

长江流域重点断面水质监测系统

实习小结



课程名称：WebGIS 与地理信息服务课程设计

专业名称：地理信息科学

学生姓名：林玥

学 号：2015301110189

2018 年 7 月 4 日

目录

1 任务概述.....	2
2 需求分析.....	2
2.1 普通用户	2
2.2 管理员	2
3 架构设计.....	3
4 系统环境搭建	4
4.1 JAVA 配置.....	4
4.2 Tomcat 安装.....	5
4.3 GeoServer 安装.....	6
4.4 uDig 安装.....	7
4.5 PostgreSQL+PostGIS 安装.....	7
4.6 Eclipse 安装与 Web 开发环境搭建.....	8
4.7 云空间部署	9
5 数据组织与管理.....	9
5.1 空间数据组织与管理.....	9
5.2 用户信息登录管理	15
6 功能模块实现.....	16
6.1 登录校验.....	16
6.2 图层展示.....	16
6.3 数据库编辑	18
6.4 站点查询.....	18
6.5 水质参数统计图表展示.....	19
6.6 水质分类热力图展示.....	20
7 成果展示.....	21
7.1 登录校验.....	21
7.2 图层展示.....	21
7.3 数据库编辑	22
7.4 站点查询与水质参数统计图表展示.....	23
7.5 水质分类热力图展示.....	24
8 问题及解决办法.....	24
9 创新与贡献.....	24
10 优化与改进.....	25

1 任务概述

本次课程设计利用长江流域重点断面水质监测公开数据，设计并实现了基于 WebGIS 的长江流域重点断面水质监测系统，不仅便于管理人员对数据进行高效、便捷地更新，还可使普通市民更方便、直观地了解长江流域的水质分布状况，有助于推动政务公开工作，使政府决策议事过程更加透明、严谨。

2 需求分析

系统主要功能包括登录校验、图层展示、数据库编辑、站点查询、水质参数统计图表展示、水质分类热力图展示等（图 1）。其中，因登录角色不同，用户在登录后可使用的页面功能也不同（图 2）。

2.1 普通用户

编号	功能	描述
1	登录校验	用户在登录界面选择 普通用户 角色，并输入用户名和密码，利用 Cookie/Session 机制进行登录校验，校验成功后跳转至操作界面。
2	图层展示	调用 OpenLayers3 类库包，实现 OSM、WMS、GeoJSON 图层叠加显示；其中，OSM 图层为基础底图，WMS 图层为长江流域河流图层，GeoJSON 图层为水质监测站点图层。
3	站点查询	包括点击查询和名称查询，即通过点击站点或在搜索框中输入站点名称，可在侧边栏中查看站点信息。
4	水质参数统计图表展示	在侧边栏中展示根据对应站点当月的水质参数值（PH、DO、CODMn、NH ₃ -N）生成的折线图。
5	水质分类热力图展示	通过在下拉框进行选择（一类水质/二类水质/三类水质），可查看不同类型水质分布热力图。

2.2 管理员

编号	功能	描述
1	登录校验	用户在登录界面选择 管理员 角色，并输入用户名和密码，利用 Cookie/Session 机制进行登录校验，校验成功后跳转至操作界面。
2	图层展示	调用 OpenLayers3 类库包，实现 OSM、WMS、GeoJSON 图层叠加显示；其中，OSM 图层为基础底图，WMS 图层为长江流域河流图层，GeoJSON 图层为水质监测站点图层。
3	数据库编辑	站点的添加、删除和编辑。添加站点即通过输入站点的位置信息（经纬度坐标）和属性信息，在数据库中新增记录；删除站点即通过输入待删站点名称，将数据库中对应站点记录删除；编辑站点即通过输入待修改站点名称及待编辑字段信息，对数据库记录进行更新。

