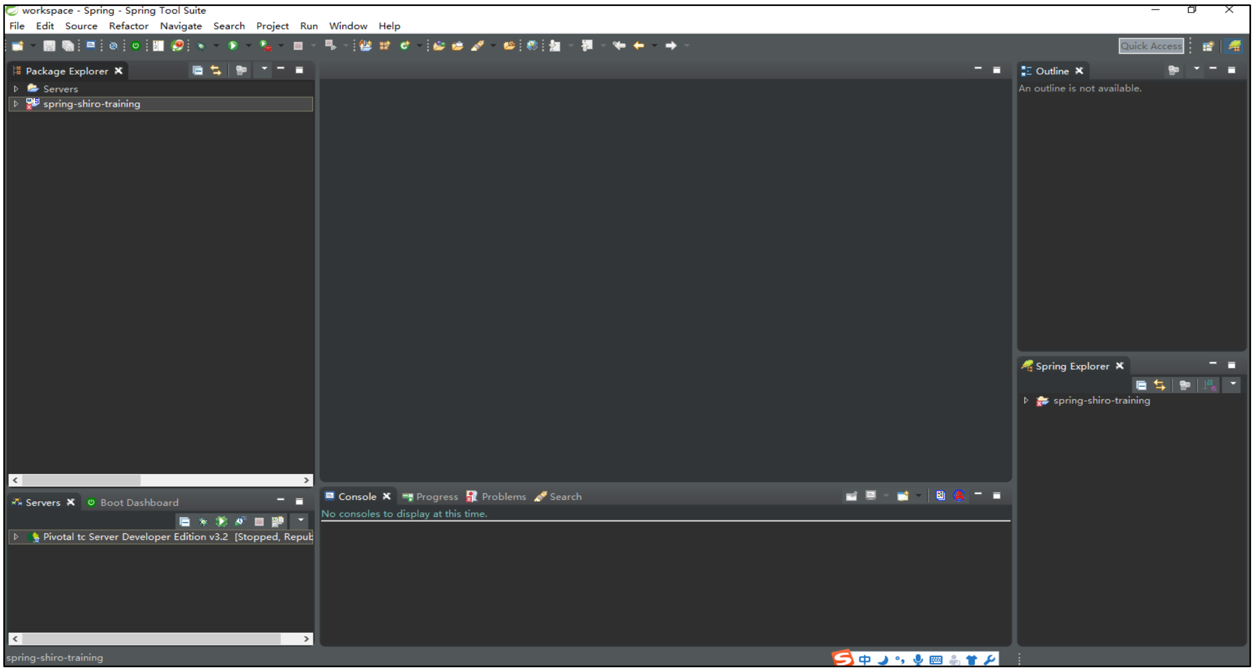
IDE 环境准备

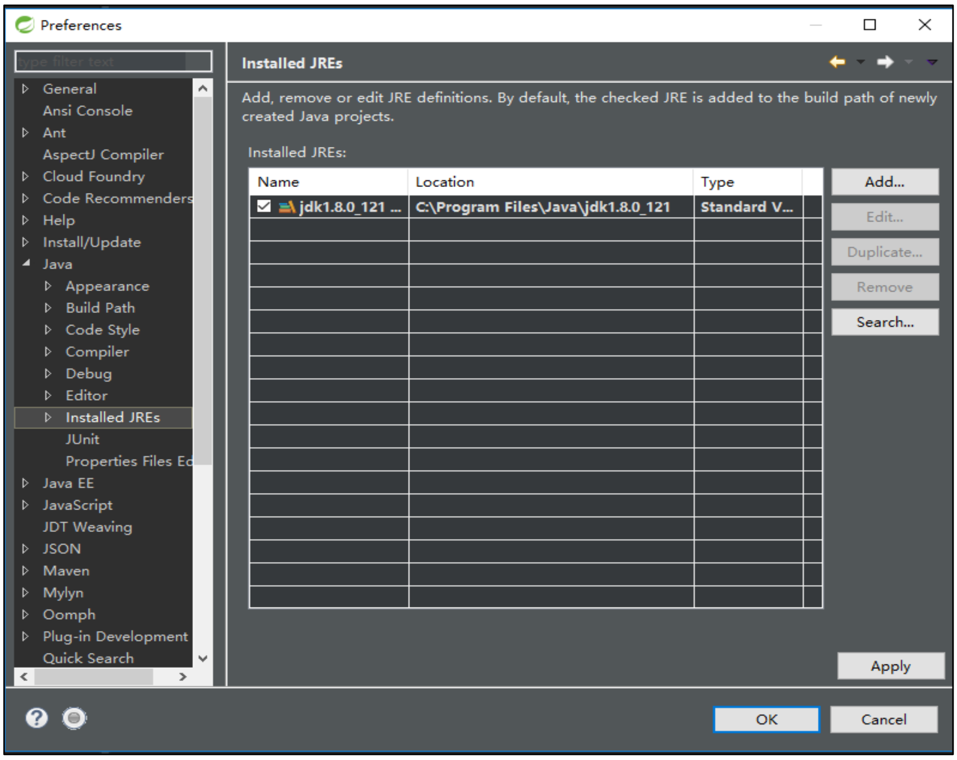
开发环境IDE选用Spring Tool Suite (STS3.8.4)
http://spring.io/tools/sts



An outline is not available.

No consoles to display at this time.

JDK1.8.0_121



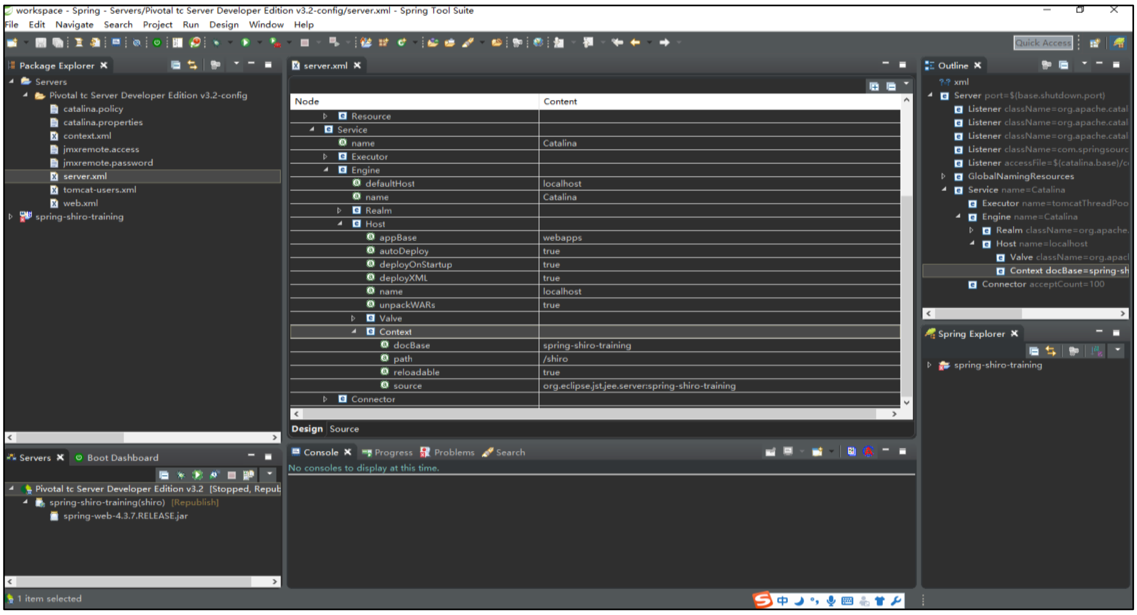
Installed JREs

Add, remove or edit JRE definitions. By default, the checked JRE is added to the build path of newly created Java projects.

Installed JREs:

Name	Location	Type
☑ jdk1.8.0_121 ...	C:\Program Files\Java\jdk1.8.0_121	Standard V...

Web Server用STS自带的 Pivotal tc Server


workspace - Spring - Servers/Pivotal tc Server Developer Edition v3.2-config/server.xml - Spring Tool Suite
File Edit Navigate Search Project Run Design Window Help
Quick Access
Package Explorer
server.xml
Outline
Spring Explorer
Design Source
Console Progress Problems Search
Servers Boot Dashboard

```
Servers
  Pivotal tc Server Developer Edition v3.2-config
    catalina.policy
    catalina.properties
    context.xml
    jmxremote.access
    jmxremote.password
    server.xml
    tomcat-users.xml
    web.xml
spring-shiro-training
```

Node	Content
Resource	
Service	
name	Catalina
Executor	
Engine	
defaultHost	localhost
name	Catalina
Realm	
Host	
appBase	webapps
autoDeploy	true
deployOnStartup	true
deployXML	true
name	localhost
unpackWARs	true
Valve	
Context	
docBase	spring-shiro-training
path	/shiro
reloadable	true
source	org.eclipse.jst.jee.server:spring-shiro-training
Connector	

```
xml
Server port=${base.shutdown.port}
  Listener className=org.apache.catal
  Listener className=org.apache.catal
  Listener className=org.apache.catal
  Listener className=com.springsourc
  Listener accessFile=${catalina.base}/c
  GlobalNamingResources
  Service name=Catalina
    Executor name=tomcatThreadPoo
    Engine name=Catalina
      Realm className=org.apache.
      Host name=localhost
        Valve className=org.apacl
        Context docBase=spring-sh
    Connector acceptCount=100
```

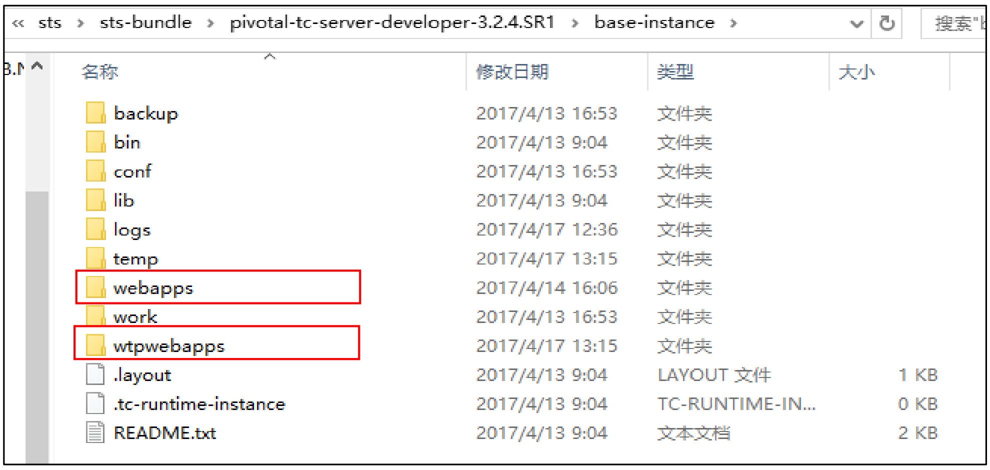
spring-shiro-training

No consoles to display at this time.

```
Pivotal tc Server Developer Edition v3.2  [Stopped, Repub
  spring-shiro-training(shiro)  [Republish]
    spring-web-4.3.7.RELEASE.jar
```

1 item selected

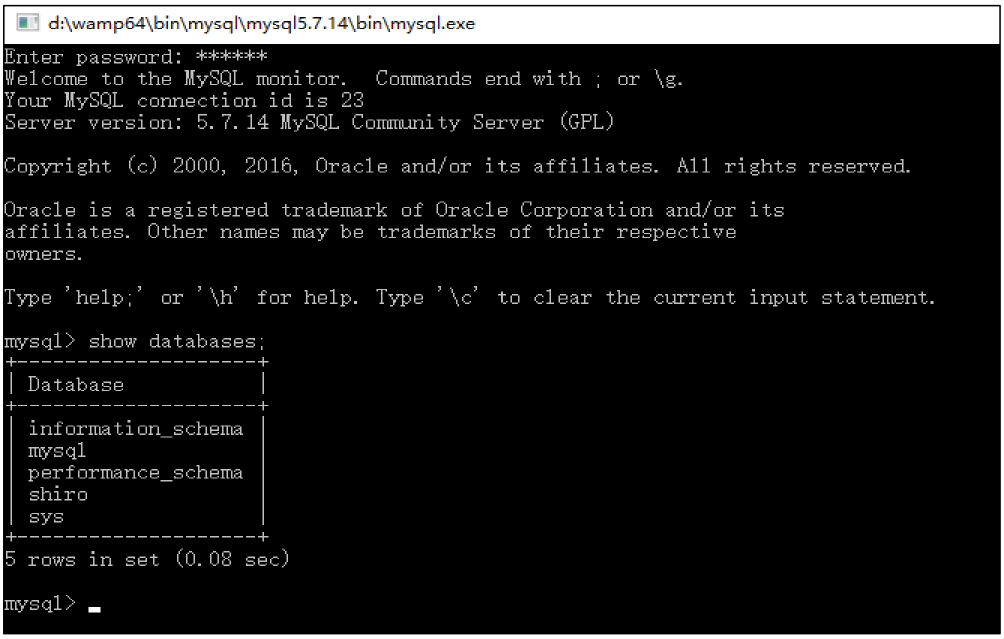
Root目录在STS的安装目录下


« sts › sts-bundle › pivotal-tc-server-developer-3.2.4.SR1 › base-instance ›
搜索"l

名称	修改日期	类型	大小
backup	2017/4/13 16:53	文件夹	
bin	2017/4/13 9:04	文件夹	
conf	2017/4/13 16:53	文件夹	
lib	2017/4/13 9:04	文件夹	
logs	2017/4/17 12:36	文件夹	
temp	2017/4/17 13:15	文件夹	
webapps	2017/4/14 16:06	文件夹	
work	2017/4/13 16:53	文件夹	
wtpwebapps	2017/4/17 13:15	文件夹	
.layout	2017/4/13 9:04	LAYOUT 文件	1 KB
.tc-runtime-instance	2017/4/13 9:04	TC-RUNTIME-IN...	0 KB
README.txt	2017/4/13 9:04	文本文档	2 KB

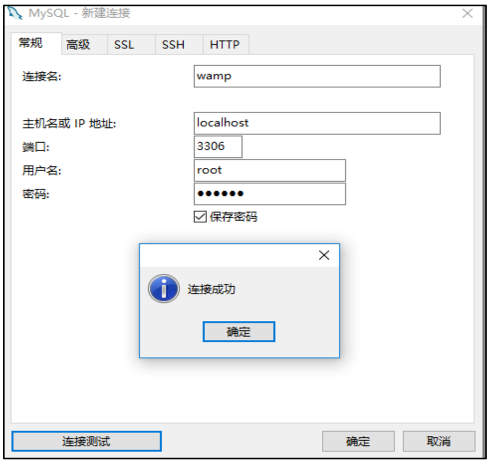
MySQL

MySQL 5.7.14 Community Server（GPL）



```
Enter password: ******
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 23
Server version: 5.7.14 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| shiro              |
| sys                |
+--------------------+
5 rows in set (0.08 sec)

mysql> _
```

Navicat Lite



连接名: wamp

主机名或 IP 地址: localhost

端口: 3306

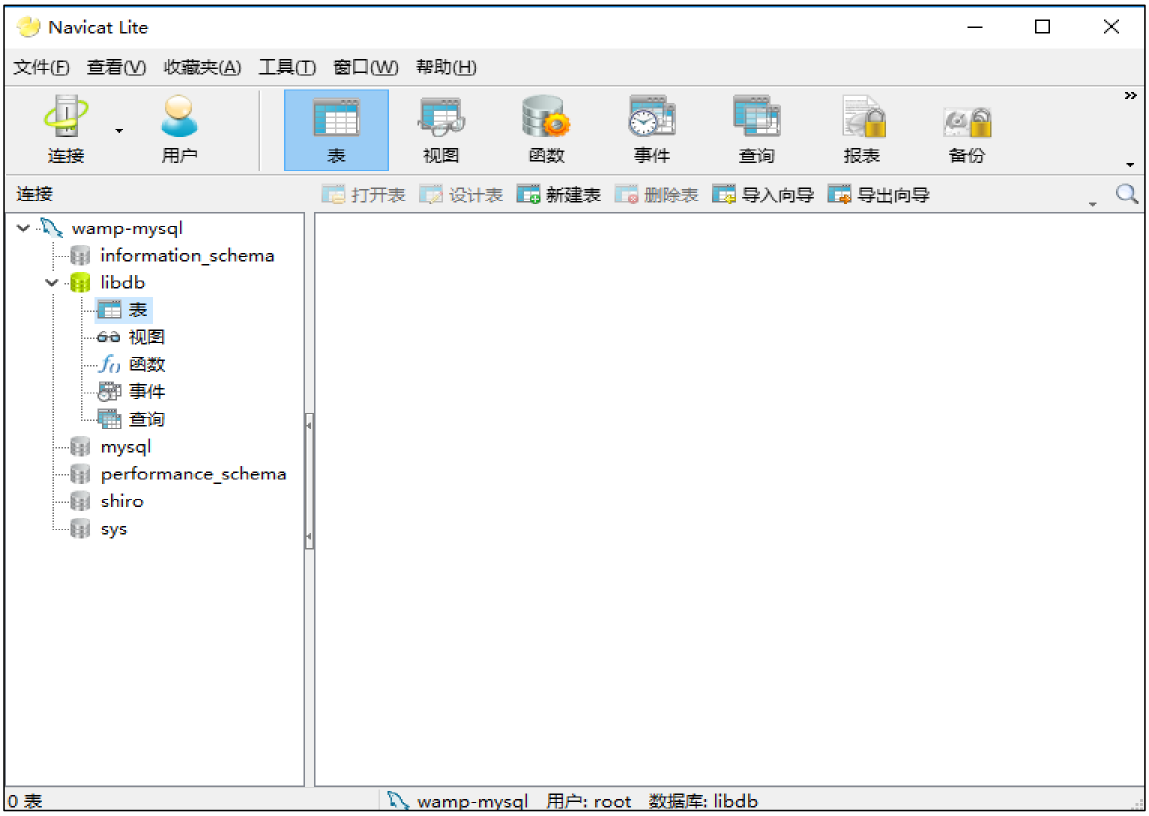
用户名: root

密码: ••••••

☑ 保存密码

连接成功

确定

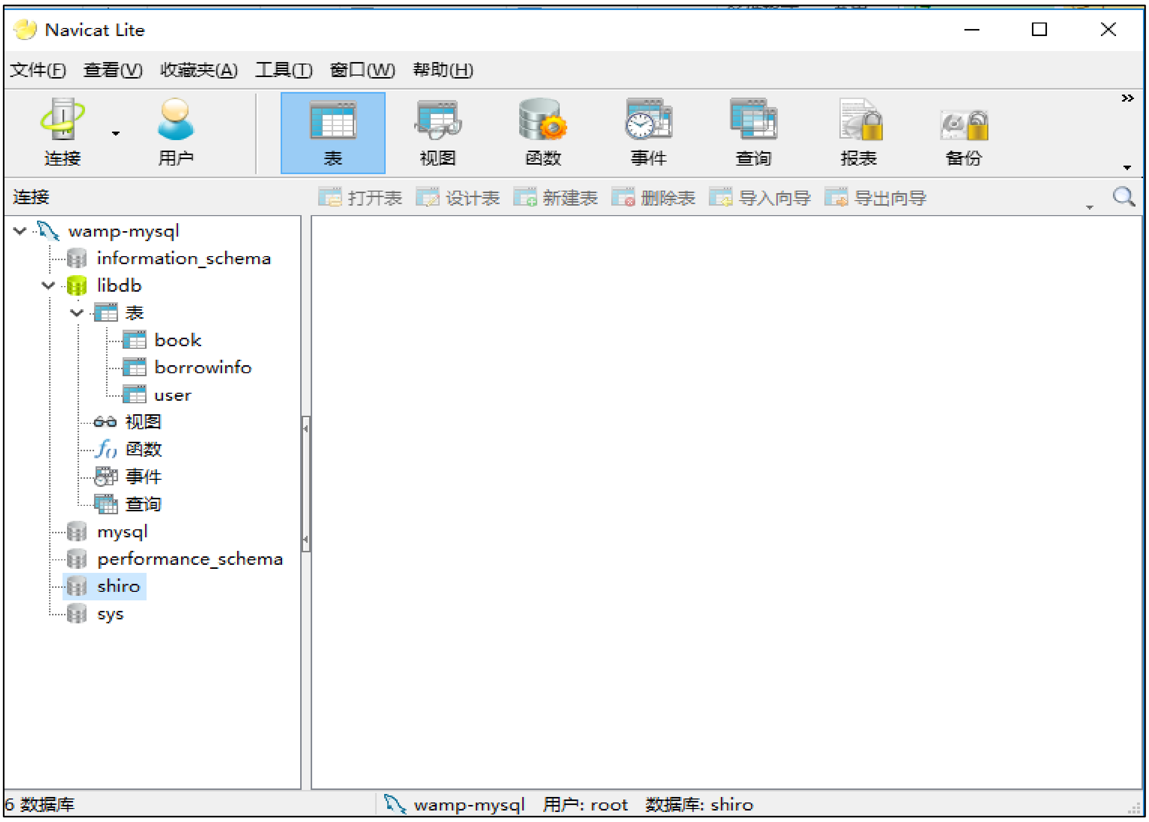


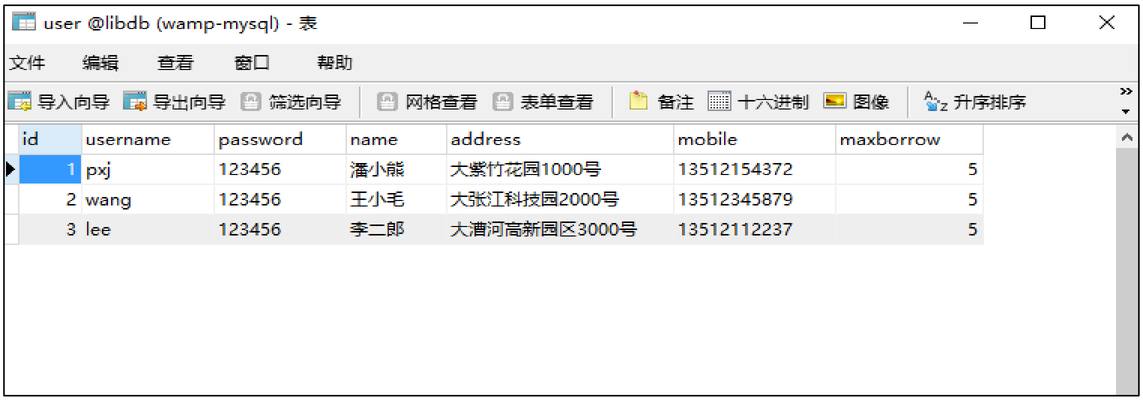
SQL 导入数据表及数据

```
DROP TABLE IF EXISTS `book`;

CREATE TABLE `book` (
  `id` int(11) NOT NULL AUTO_INCREMENT COMMENT '主键id',
  `name` varchar(150) NOT NULL COMMENT '书名',
  `author` varchar(50) NOT NULL COMMENT '作者',
  `publishing` varchar(50) NOT NULL COMMENT '出版社',
  `class` varchar(40) NOT NULL COMMENT '图书分类',
  `ISBN` varchar(20) NOT NULL COMMENT '图书ISBN编号',
  `total` tinyint(2) NOT NULL DEFAULT '5' COMMENT '库存量',
  `borrowed` tinyint(2) NOT NULL DEFAULT '0' COMMENT '已借数量',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8 COMMENT='图书';

INSERT INTO `book` VALUES (NULL, '极简欧洲史', '宇宙', '世界图书出版社', '历史', '123-N111', '5', '1');
INSERT INTO `book` VALUES (NULL, '麦哲伦航海纪 ', '苏惠玲', '希望出版社', '历史', '123-N112', '10', '2');
INSERT INTO `book` VALUES (NULL, '未来简史', '尤瓦尔', '中信出版社', '历史', '123-N113', '8', '8');
```