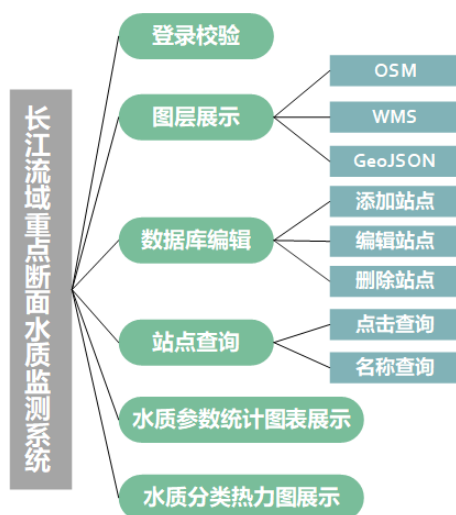


图 1 系统功能结构

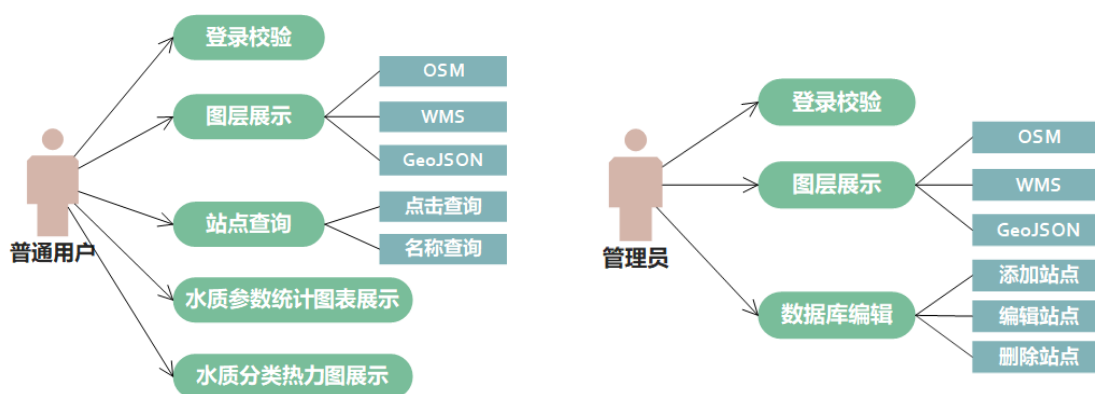


图 2 系统用例图

3 架构设计

本系统利用 B/S 架构，即浏览器/服务器架构；系统主要构成有前端显示层、业务逻辑层和数据访问层（图 3）。

(1) 前端显示层：

利用 HTML+CSS 进行页面展示，jQuery 框架与原生 JavaScript 相结合的方式实现前端校验；引入 OpenLayers 及 Echarts 库用于底图显示与统计图表显示；利用 GeoServer 2.4 作为地图服务器，用于发布底图静态数据。

Web 显示层主要包括登录页面（Login.jsp）、管理员操作页面（Manager.jsp）、普通用户操作页面（User.jsp）。

(2) 业务逻辑层：

利用 JSP 技术进行前后端数据的交互，Java Servlet 作为来自 Web 浏览器与 HTTP 服务器上的数据库或应用程序之间的中间层。

业务逻辑类包括 Session 验证（SessionCheck.jsp）、登录表单处理（LoginAction.jsp）、站点类（Station.java）等。

(3) 数据访问层:

利用 Java 语言作为服务器端开发语言, Apache Tomcat 9.0 作为应用服务器, PostgreSQL + PostGIS 空间数据库作为专题数据管理数据库, 对矢量数据存储。

数据库操作类包括数据库连接 (ConnUtil.java)、站点增删查改 (StationDao.java)、图片读取与写入 (ImageUtil.java)。

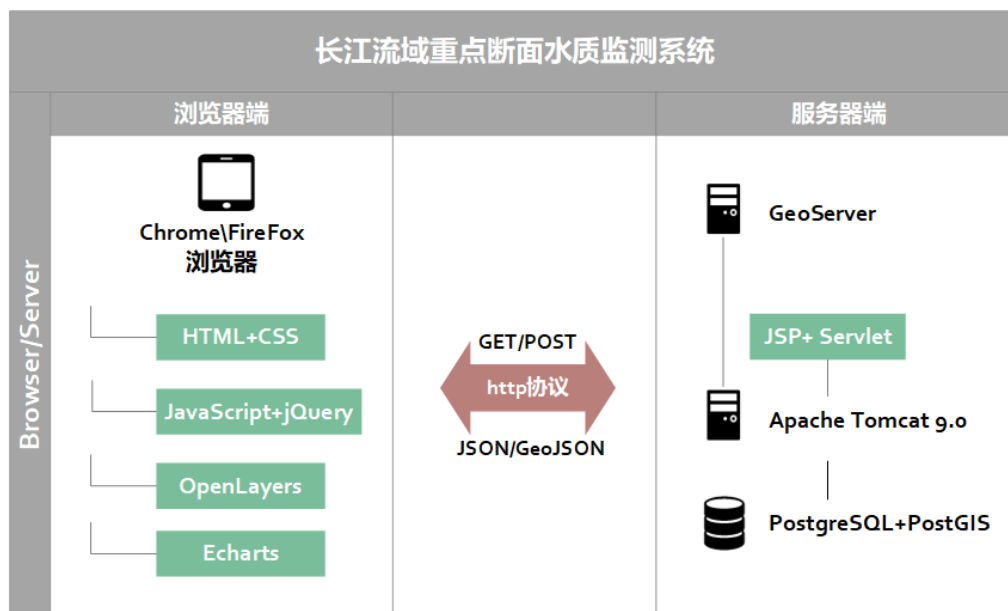


图 3 系统总体架构

4 系统环境搭建

4.1 JAVA 配置

(1) 安装 JDK:

在官网 (<http://www.oracle.com/technetwork/java/javase/downloads/index.html>) 下载相应版本的 JDK 并安装。本系统使用的 JDK 版本为 **jdk-8u171-windows-i586**。

(2) 配置环境变量:

单击“计算机-属性-高级系统设置-环境变量”，在“系统变量”栏下对系统环境变量进行添加和编辑：①新建->变量名“**JAVA_HOME**”，变量值“**C:\Program Files (x86)\Java\jdk1.8.0_171**”（即 JDK 的安装路径）；②编辑->变量名“**Path**”，在原变量值的最后面加上“**;%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin**”；③新建->变量名“**CLASSPATH**”，变量值“**.;%JAVA_HOME%\lib;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar**”。

(3) 确认环境变量配置是否正确:

在控制台分别输入 **java**, **javac**, **java -version** 命令, 出现如下所示的 JDK 的编译器信息, 包括修改命令的语法和参数选项等信息, 则配置成功。

```
C:\Users\linyue>java -version
java version "1.8.0_171"
Java(TM) SE Runtime Environment (build 1.8.0_171-b11)
Java HotSpot(TM) Client VM (build 25.171-b11, mixed mode, sharing)
```

图 4 JAVA 配置成功

4.2 Tomcat 安装

(1) 下载 Tomcat:

在官网 (<http://tomcat.apache.org/>) 下载与 Java 版本一致的 Tomcat 的压缩包, 放在一个没有中文字符的路径下, 并解压。本系统使用的是 **apache-tomcat-9.0.8-windows-x86**。

(2) 配置环境变量:

单击“计算机-属性-高级系统设置-环境变量”, 在“系统变量”栏下对系统环境变量进行添加和编辑: ①新建->变量名“CATALINA_HOME”, 变量值“C:\apache-tomcat-9.0.8”(即 Tomcat 解压后的目录); ②编辑->变量名“Path”, 在原变量值的最后面加上“;%CATALINA_HOME%\bin”。

(3) 修改默认端口:

用记事本打开 conf 目录下的 **server.xml** 文件, 找到如下位置, 将端口值从 8080 改为 80。这一步不是必要的, 但是建议做, 否则可能会与 GeoServer 自带的 Tomcat 服务器端口相冲突。

```
APR (HTTP/AJP) Connector: /docs/apr.html
Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
-->
<Connector connectionTimeout="20000" port="80" protocol="HTTP/1.1" redirectPort="8443"/>
<!-- A "Connector" using the shared thread pool-->
<!--
<Connector executor="tomcatThreadPool"
```

图 5 修改 Tomcat 默认端口

(4) 设置用户名和密码:

用记事本打开 conf 目录下的 **tomcat-users.xml** 文件, 添加如下代码。

```
<user password="[你的密码]" roles="manager-gui" username="admin"/>
```

(5) 启动 Tomcat:

双击运行 bin 目录下的 **startup.bat** 文件; 若出现如下信息, 则运行成功。

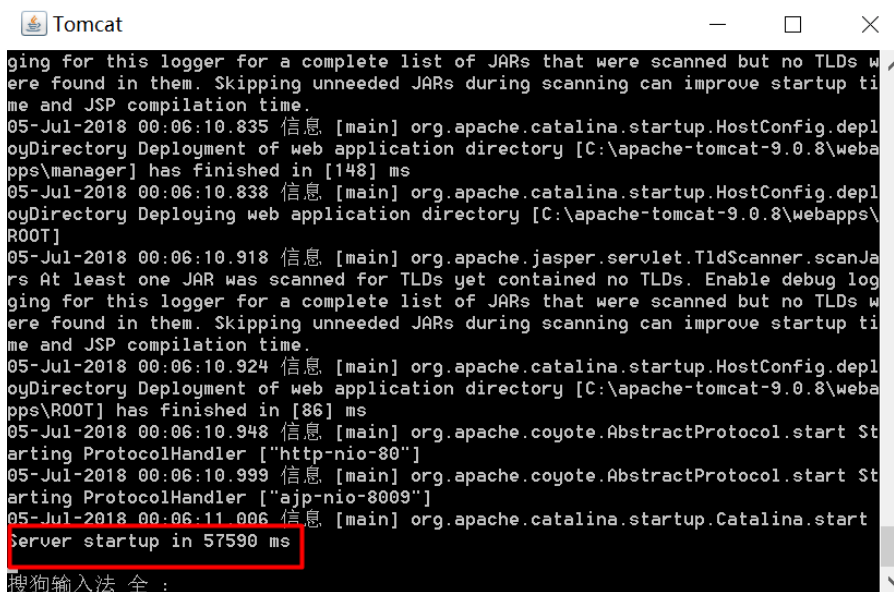


图 6 Tomcat 启动

(6) 打开 localhost 界面:

在浏览器中输入 `http://localhost:80` 打开 Tomcat 主页，出现欢迎界面，说明安装成功。

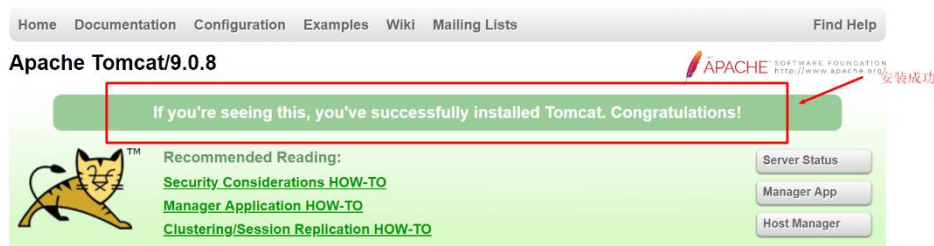


图 7 Tomcat 安装成功

4.3 GeoServer 安装

(1) 下载 GeoServer:

在官网(<http://geoserver.org/>)下载 GeoServer Web Archive 版本,解压提取 `geoserver.war`。本系统使用的是 **GeoServer2.4-RC2**。

(2) 部署 GeoServer:

在浏览器中输入 `http://localhost:80` 打开 Tomcat 主页，点击“Manager APP”，输入上面设置的 Tomcat 用户名和密码；验证通过后，点击下方的“选择文件”，将 `geoserver.war` 上传，点击 Deploy 进行发布。

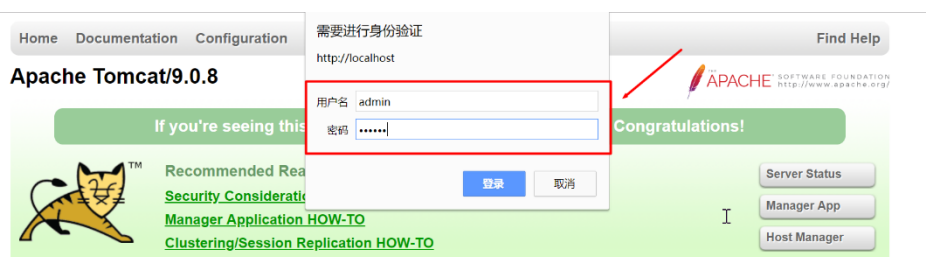


图 8 Tomcat 登录

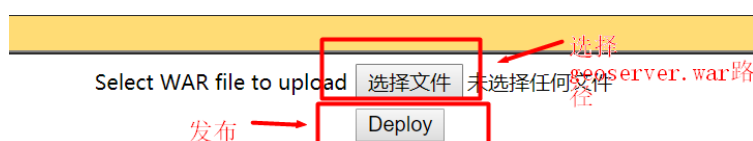


图 9 GeoServer 发布

/geoserver	None specified	GeoServer	true	0	Start Stop Reload Undeploy
					Expire sessions with idle ≥ 30 minutes

图 10 GeoServer 部署成功

(3) 验证安装是否成功:

在浏览器中输入 `http://localhost:80/geoserver` 出现欢迎界面，则安装成功。

(4) 登录 GeoServer:

GeoServer 的默认管理员用户名和密码为: `admin/geoserver`，在右上方输入后即可登录。

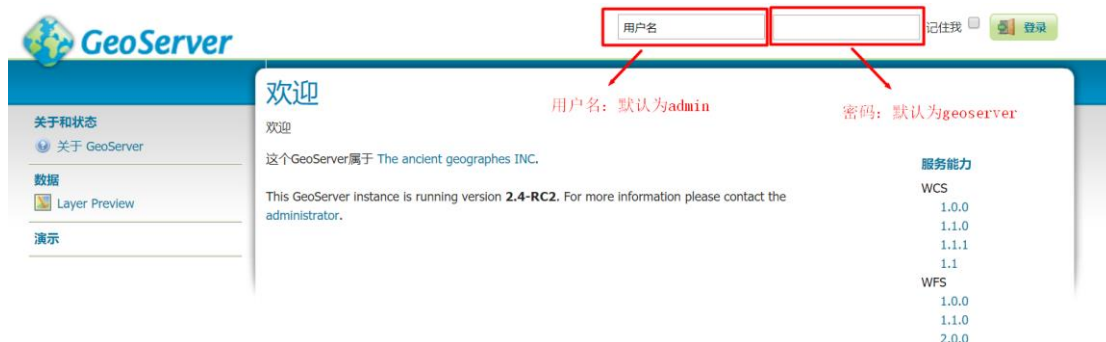


图 11 GeoServer 配置成功及登录

4.4 uDig 安装

在官网(<http://udig.refractory.net/>)下载最新版本的 uDig。这里采用的是 **udig-2.0.0.win32.x86**。

4.5 PostgreSQL+PostGIS 安装

(1) 下载 PostgreSQL 和 PostGIS:

在 PostgreSQL 官网(<https://www.postgresql.org/>)和 PostGIS 官网(<http://www.postgis.org/>)下载相应版本的软件，并注意两者对应版本间的匹配。本系统使用的是 **postgresql-9.6.9-1-windows-x64** 和 **postgis_2_4_pg96**。

(2) 安装 PostgreSQL 和 PostGIS:

在安装 PostgreSQL 过程中，会设置其用户名和密码用于后续数据库的登录；在安装 PostGIS 时，应注意勾选“Create spatial database”，默认同时创建空间数据库。

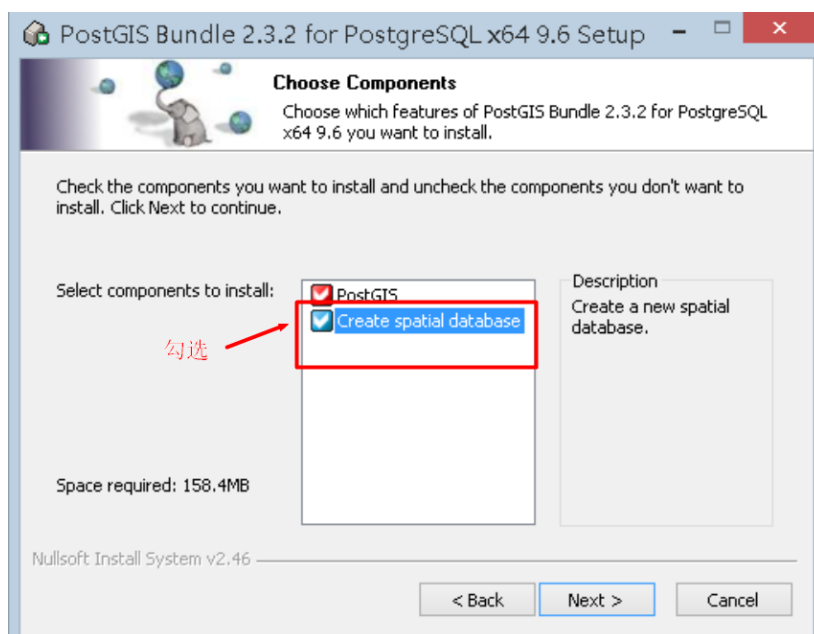


图 12 默认创建空间数据库

4.6 Eclipse 安装与 Web 开发环境搭建

(1) 下载 Eclipse:

在官网(<https://www.eclipse.org/downloads/eclipse-packages/>)下载相应版本的 Eclipse IDE for Java EE Developers 并安装。本系统使用的是 **eclipse-inst-win32**。

(2) 服务器环境设置:

在 Eclipse 菜单栏依次点击 Window - Preferences - Server - Runtime Environment - Add, 然后选择对应的 Tomcat 版本和之前的 Tomcat 安装路径, 以及对应的 JRE 版本 (图 13)。

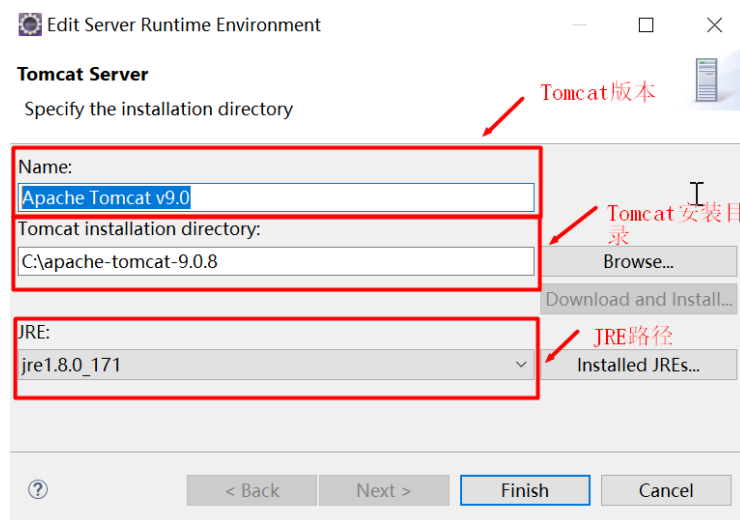


图 13 Eclipse 服务器环境配置

(3) 创建 Web 工程:

在 Eclipse 菜单栏依次点击 File - New - Dynamic Web Project, 将 classes 默认路径从 build/classes 修改至 WebContent\WEB-INF\classes, 否则生成的类将无法被前端 JSP 找到; 勾选“Generate web.xml development descriptor”, 否则新建的 web 项目将无法自动生成 web.xml 配置文件。

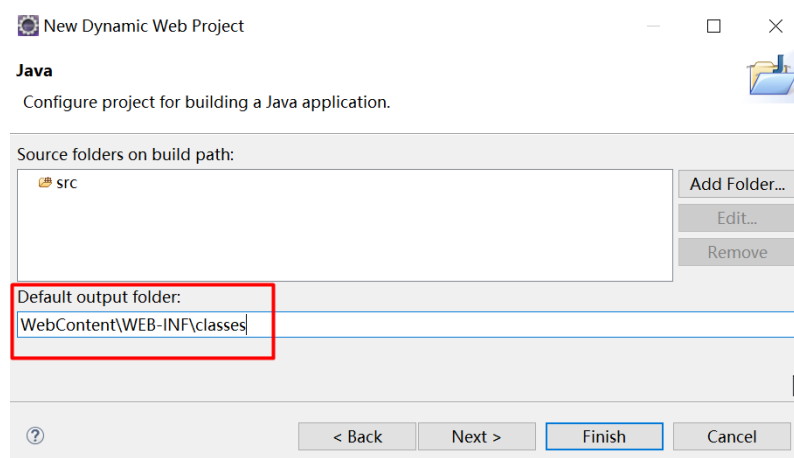


图 14 修改 classes 默认路径

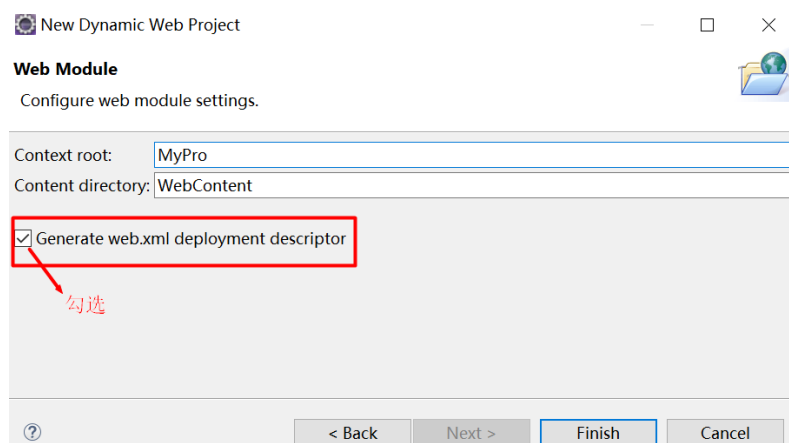


图 15 生成 web.xml 配置文件

4.7 云空间部署

(1) 购买服务器：

在阿里云官网云翼计划(<https://promotion.aliyun.com/ntms/campus2017.html>)经学生认证后购买服务器，这里购买的是轻量应用服务器，系统为 Windows 2012 R2。域名非必须，没有域名可以直接用公网 IP 访问。

(2) 配置服务器：

按照上述流程在云服务器上配置相应版本的 JAVA 和 Tomcat。后期可使用 xshell 和 xftp 工具连接到云服务器，将 war 包放到云服务器的.../apache-tomcat/webapps/下并解压即可。

5 数据组织与管理

5.1 空间数据组织与管理

5.1.1 数据准备

本系统采用的基础数据主要包括长江流域重点断面主要监测站点数据，以及长江流域主要河流数据。

(1) 专题数据——长江流域重点断面水质状况数据：

从中华人民共和国生态环境部数据中心网站¹下载全国主要流域重点断面水质状况表，并提取出其中长江流域的部分站点信息；在百度地图坐标拾取系统²中查询各个点位的经纬度坐标并添加到表格中；将表格转为.csv 格式，在 ArcGIS 中设置 x\y 坐标对应的字段，并导出为 shp 格式。