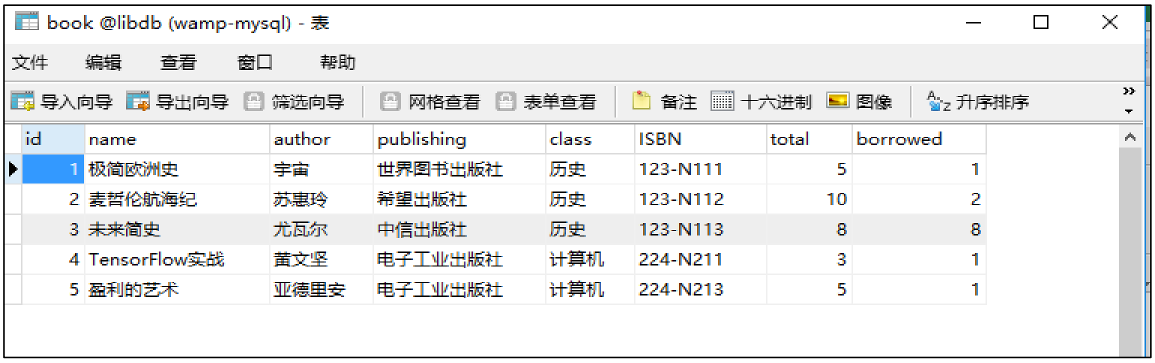
```
DROP TABLE IF EXISTS `user`;

CREATE TABLE `user` (
  `id` int(11) NOT NULL AUTO_INCREMENT COMMENT '主键id',
  `username` varchar(50) NOT NULL COMMENT '用户名',
  `password` varchar(100) NOT NULL COMMENT '密码',
  `name` varchar(20) NOT NULL COMMENT '姓名',
  `address` varchar(200) NOT NULL COMMENT '住址',
  `mobile` varchar(20) NOT NULL COMMENT '手机号码',
  `maxborrow` tinyint(2) NOT NULL DEFAULT '5' COMMENT '最大允许借书量',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8 COMMENT='读者信息';

INSERT INTO `user` VALUES (NULL, 'pxj', '123456', '潘小熊', '大紫竹花园1000号', '13512154372', '5');
```

```
DROP TABLE IF EXISTS `borrowinfo`;

CREATE TABLE `borrowinfo` (
 `id` int(11) NOT NULL AUTO_INCREMENT COMMENT '主键id',
 `userid` int(11) NOT NULL COMMENT '借阅用户ID',
 `bookid` int(11) NOT NULL COMMENT '借阅图书ID',
 `borrowdate` datetime NOT NULL COMMENT '借阅时间',
 `returndate` datetime NOT NULL COMMENT '归还时间',
 `returnflag` tinyint(2) NOT NULL DEFAULT '0' COMMENT '归还状态标志',
 `expireflag` tinyint(2) NOT NULL DEFAULT '0' COMMENT '过期状态标志',
 PRIMARY KEY (`id`),
 FOREIGN KEY (`userid`) REFERENCES user(`id`),
 FOREIGN KEY (`bookid`) REFERENCES book(`id`)
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8 COMMENT='借阅信息明细';

INSERT INTO `borrowinfo` VALUES (NULL, '1', '1', '2017-4-6 10:27:55', '2017-5-6 23:59:59', '0', '0');
INSERT INTO `borrowinfo` VALUES (NULL, '1', '2', '2017-4-6 10:27:55', '2017-5-6 23:59:59', '0', '0');
```

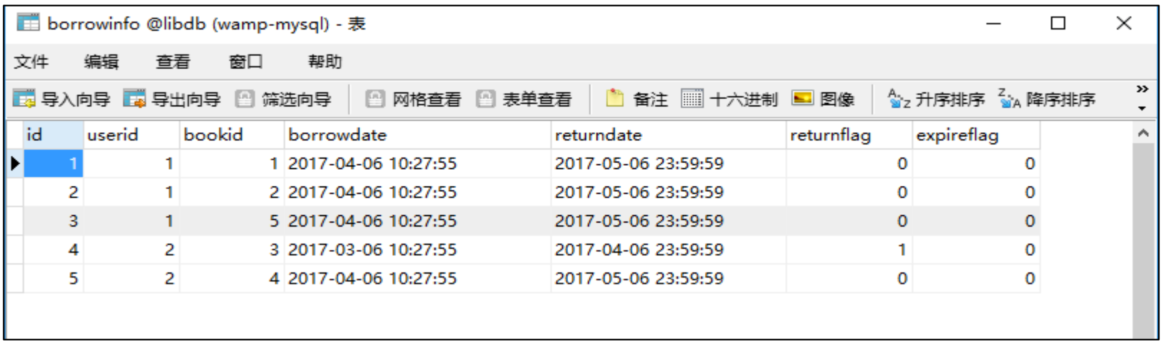



user @libdb (wamp-mysql) - 表
文件 编辑 查看 窗口 帮助
导入向导 导出向导 筛选向导 网格查看 表单查看 备注 十六进制 图像 升序排序

id	username	password	name	address	mobile	maxborrow
1	pxj	123456	潘小熊	大紫竹花园1000号	13512154372	5
2	wang	123456	王小毛	大张江科技园2000号	13512345879	5
3	lee	123456	李二郎	大漕河高新园区3000号	13512112237	5



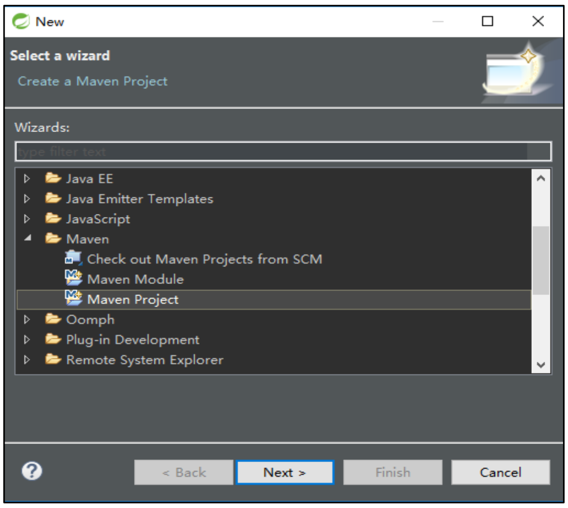
id	name	author	publishing	class	ISBN	total	borrowed
1	极简欧洲史	宇宙	世界图书出版社	历史	123-N111	5	1
2	麦哲伦航海纪	苏惠玲	希望出版社	历史	123-N112	10	2
3	未来简史	尤瓦尔	中信出版社	历史	123-N113	8	8
4	TensorFlow实战	黄文坚	电子工业出版社	计算机	224-N211	3	1
5	盈利的艺术	亚德里安	电子工业出版社	计算机	224-N213	5	1



id	userid	bookid	borrowdate	returndate	returnflag	expireflag
1	1	1	2017-04-06 10:27:55	2017-05-06 23:59:59	0	0
2	1	2	2017-04-06 10:27:55	2017-05-06 23:59:59	0	0
3	1	5	2017-04-06 10:27:55	2017-05-06 23:59:59	0	0
4	2	3	2017-03-06 10:27:55	2017-04-06 23:59:59	1	0
5	2	4	2017-04-06 10:27:55	2017-05-06 23:59:59	0	0

Maven Project 配置

创建Maven Project



New

Select a wizard

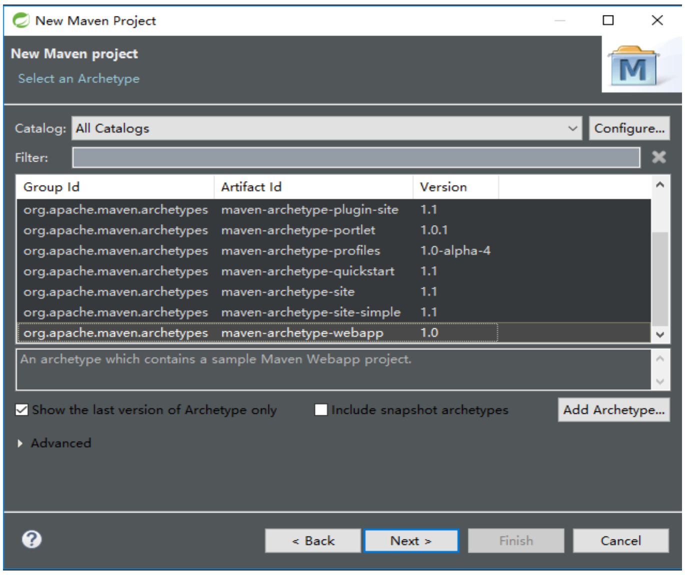
Create a Maven Project

Wizards:

type filter text

- Java EE
- Java Emitter Templates
- JavaScript
- Maven
  - Check out Maven Projects from SCM
  - Maven Module
  - Maven Project
- Oomph
- Plug-in Development
- Remote System Explorer

选择： maven-archetype-webapp 1.0



New Maven Project

New Maven project

Select an Archetype

Catalog: All Catalogs

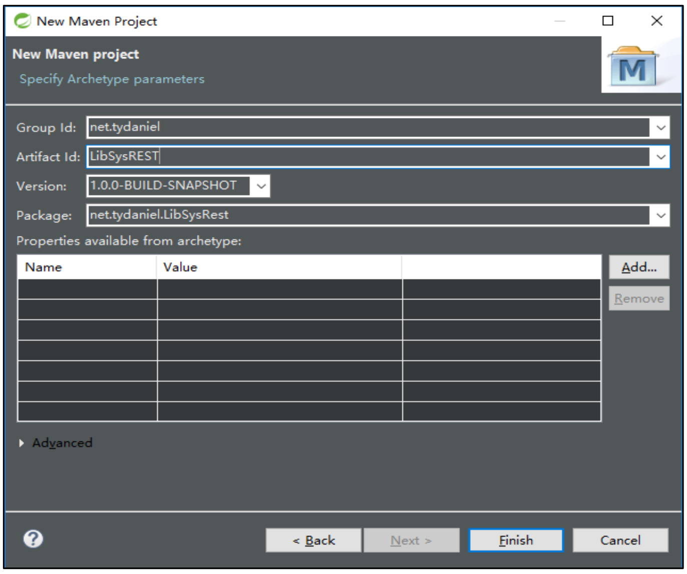
Filter:

Group Id	Artifact Id	Version
org.apache.maven.archetypes	maven-archetype-plugin-site	1.1
org.apache.maven.archetypes	maven-archetype-portlet	1.0.1
org.apache.maven.archetypes	maven-archetype-profiles	1.0-alpha-4
org.apache.maven.archetypes	maven-archetype-quickstart	1.1
org.apache.maven.archetypes	maven-archetype-site	1.1
org.apache.maven.archetypes	maven-archetype-site-simple	1.1
org.apache.maven.archetypes	maven-archetype-webapp	1.0

An archetype which contains a sample Maven Webapp project.

☑ Show the last version of Archetype only　□ Include snapshot archetypes

Advanced



New Maven Project

New Maven project

Specify Archetype parameters

Group Id: net.tydaniel

Artifact Id: LibSysREST

Version: 1.0.0-BUILD-SNAPSHOT

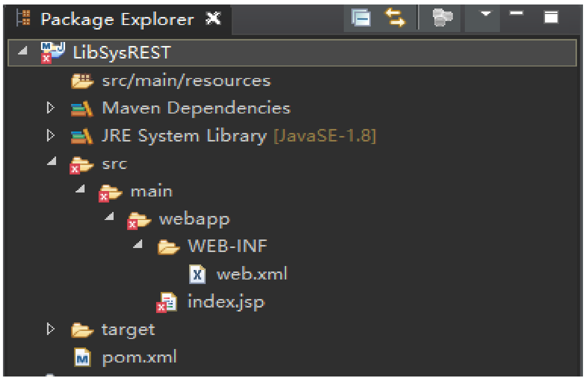
Package: net.tydaniel.LibSysRest

Properties available from archetype:

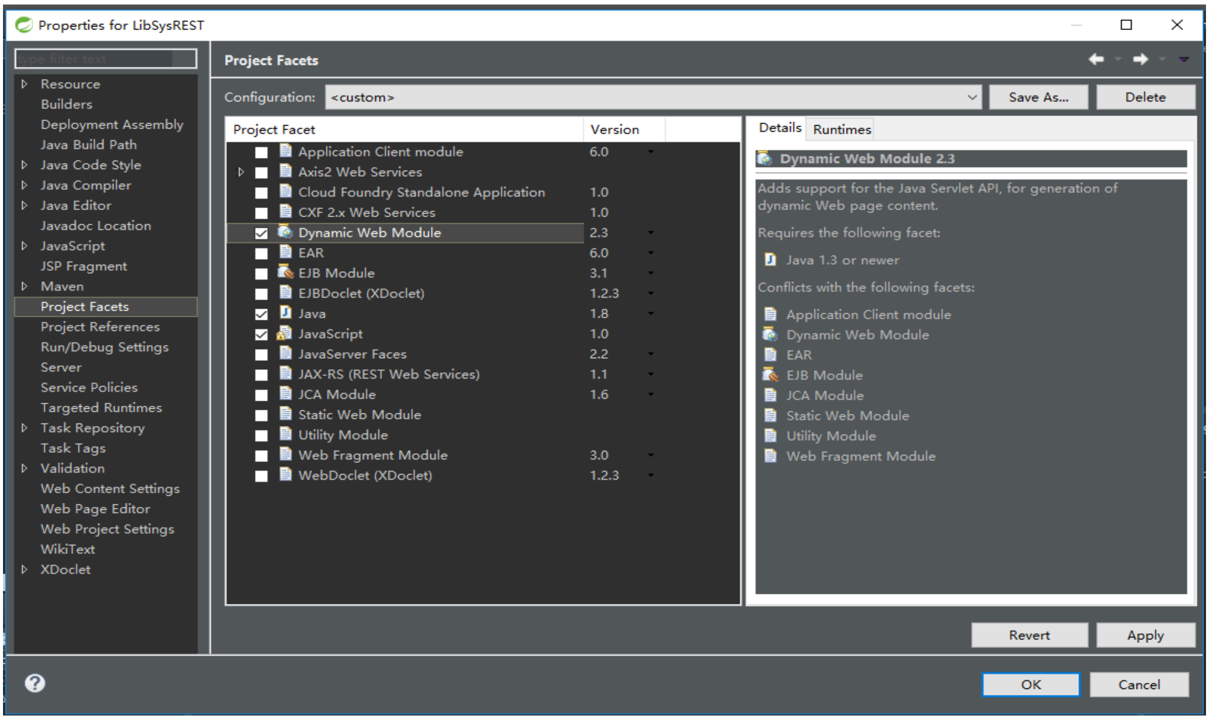
Name	Value

Advanced

创建后成功的Maven Project



```
Package Explorer
LibSysREST
  src/main/resources
  Maven Dependencies
  JRE System Library [JavaSE-1.8]
  src
    main
      webapp
        WEB-INF
          web.xml
        index.jsp
  target
  pom.xml
```



Properties for LibSysREST

```
Resource
Builders
Deployment Assembly
Java Build Path
Java Code Style
Java Compiler
Java Editor
Javadoc Location
JavaScript
JSP Fragment
Maven
Project Facets
Project References
Run/Debug Settings
Server
Service Policies
Targeted Runtimes
Task Repository
Task Tags
Validation
Web Content Settings
Web Page Editor
Web Project Settings
WikiText
XDoclet
```

Project Facets

Configuration: <custom>

Project Facet	Version
□ Application Client module	6.0
□ Axis2 Web Services	
□ Cloud Foundry Standalone Application	1.0
□ CXF 2.x Web Services	1.0
☑ Dynamic Web Module	2.3
□ EAR	6.0
□ EJB Module	3.1
□ EJBDoclet (XDoclet)	1.2.3
☑ Java	1.8
☑ JavaScript	1.0
□ JavaServer Faces	2.2
□ JAX-RS (REST Web Services)	1.1
□ JCA Module	1.6
□ Static Web Module	
□ Utility Module	
□ Web Fragment Module	3.0
□ WebDoclet (XDoclet)	1.2.3

Details Runtimes

Dynamic Web Module 2.3

Adds support for the Java Servlet API, for generation of dynamic Web page content.

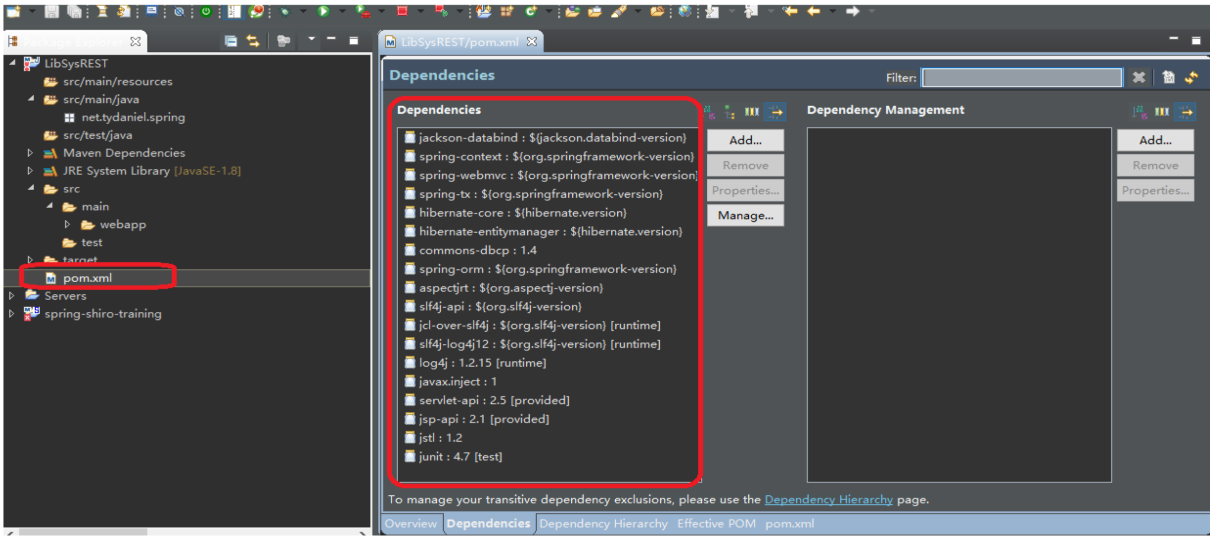
Requires the following facet:

Java 1.3 or newer

Conflicts with the following facets:

- Application Client module
- Dynamic Web Module
- EAR
- EJB Module
- JCA Module
- Static Web Module
- Utility Module
- Web Fragment Module

设置POM



```
Package Explorer
LibSysREST
  src/main/resources
  src/main/java
    net.tydaniel.spring
  src/test/java
  Maven Dependencies
  JRE System Library [JavaSE-1.8]
  src
    main
      webapp
    test
  target
  pom.xml
Servers
spring-shiro-training
```

LibSysREST/pom.xml

Dependencies

Filter:

Dependencies

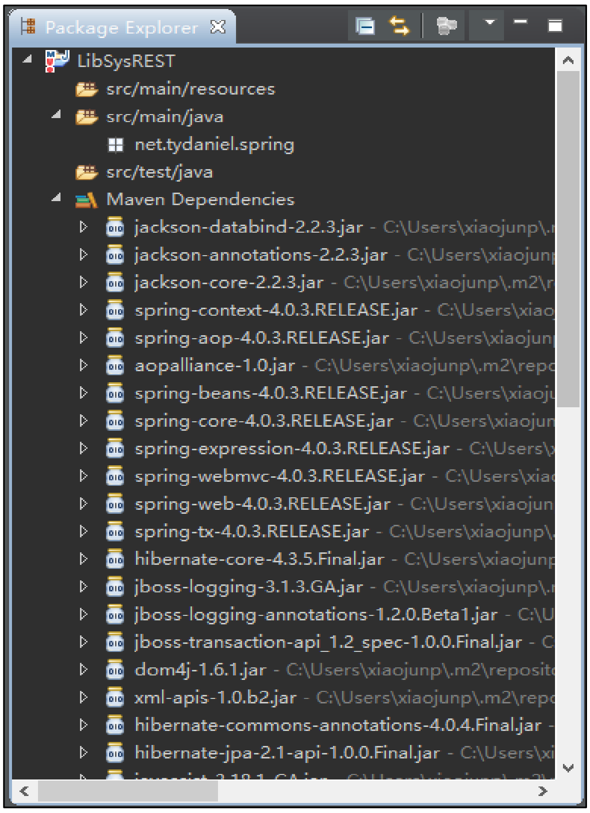
```
jackson-databind : ${jackson.databind-version}
spring-context : ${org.springframework-version}
spring-webmvc : ${org.springframework-version}
spring-tx : ${org.springframework-version}
hibernate-core : ${hibernate.version}
hibernate-entitymanager : ${hibernate.version}
commons-dbcp : 1.4
spring-orm : ${org.springframework-version}
aspectjrt : ${org.aspectj-version}
slf4j-api : ${org.slf4j-version}
jcl-over-slf4j : ${org.slf4j-version} [runtime]
slf4j-log4j12 : ${org.slf4j-version} [runtime]
log4j : 1.2.15 [runtime]
javax.inject : 1
servlet-api : 2.5 [provided]
jsp-api : 2.1 [provided]
jstl : 1.2
junit : 4.7 [test]
```

Dependency Management

To manage your transitive dependency exclusions, please use the Dependency Hierarchy page.

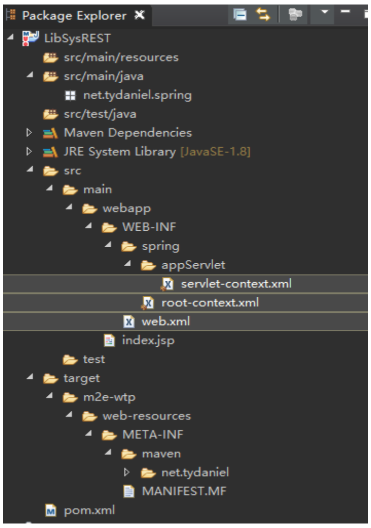
```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
 <modelVersion>4.0.0</modelVersion>
 <groupId>net.tydaniel</groupId>
 <artifactId>LibSysREST</artifactId>
 <packaging>war</packaging>
 <version>1.0.0-BUILD-SNAPSHOT</version>
 <name>LibSysREST</name>
 <url>http://maven.apache.org</url>


 <properties>
  <java-version>1.8</java-version>
  <org.springframework-version>4.0.3.RELEASE</org.springframework-version>
  <org.aspectj-version>1.7.4</org.aspectj-version>
  <org.slf4j-version>1.7.5</org.slf4j-version>
  <hibernate.version>4.3.5.Final</hibernate.version>
  <jackson.databind-version>2.2.3</jackson.databind-version>
 </properties>


 <dependencies>
  <!-- Jackson -->
<dependency>
 <groupId>com.fasterxml.jackson.core</groupId>
 <artifactId>jackson-databind</artifactId>
 <version>${jackson.databind-version}</version>
</dependency>

<!-- Spring -->
<dependency>
 <groupId>org.springframework</groupId>
 <artifactId>spring-context</artifactId>
 <version>${org.springframework-version}</version>
 <exclusions>
  <!-- Exclude Commons Logging in favor of SLF4j -->
<exclusion>
 <groupId>commons-logging</groupId>
 <artifactId>commons-logging</artifactId>
</exclusion>
 </exclusions>
</dependency>

<dependency>
 <groupId>org.springframework</groupId>
 <artifactId>spring-webmvc</artifactId>
 <version>${org.springframework-version}</version>
</dependency>

<dependency>
 <groupId>org.springframework</groupId>
 <artifactId>spring-tx</artifactId>
 <version>${org.springframework-version}</version>
</dependency>

<!-- Hibernate -->
<dependency>
 <groupId>org.hibernate</groupId>
 <artifactId>hibernate-core</artifactId>
 <version>${hibernate.version}</version>
</dependency>

<dependency>
 <groupId>org.hibernate</groupId>
 <artifactId>hibernate-entitymanager</artifactId>
 <version>${hibernate.version}</version>
 <version>${hibernate.version}</version>
</dependency>

<!-- Apache Commons DBCP -->
<dependency>
 <groupId>commons-dbcp</groupId>
 <artifactId>commons-dbcp</artifactId>
 <version>1.4</version>
</dependency>

<!-- Spring ORM -->
<dependency>
 <groupId>org.springframework</groupId>
 <artifactId>spring-orm</artifactId>
 <version>${org.springframework-version}</version>
</dependency>

<!-- AspectJ -->
<dependency>
 <groupId>org.aspectj</groupId>
 <artifactId>aspectjrt</artifactId>
 <version>${org.aspectj-version}</version>
</dependency>

<!-- Logging -->
<dependency>
 <groupId>org.slf4j</groupId>
 <artifactId>slf4j-api</artifactId>
 <version>${org.slf4j-version}</version>
</dependency>

<dependency>
 <groupId>org.slf4j</groupId>
 <artifactId>jcl-over-slf4j</artifactId>
 <version>${org.slf4j-version}</version>
 <scope>runtime</scope>
</dependency>

<dependency>
 <groupId>org.slf4j</groupId>
 <artifactId>slf4j-log4j12</artifactId>
 <version>${org.slf4j-version}</version>
 <scope>runtime</scope>
</dependency>

<dependency>
 <groupId>log4j</groupId>
 <artifactId>log4j</artifactId>
 <version>1.2.15</version>
 <exclusions>
<exclusion>
 <groupId>javax.mail</groupId>
 <artifactId>mail</artifactId>
  </exclusion>
  <exclusion>
 <groupId>javax.jms</groupId>
 <artifactId>jms</artifactId>
  </exclusion>
  <exclusion>
 <groupId>com.sun.jdmk</groupId>
 <artifactId>jmxtools</artifactId>
  </exclusion>
  <exclusion>
 <groupId>com.sun.jmx</groupId>
 <artifactId>jmxri</artifactId>
  </exclusion>
 </exclusions>
 <scope>runtime</scope>
 <scope>runtime</scope>
</dependency>

<!-- @Inject -->
<dependency>
 <groupId>javax.inject</groupId>
 <artifactId>javax.inject</artifactId>
 <version>1</version>
</dependency>

<!-- Servlet -->
<dependency>
 <groupId>javax.servlet</groupId>
 <artifactId>servlet-api</artifactId>
 <version>2.5</version>
 <scope>provided</scope>
</dependency>

<dependency>
 <groupId>javax.servlet.jsp</groupId>
 <artifactId>jsp-api</artifactId>
 <version>2.1</version>
 <scope>provided</scope>
</dependency>

<dependency>
 <groupId>javax.servlet</groupId>
 <artifactId>jstl</artifactId>
 <version>1.2</version>
</dependency>

<!-- Test -->
  <dependency>
   <groupId>junit</groupId>
   <artifactId>junit</artifactId>
   <version>4.7</version>
   <scope>test</scope>
  </dependency>
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>5.1.40</version>
  </dependency>
 </dependencies>
 <build>
  <plugins>
 <plugin>
<artifactId>maven-eclipse-plugin</artifactId>
<version>2.9</version>
<configuration>
 <additionalProjectnatures>
<projectnature>org.springframework.ide.eclipse.core.springnature</projectnature>
 </additionalProjectnatures>
 <additionalBuildcommands>
<buildcommand>org.springframework.ide.eclipse.core.springbuilder</buildcommand>
 </additionalBuildcommands>
 <downloadSources>true</downloadSources>
 <downloadJavadocs>true</downloadJavadocs>
</configuration>
 </plugin>
```