¹ 中华人民共和国生态环境部数据中心网站：

<http://datacenter.mep.gov.cn/websjzx/dataproduct/resourceproduct/queryDataToReport.vm?id=1&url=/websjzx/report!list.vm?xmlname=1512553411227&ftype=zxml>

² 百度地图坐标拾取系统：<http://api.map.baidu.com/lbsapi/getpoint/index.html>

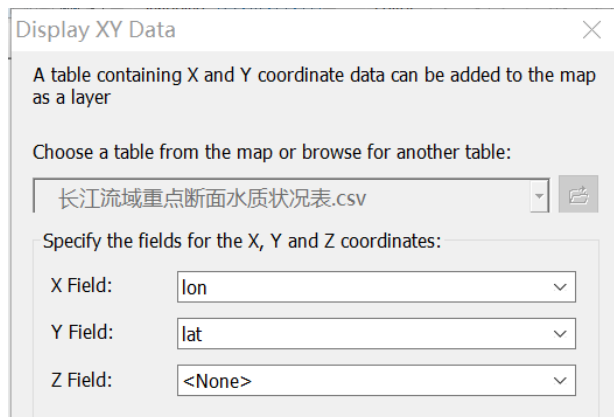


图 16 设置 x/y 坐标对应字段

(2) 基础底图数据——长江流域主要河流数据：

在 OpenStreetMap 官网提供的导出链接中（<https://download.geofabrik.de/asia.html>）下载中国基础地图要素的 shp 文件压缩包，并导出文件夹中 gis.osm_waterways_free_1.shp 数据的长江流域部分线状要素。

5.1.2 shp 文件导入 PostgreSQL

(1) 连接空间数据库：

打开 PostGIS Shapefile and DBF Loader Exporter，单击“View connection details...”，输入安装 PostGIS 过程中设置的用户名、密码、端口号和空间数据库名称。

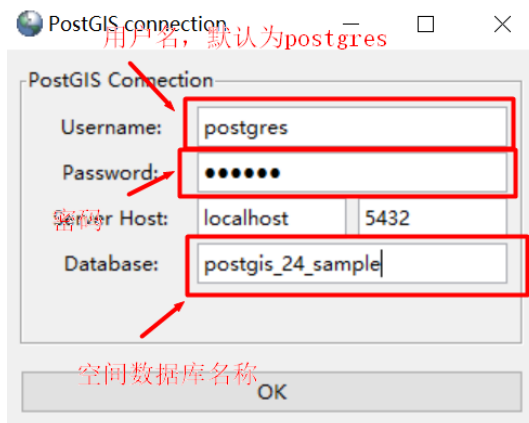


图 17 连接空间数据库

(2) 设置字符编码：

单击“Options...”，将“DBF file character encoding”从 UTF-8 改为 GBK；否则当 shp 文件中含有中文字段时将无法成功导入。

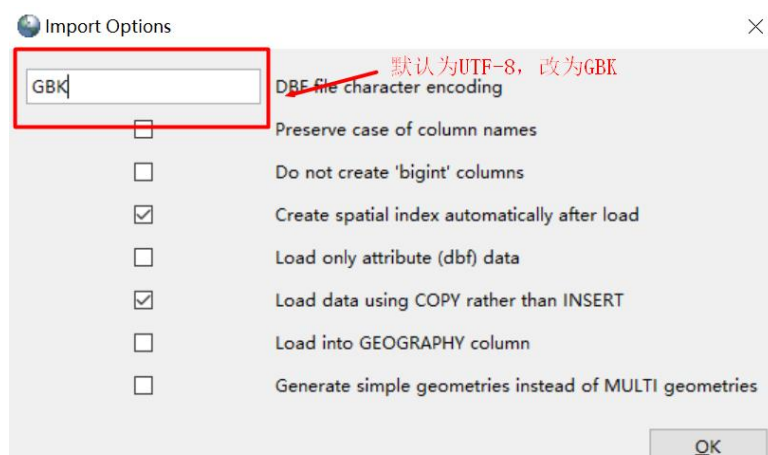


图 18 设置字符编码

(3) 导入 shp 文件:

单击“Add File”，选择要导入的 shp（路径不能有中文，否则可能会失败），最后单击“Import”即可。这里导入的数据有水质监测站点数据（**stations.shp**）与长江流域河流数据（**yangtze_river.shp**）。

5.1.3 PostGIS 中的空间数据存储

空间数据存储于空间数据库 **postgis_24_sample** 中。其中水质监测站点数据存储于表 **stations** 中，长江流域河流数据存储于表 **yangtze_river** 中。空间信息统一存储于 **geom** 字段中。