自动加载Maven需要的Jar包



```
◢ LibSysREST
    src/main/resources
  ◢ src/main/java
      net.tydaniel.spring
    src/test/java
  ◢ Maven Dependencies
    ▷ jackson-databind-2.2.3.jar - C:\Users\xiaojunp\.r
    ▷ jackson-annotations-2.2.3.jar - C:\Users\xiaojunp
    ▷ jackson-core-2.2.3.jar - C:\Users\xiaojunp\.m2\r
    ▷ spring-context-4.0.3.RELEASE.jar - C:\Users\xiao
    ▷ spring-aop-4.0.3.RELEASE.jar - C:\Users\xiaojun
    ▷ aopalliance-1.0.jar - C:\Users\xiaojunp\.m2\repo
    ▷ spring-beans-4.0.3.RELEASE.jar - C:\Users\xiaoju
    ▷ spring-core-4.0.3.RELEASE.jar - C:\Users\xiaojun
    ▷ spring-expression-4.0.3.RELEASE.jar - C:\Users\
    ▷ spring-webmvc-4.0.3.RELEASE.jar - C:\Users\xiao
    ▷ spring-web-4.0.3.RELEASE.jar - C:\Users\xiaojun
    ▷ spring-tx-4.0.3.RELEASE.jar - C:\Users\xiaojunp\
    ▷ hibernate-core-4.3.5.Final.jar - C:\Users\xiaojunp
    ▷ jboss-logging-3.1.3.GA.jar - C:\Users\xiaojunp\.r
    ▷ jboss-logging-annotations-1.2.0.Beta1.jar - C:\U
    ▷ jboss-transaction-api_1.2_spec-1.0.0.Final.jar - C
    ▷ dom4j-1.6.1.jar - C:\Users\xiaojunp\.m2\reposit
    ▷ xml-apis-1.0.b2.jar - C:\Users\xiaojunp\.m2\repo
    ▷ hibernate-commons-annotations-4.0.4.Final.jar -
    ▷ hibernate-jpa-2.1-api-1.0.0.Final.jar - C:\Users\xi
    ▷ javassist-3.18.1-GA.jar - C:\Users\xiaojunp\.m2\
```

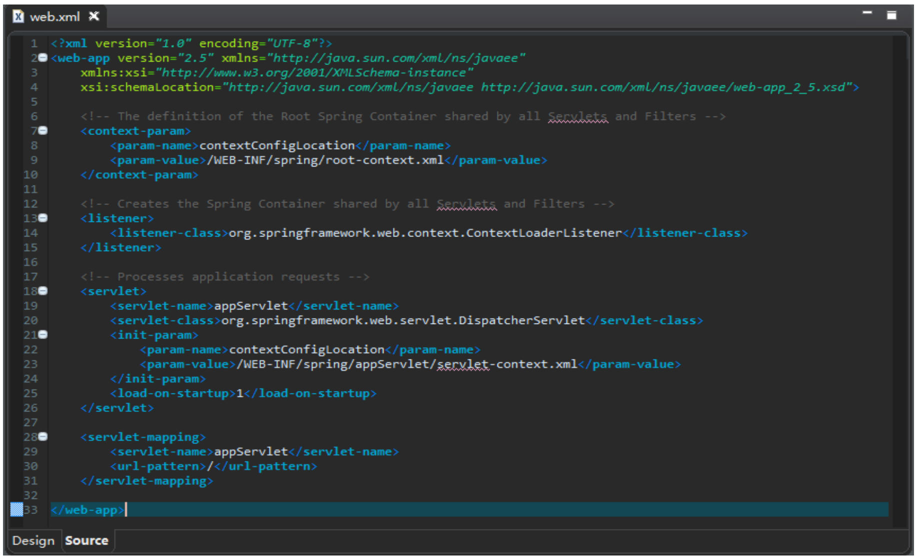
Hibernate



```
Package Explorer
LibSysREST
  src/main/resources
  src/main/java
    net.tydaniel.spring
  src/test/java
  Maven Dependencies
  JRE System Library [JavaSE-1.8]
  src
    main
      webapp
        WEB-INF
          spring
            appServlet
              servlet-context.xml
            root-context.xml
          web.xml
        index.jsp
    test
  target
    m2e-wtp
      web-resources
        META-INF
          maven
            net.tydaniel
            MANIFEST.MF
  pom.xml
```

创建Spring下的configure文件

web.xml

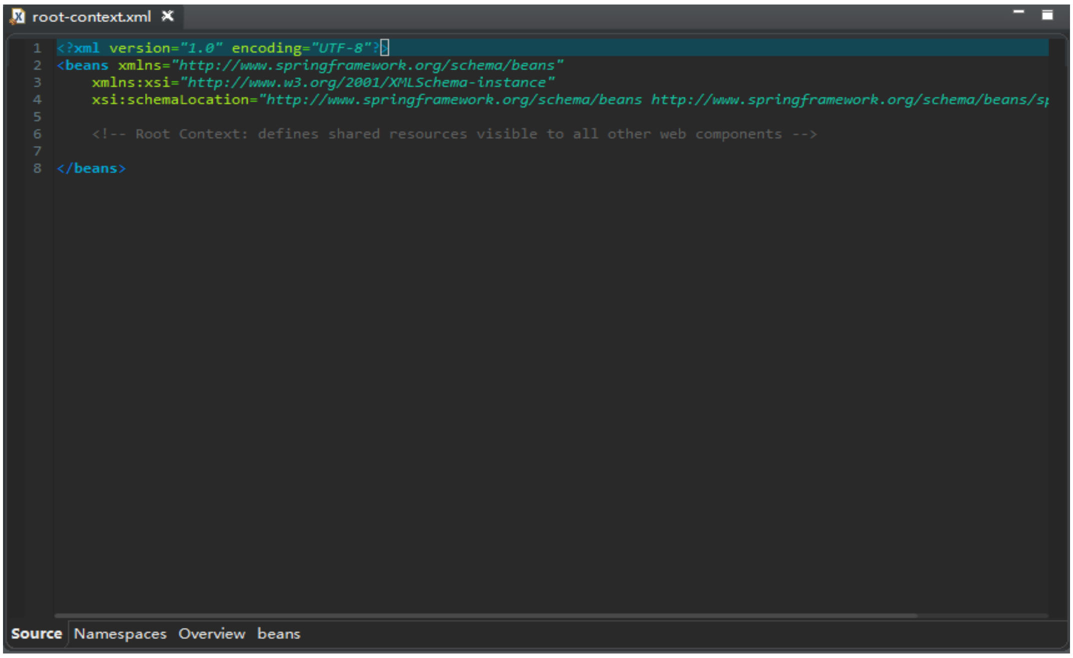


```xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">

    <!-- The definition of the Root Spring Container shared by all Servlets and Filters -->
    <context-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>/WEB-INF/spring/root-context.xml</param-value>
    </context-param>

    <!-- Creates the Spring Container shared by all Servlets and Filters -->
    <listener>
        <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
    </listener>

    <!-- Processes application requests -->
    <servlet>
        <servlet-name>appServlet</servlet-name>
        <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
        <init-param>
            <param-name>contextConfigLocation</param-name>
            <param-value>/WEB-INF/spring/appServlet/servlet-context.xml</param-value>
        </init-param>
        <load-on-startup>1</load-on-startup>
    </servlet>

    <servlet-mapping>
        <servlet-name>appServlet</servlet-name>
        <url-pattern>/</url-pattern>
    </servlet-mapping>

</web-app>
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">

<!-- The definition of the Root Spring Container shared by all Servlets and Filters -->
<context-param>
<param-name>contextConfigLocation</param-name>
<param-value>/WEB-INF/spring/root-context.xml</param-value>
</context-param>

<!-- Creates the Spring Container shared by all Servlets and Filters -->
<listener>
<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>

<!-- Processes application requests -->
<servlet>
<servlet-name>appServlet</servlet-name>
<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
<init-param>
<param-name>contextConfigLocation</param-name>
<param-value>/WEB-INF/spring/appServlet/servlet-context.xml</param-value>
</init-param>
<load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
<servlet-name>appServlet</servlet-name>
<url-pattern>/</url-pattern>
```

root-context.xml

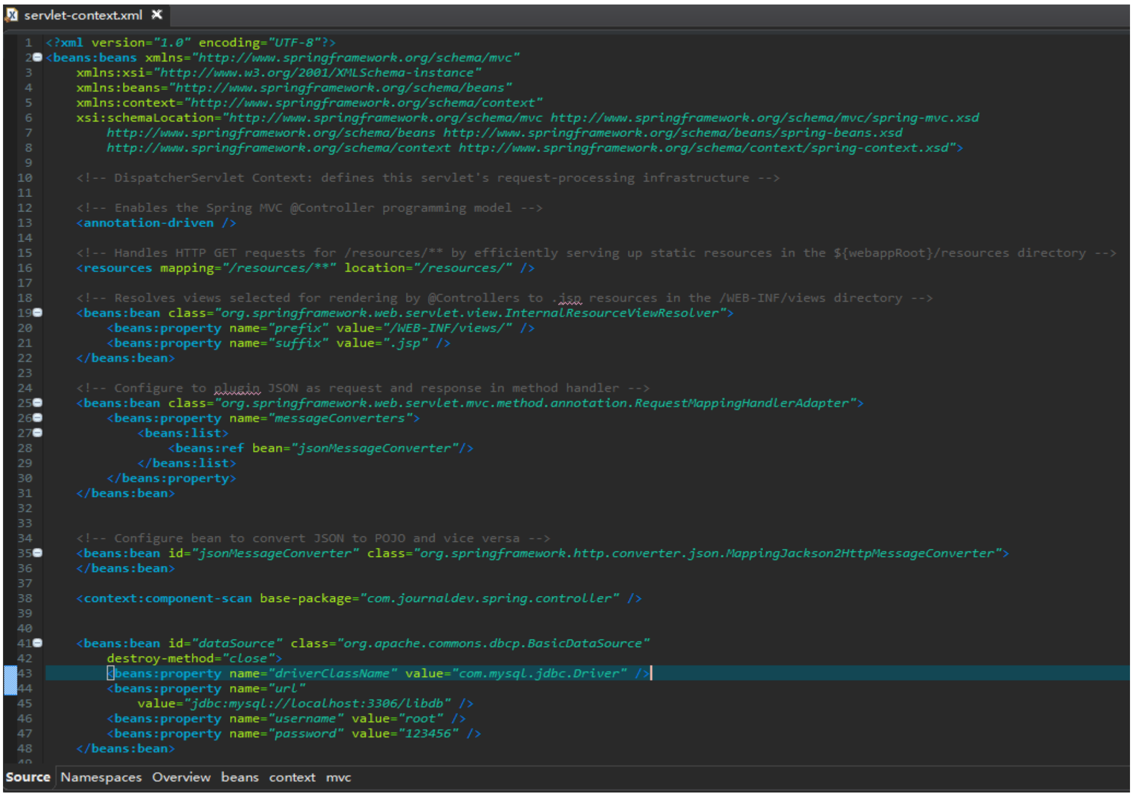


```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/s

    <!-- Root Context: defines shared resources visible to all other web components -->

</beans>
```

root-context.xml

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

<!-- Root Context: defines shared resources visible to all other web components -->
```

servlet-context.xml

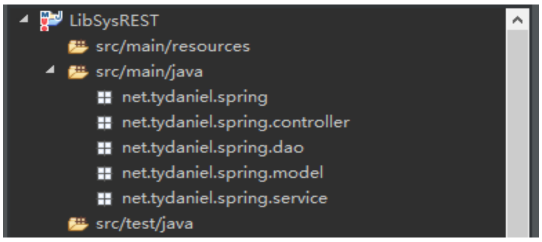


```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans:beans xmlns="http://www.springframework.org/schema/mvc"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:beans="http://www.springframework.org/schema/beans"
    xmlns:context="http://www.springframework.org/schema/context"
    xsi:schemaLocation="http://www.springframework.org/schema/mvc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

    <!-- DispatcherServlet Context: defines this servlet's request-processing infrastructure -->

    <!-- Enables the Spring MVC @Controller programming model -->
    <annotation-driven />

    <!-- Handles HTTP GET requests for /resources/** by efficiently serving up static resources in the ${webappRoot}/resources directory -->
    <resources mapping="/resources/**" location="/resources/" />

    <!-- Resolves views selected for rendering by @Controllers to .jsp resources in the /WEB-INF/views directory -->
    <beans:bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <beans:property name="prefix" value="/WEB-INF/views/" />
        <beans:property name="suffix" value=".jsp" />
    </beans:bean>

    <!-- Configure to plugin JSON as request and response in method handler -->
    <beans:bean class="org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter">
        <beans:property name="messageConverters">
            <beans:list>
                <beans:ref bean="jsonMessageConverter"/>
            </beans:list>
        </beans:property>
    </beans:bean>


    <!-- Configure bean to convert JSON to POJO and vice versa -->
    <beans:bean id="jsonMessageConverter" class="org.springframework.http.converter.json.MappingJackson2HttpMessageConverter">
    </beans:bean>

    <context:component-scan base-package="com.journaldev.spring.controller" />


    <beans:bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource"
        destroy-method="close">
        <beans:property name="driverClassName" value="com.mysql.jdbc.Driver" />
        <beans:property name="url"
            value="jdbc:mysql://localhost:3306/libdb" />
        <beans:property name="username" value="root" />
        <beans:property name="password" value="123456" />
    </beans:bean>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans:beans xmlns="http://www.springframework.org/schema/mvc"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:beans="http://www.springframework.org/schema/beans"
xmlns:context="http://www.springframework.org/schema/context"
xmlns:tx="http://www.springframework.org/schema/tx"
xsi:schemaLocation="http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc.xsd
http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd
http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-4.0.xsd">

<!-- DispatcherServlet Context: defines this servlet's request-processing infrastructure -->

<!-- Enables the Spring MVC @Controller programming model -->
<annotation-driven />

<!-- Handles HTTP GET requests for /resources/** by efficiently serving up static resources in the ${webappRoot}/resources directory -->
<resources mapping="/resources/**" location="/resources/" />

<!-- Resolves views selected for rendering by @Controllers to .jsp resources in the /WEB-INF/views directory -->
<beans:bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
<beans:property name="prefix" value="/WEB-INF/views/" />
<beans:property name="suffix" value=".jsp" />
</beans:bean>

<!-- Configure to plugin JSON as request and response in method handler -->
<beans:bean class="org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter">
<beans:property name="messageConverters">
<beans:list>
<beans:ref bean="jsonMessageConverter"/>
</beans:list>
</beans:property>
</beans:bean>
```

```
</beans:bean>


<!-- Configure bean to convert JSON to POJO and vice versa -->
<beans:bean id="jsonMessageConverter"
class="org.springframework.http.converter.json.MappingJackson2HttpMessageConverter">
</beans:bean>


<!-- Hibernate -->
<beans:bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource"
destroy-method="close">
<beans:property name="driverClassName" value="com.mysql.jdbc.Driver" />
<beans:property name="url"
value="jdbc:mysql://localhost:3306/libdb" />
<beans:property name="username" value="root" />
<beans:property name="password" value="123456" />
</beans:bean>
<!-- Hibernate 4 SessionFactory Bean definition -->
<beans:bean id="hibernate4AnnotatedSessionFactory"
class="org.springframework.orm.hibernate4.LocalSessionFactoryBean">
<beans:property name="dataSource" ref="dataSource" />
<beans:property name="annotatedClasses">
<beans:list>
<beans:value>net.tydaniel.spring.model.User</beans:value>
</beans:list>
</beans:property>
<beans:property name="hibernateProperties">
<beans:props>
<beans:prop key="hibernate.dialect">org.hibernate.dialect.MySQLDialect
</beans:prop>
<beans:prop key="hibernate.show_sql">true</beans:prop>
</beans:props>
</beans:property>
</beans:bean>
<beans:bean id="userDAO" class="net.tydaniel.spring.dao.UserDAOImpl">
<beans:property name="sessionFactory" ref="hibernate4AnnotatedSessionFactory" />
</beans:bean>
<beans:bean id="userService" class="net.tydaniel.spring.service.UserServiceImpl">
<beans:property name="userDAO" ref="userDAO"></beans:property>
</beans:bean>
```

创建Package

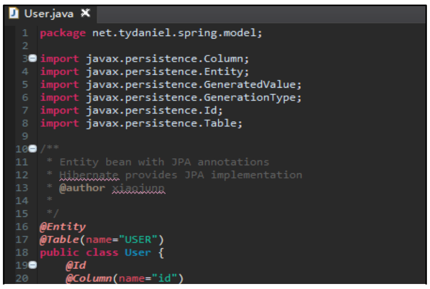


```
LibSysREST
    src/main/resources
    src/main/java
        net.tydaniel.spring
        net.tydaniel.spring.controller
        net.tydaniel.spring.dao
        net.tydaniel.spring.model
        net.tydaniel.spring.service
    src/test/java
```

Hibernate　JPA

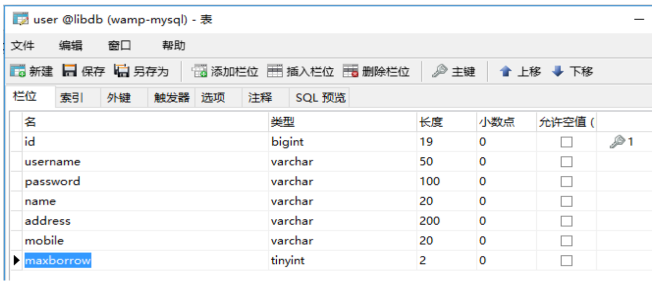
Ref URI: http://docs.oracle.com/javaee/7/api/javax/persistence/package-summary.html
Table　http://docs.oracle.com/javaee/7/api/javax/persistence/Table.html
Column　http://docs.oracle.com/javaee/7/api/javax/persistence/Column.html
GeneratedVa http://docs.oracle.com/javaee/7/api/javax/persistence/GeneratedValue.html

实体Bean　Model



User.java

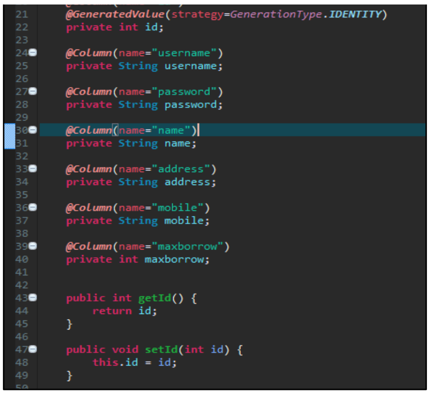
```java
package net.tydaniel.spring.model;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

/**
 * Entity bean with JPA annotations
 * Hibernate provides JPA implementation
 * @author xiaojunp
 *
 */
@Entity
@Table(name="USER")
public class User {
    @Id
    @Column(name="id")
```



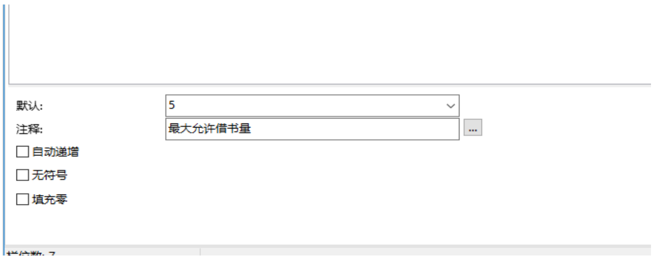
user @libdb (wamp-mysql) - 表

文件　编辑　窗口　帮助
新建　保存　另存为　添加栏位　插入栏位　删除栏位　主键　上移　下移
栏位　索引　外键　触发器　选项　注释　SQL 预览

名	类型	长度	小数点	允许空值 (	
id	bigint	19	0	□	🔑1
username	varchar	50	0	□	
password	varchar	100	0	□	
name	varchar	20	0	□	
address	varchar	200	0	□	
mobile	varchar	20	0	□	
maxborrow	tinyint	2	0	□	



```java
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private int id;

    @Column(name="username")
    private String username;

    @Column(name="password")
    private String password;

    @Column(name="name")
    private String name;

    @Column(name="address")
    private String address;

    @Column(name="mobile")
    private String mobile;

    @Column(name="maxborrow")
    private int maxborrow;


    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }
```



默认: 5

注释: 最大允许借书量

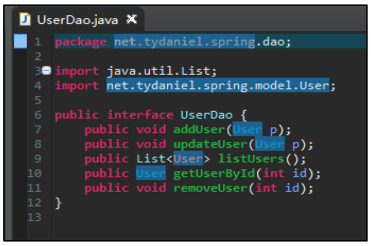
□ 自动递增

□ 无符号

□ 填充零

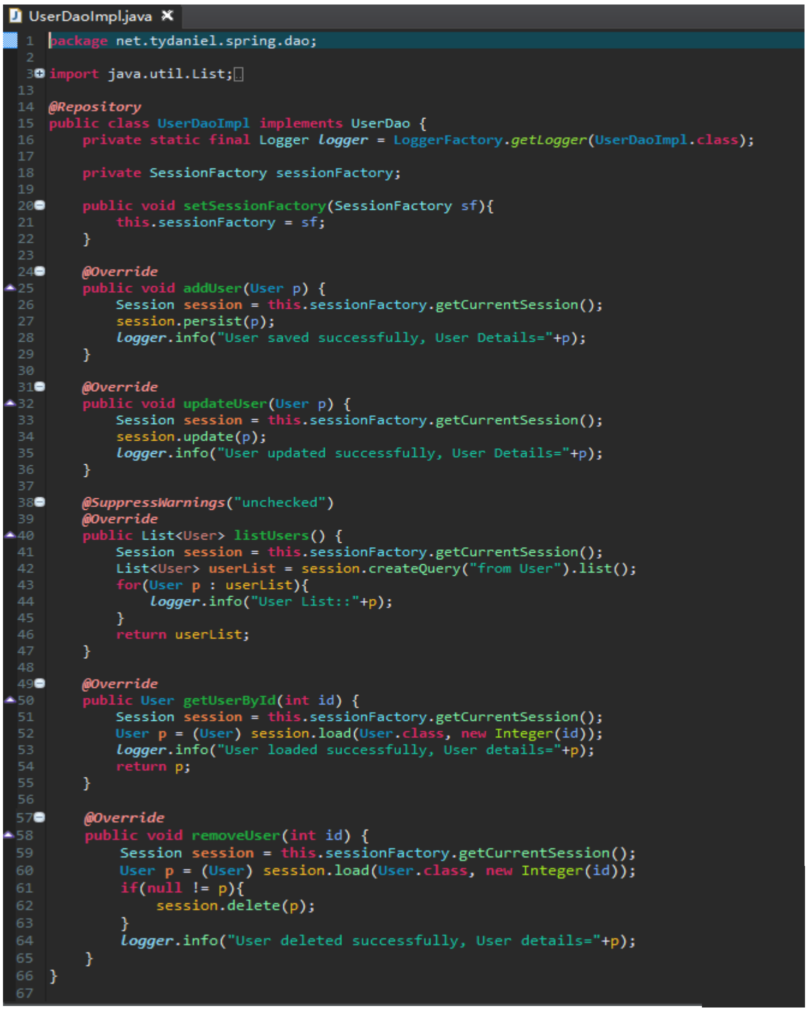
栏位数: 7

DAO



UserDao.java

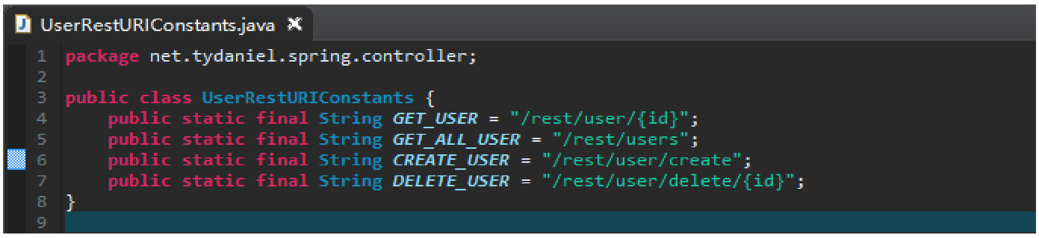
```java
package net.tydaniel.spring.dao;

import java.util.List;
import net.tydaniel.spring.model.User;

public interface UserDao {
    public void addUser(User p);
    public void updateUser(User p);
    public List<User> listUsers();
    public User getUserById(int id);
    public void removeUser(int id);
}
```



```java
package net.tydaniel.spring.dao;

import java.util.List;

@Repository
public class UserDaoImpl implements UserDao {
    private static final Logger logger = LoggerFactory.getLogger(UserDaoImpl.class);

    private SessionFactory sessionFactory;

    public void setSessionFactory(SessionFactory sf){
        this.sessionFactory = sf;
    }

    @Override
    public void addUser(User p) {
        Session session = this.sessionFactory.getCurrentSession();
        session.persist(p);
        logger.info("User saved successfully, User Details="+p);
    }

    @Override
    public void updateUser(User p) {
        Session session = this.sessionFactory.getCurrentSession();
        session.update(p);
        logger.info("User updated successfully, User Details="+p);
    }

    @SuppressWarnings("unchecked")
    @Override
    public List<User> listUsers() {
        Session session = this.sessionFactory.getCurrentSession();
        List<User> userList = session.createQuery("from User").list();
        for(User p : userList){
            logger.info("User List::"+p);
        }
        return userList;
    }

    @Override
    public User getUserById(int id) {
        Session session = this.sessionFactory.getCurrentSession();
        User p = (User) session.load(User.class, new Integer(id));
        logger.info("User loaded successfully, User details="+p);
        return p;
    }

    @Override
    public void removeUser(int id) {
        Session session = this.sessionFactory.getCurrentSession();
        User p = (User) session.load(User.class, new Integer(id));
        if(null != p){
            session.delete(p);
        }
        logger.info("User deleted successfully, User details="+p);
    }
}
```

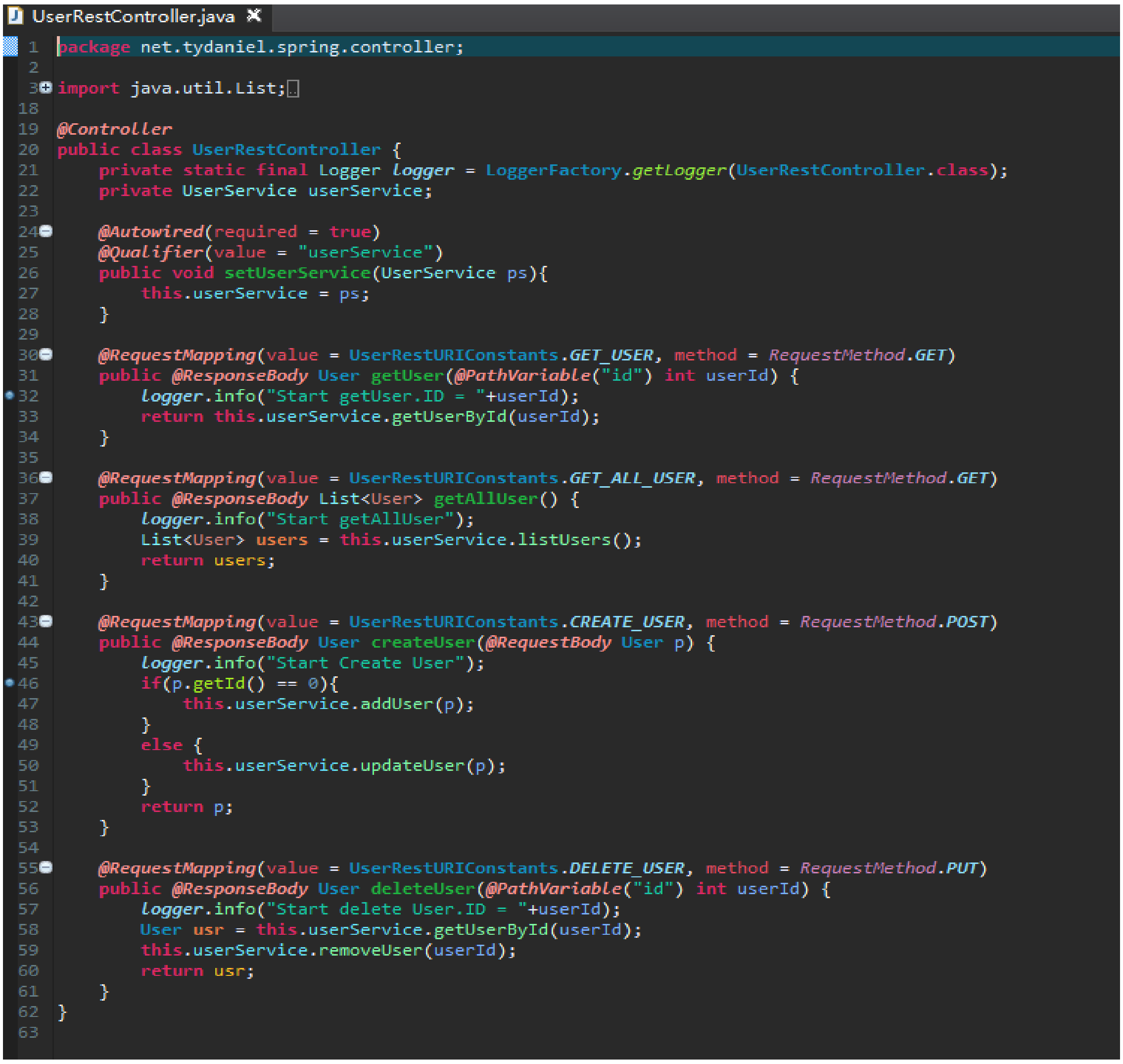
REST Controller

REST API 接口定义



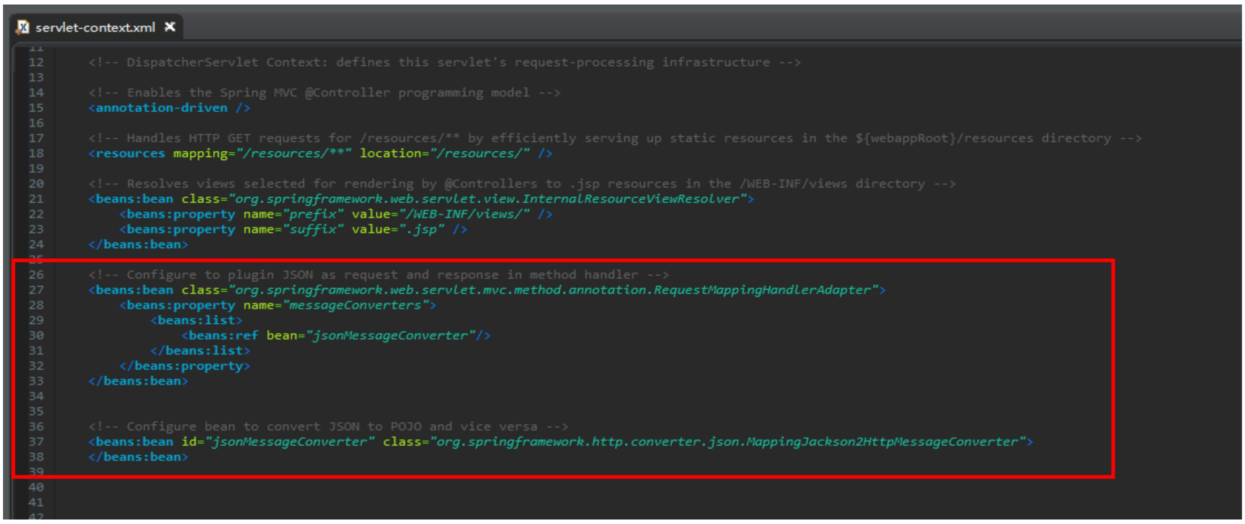
```java
package net.tydaniel.spring.controller;

public class UserRestURIConstants {
    public static final String GET_USER = "/rest/user/{id}";
    public static final String GET_ALL_USER = "/rest/users";
    public static final String CREATE_USER = "/rest/user/create";
    public static final String DELETE_USER = "/rest/user/delete/{id}";
}
```

UserRestController.java



```java
package net.tydaniel.spring.controller;

import java.util.List;

@Controller
public class UserRestController {
    private static final Logger logger = LoggerFactory.getLogger(UserRestController.class);
    private UserService userService;

    @Autowired(required = true)
    @Qualifier(value = "userService")
    public void setUserService(UserService ps){
        this.userService = ps;
    }

    @RequestMapping(value = UserRestURIConstants.GET_USER, method = RequestMethod.GET)
    public @ResponseBody User getUser(@PathVariable("id") int userId) {
        logger.info("Start getUser.ID = "+userId);
        return this.userService.getUserById(userId);
    }

    @RequestMapping(value = UserRestURIConstants.GET_ALL_USER, method = RequestMethod.GET)
    public @ResponseBody List<User> getAllUser() {
        logger.info("Start getAllUser");
        List<User> users = this.userService.listUsers();
        return users;
    }

    @RequestMapping(value = UserRestURIConstants.CREATE_USER, method = RequestMethod.POST)
    public @ResponseBody User createUser(@RequestBody User p) {
        logger.info("Start Create User");
        if(p.getId() == 0){
            this.userService.addUser(p);
        }
        else {
            this.userService.updateUser(p);
        }
        return p;
    }

    @RequestMapping(value = UserRestURIConstants.DELETE_USER, method = RequestMethod.PUT)
    public @ResponseBody User deleteUser(@PathVariable("id") int userId) {
        logger.info("Start delete User.ID = "+userId);
        User usr = this.userService.getUserById(userId);
        this.userService.removeUser(userId);
        return usr;
    }
}
```

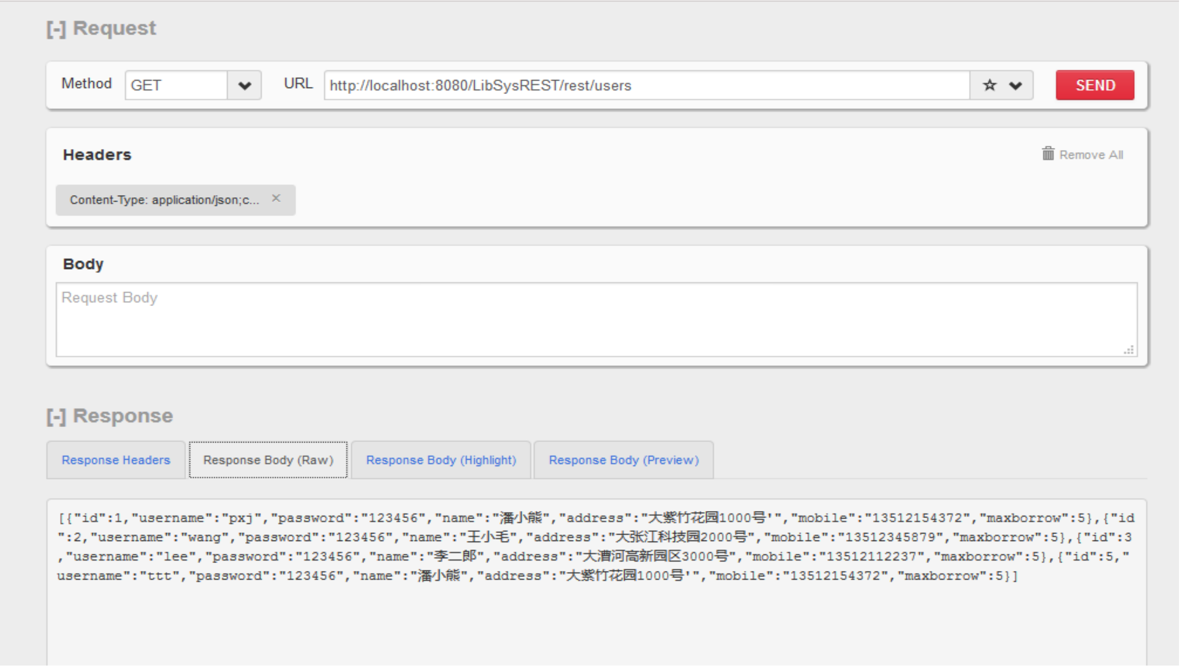
配置 JSON格式的转换



```xml
<!-- DispatcherServlet Context: defines this servlet's request-processing infrastructure -->

<!-- Enables the Spring MVC @Controller programming model -->
<annotation-driven />

<!-- Handles HTTP GET requests for /resources/** by efficiently serving up static resources in the ${webappRoot}/resources directory -->
<resources mapping="/resources/**" location="/resources/" />

<!-- Resolves views selected for rendering by @Controllers to .jsp resources in the /WEB-INF/views directory -->
<beans:bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <beans:property name="prefix" value="/WEB-INF/views/" />
    <beans:property name="suffix" value=".jsp" />
</beans:bean>

<!-- Configure to plugin JSON as request and response in method handler -->
<beans:bean class="org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter">
    <beans:property name="messageConverters">
        <beans:list>
            <beans:ref bean="jsonMessageConverter"/>
        </beans:list>
    </beans:property>
</beans:bean>


<!-- Configure bean to convert JSON to POJO and vice versa -->
<beans:bean id="jsonMessageConverter" class="org.springframework.http.converter.json.MappingJackson2HttpMessageConverter">
</beans:bean>
```

Running Image

GET http://localhost:8080/LibSysREST/rest/users



Method GET

URL http://localhost:8080/LibSysREST/rest/users

Headers

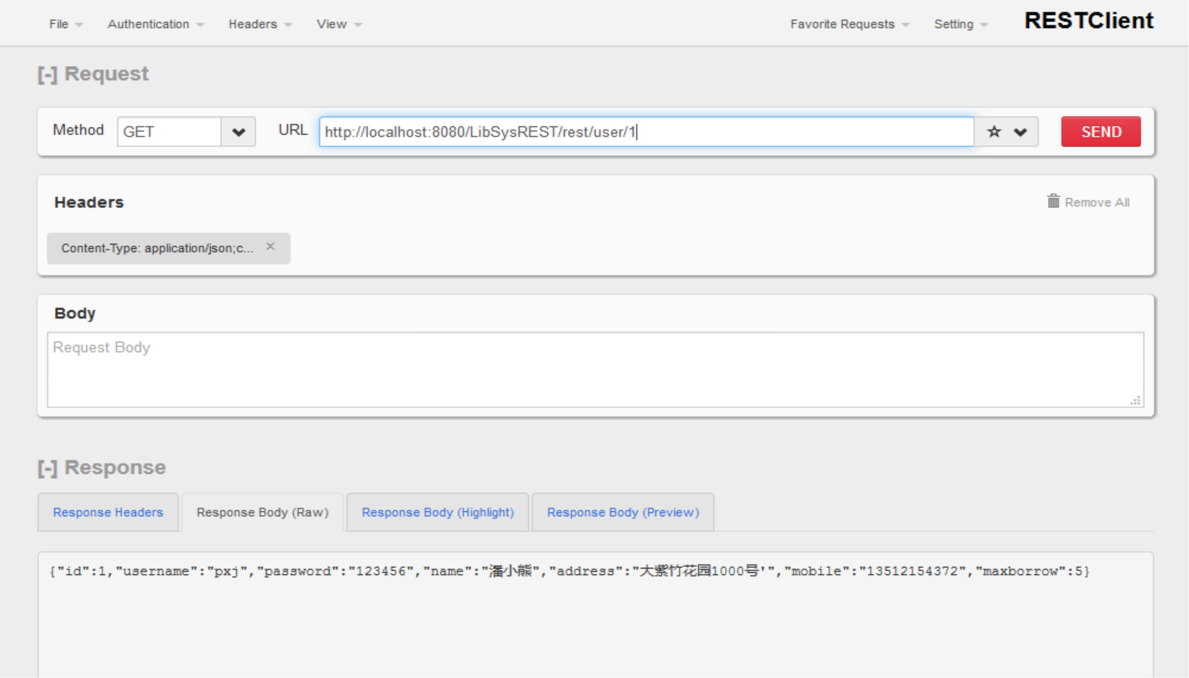
Content-Type: application/json;c...

Body

Request Body

```
[{"id":1,"username":"pxj","password":"123456","name":"潘小熊","address":"大紫竹花园1000号'","mobile":"13512154372","maxborrow":5},{"id
":2,"username":"wang","password":"123456","name":"王小毛","address":"大张江科技园2000号","mobile":"13512345879","maxborrow":5},{"id":3
,"username":"lee","password":"123456","name":"李二郎","address":"大漕河高新园区3000号","mobile":"13512112237","maxborrow":5},{"id":5,"
username":"ttt","password":"123456","name":"潘小熊","address":"大紫竹花园1000号'","mobile":"13512154372","maxborrow":5}]
```

GET http://localhost:8080/LibSysREST/rest/user/1



Method GET

URL http://localhost:8080/LibSysREST/rest/user/1

Headers

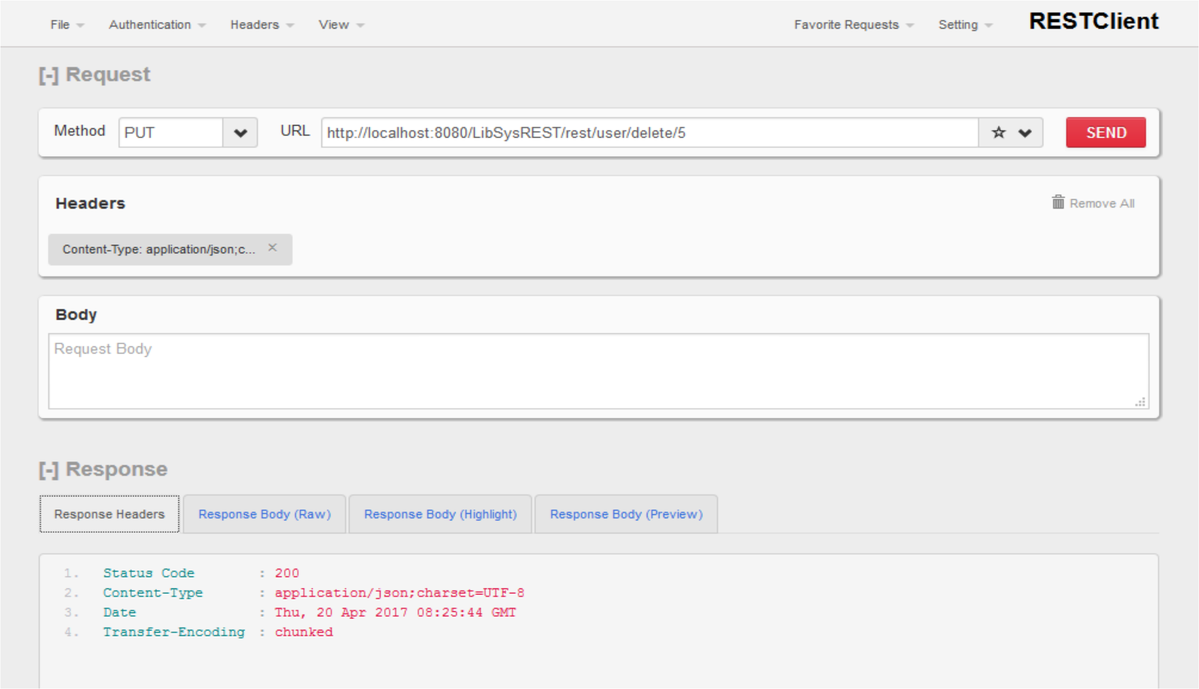
Content-Type: application/json;c...

Body

Request Body

```
{"id":1,"username":"pxj","password":"123456","name":"潘小熊","address":"大紫竹花园1000号'","mobile":"13512154372","maxborrow":5}
```

PUT： http://localhost:8080/LibSysREST/rest/user/delete/5



Method PUT URL http://localhost:8080/LibSysREST/rest/user/delete/5

Headers

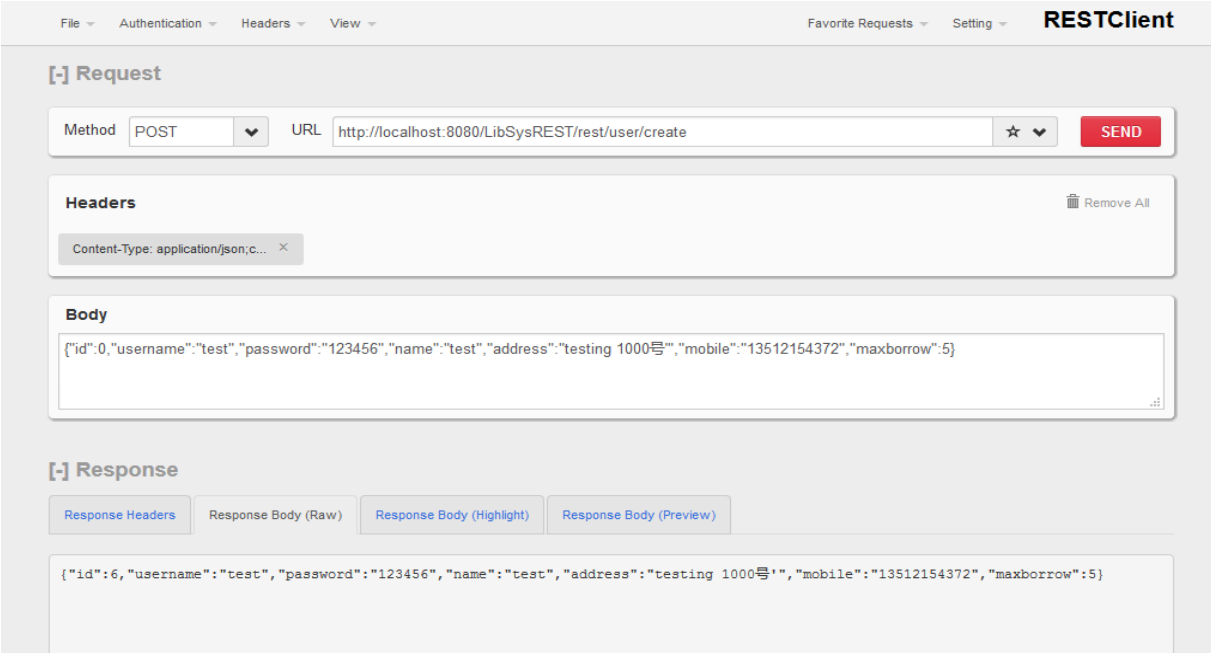
Content-Type: application/json;c...

Body

Request Body

```
Status Code       : 200
Content-Type      : application/json;charset=UTF-8
Date              : Thu, 20 Apr 2017 08:25:44 GMT
Transfer-Encoding : chunked
```

POST： http://localhost:8080/LibSysREST/rest/user/create
Header： Content-Type: application/json;charset=UTF-8
Body： {"id":0,"username":"test","password":"123456","name":"test","address":"testing 1000号'","mobile":"13512154372","maxb



Method POST URL http://localhost:8080/LibSysREST/rest/user/create

Headers

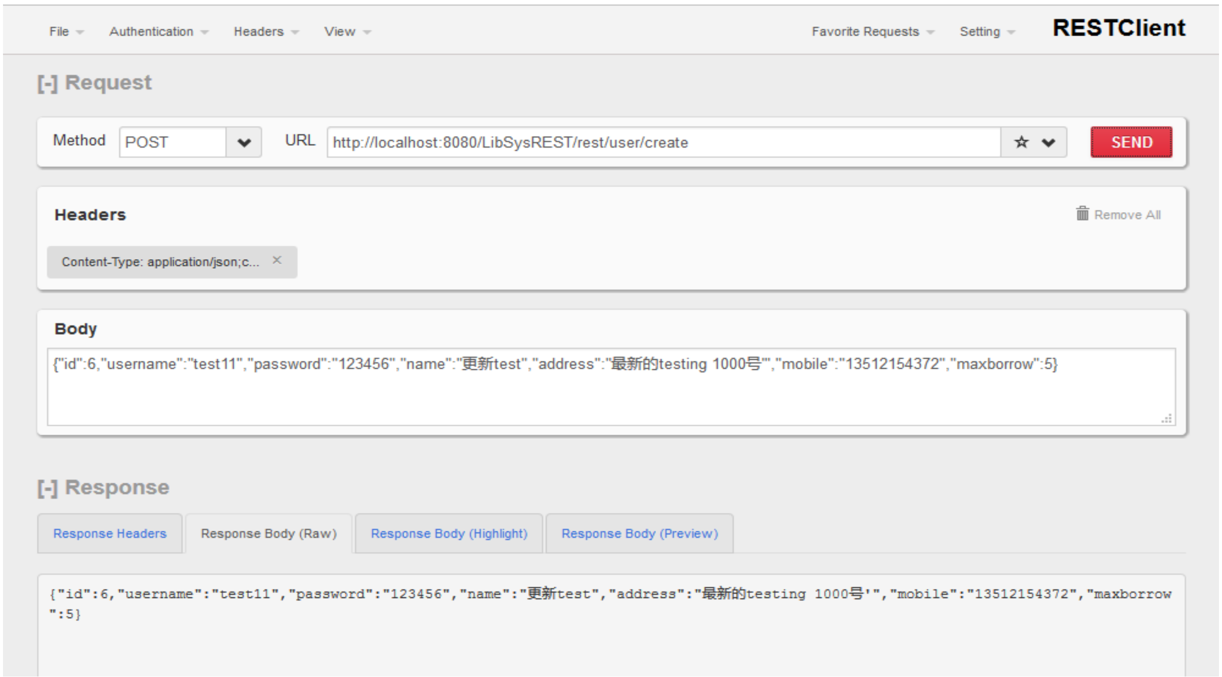
Content-Type: application/json;c...

Body

{"id":0,"username":"test","password":"123456","name":"test","address":"testing 1000号'","mobile":"13512154372","maxborrow":5}

```
{"id":6,"username":"test","password":"123456","name":"test","address":"testing 1000号'","mobile":"13512154372","maxborrow":5}
```

POST： http://localhost:8080/LibSysREST/rest/user/create
Header： Content-Type: application/json;charset=UTF-8
Body：
{"id":6,"username":"test11","password":"123456","name":"更新test","address":"最新的testing 1000号'","mobile":"13512154372","



[-] Request

Method: POST

URL: http://localhost:8080/LibSysREST/rest/user/create

Headers

Content-Type: application/json;c... ×

Body

{"id":6,"username":"test11","password":"123456","name":"更新test","address":"最新的testing 1000号'","mobile":"13512154372","maxborrow":5}

[-] Response

```
{"id":6,"username":"test11","password":"123456","name":"更新test","address":"最新的testing 1000号'","mobile":"13512154372","maxborrow
":5}
```