水质监测站点数据的逻辑数据结构为：stations (id, 点位名称, 河流名称, 断面情况, ph, do_mg_l, codmn_mg_l, nh3_n_mg_l, 本周水质, 上周水质, 主要污染指标, geom, 创建表人员, 更新表人员, 创建表时间, 更新表时间, ph_fig, do_fig, codmn_fig, nh3_n_fig); 物理数据结构如下图。

	gid [PK] integer	点位名称 character varying (254)	河流名称 character varying (254)	断面情况 character varying (254)	ph numeric	do_mg_l numeric	codmn_mg_l numeric	nh3_n nume
1	1	攀枝花龙洞	长江	[null]	5000000000	9900000000	1.3000000000	0.06000
2	2	乐山岷江大桥	岷江	与大渡河汇合前	4000000000	5500000000	1.7000000000	0.1600
3	3	宜宾凉姜沟	岷江	入长江前	8000000000	2600000000	2.7000000000	0.2200
4	4	泸州沱江二桥	沱江	入长江前	6000000000	8600000000	5.5000000000	0.1600
5	5	赤水鲢鱼溪	赤水河	黔川省界	5000000000	4800000000	1.3000000000	0.1800
6	6	重庆朱沱	长江	川渝省界	2000000000	4400000000	2.7000000000	0.2000
7	7	广元清风峡	嘉陵江	陕川省界	8000000000	9700000000	1.9000000000	0.1200
8	8	宜昌南津关	长江	三峡水库出口	8000000000	3400000000	1.6000000000	0.2500
9	9	丹江口胡家岭	丹江口水库	库体	3000000000	6400000000	2.1000000000	0.1900
10	10	南阳陶岔	丹江口水库	南水北调中线取水口	4000000000	0900000000	2.6000000000	0.03000
11	11	常德沙河口	澧水	入洞庭湖	0000000000	9700000000	1.7000000000	0.6800

图 19 stations 物理数据结构（部分）

长江流域河流数据的逻辑数据结构为：yangtze_river (gid, fnode_, tnode_, lpoly_, rpoly_, length, hyd2_4m_id, gbcode, name, leval_lake, geom); 物理数据结构如下图。

	gid [PK] integer	fnode_ double precision	tnode_ double precision	lpoly_ double precision	rpoly_ double precision	length double precision	hyd2_4m_ double precision	hyd2_4m_ double precision
1	1	320	320	334	329	0.143	217	
2	2	324	325	336	337	0.533	220	
3	3	326	326	392	381	0.455	221	
4	4	327	328	401	400	0.644	222	
5	5	329	329	402	433	0.029	223	
6	6	330	330	402	437	0.036	224	
7	7	331	332	423	438	0.012	225	
8	8	335	335	402	444	0.027	228	
9	9	337	337	402	455	0.039	230	
10	10	339	339	402	460	0.038	232	
11	11	340	340	402	469	0.029	233	

图 20 yangtze_river 物理数据结构（部分）

5.1.4 GeoServer 发布 WMS 服务

(1) 新建工作区：

按照 3.3(4) 的方法登录 GeoServer 后，点击左侧栏“数据-工作区”，新建工作区 yangtze_river，并设置其命名空间用于后续访问。



图 21 创建工作区

(2) 新建数据存储：

点击左侧栏“数据-数据存储”，新建矢量数据源 yangtze_river_1，并按照下图设置连接参数，使 GeoServer 与 4.1.3 中创建的空间数据库相连接。



图 22 新建数据存储

(3) 新建图层:

点击左侧栏“数据-图层”，选择上一步建立的数据存储，可看到下方出现了上一步连接的空间数据库中的表名称。选择其中的 **yangtze_river**，即长江流域河流数据进行发布；设置坐标参考为 WGS 84 参考系（不要设成 3827，否则后续转换会比较麻烦），自动计算边框即可。



图 23 新建图层



图 24 设置坐标参考及边框

(4) 设置地图符号效果:

打开 uDig, 单击 File-Import, 打开本地 yangtze_river.shp 文件; 右键点击图层, 选择 Change Style, 在 Style Properties 中设置线宽、颜色及透明度, 并导出为.sld 格式。(shp 文件名最好不含中文, 否则在制作 XML 后会出现中文, 但是 GeoServer 不支持中文文件名和文件路径)。

在 GeoServer 中点击左侧栏“数据-Styles”, 点击下方“选择文件”导入该 sld 文件; 打开(3)中新建的图层, 在“发布”选项中设置 WMS 图层样式。

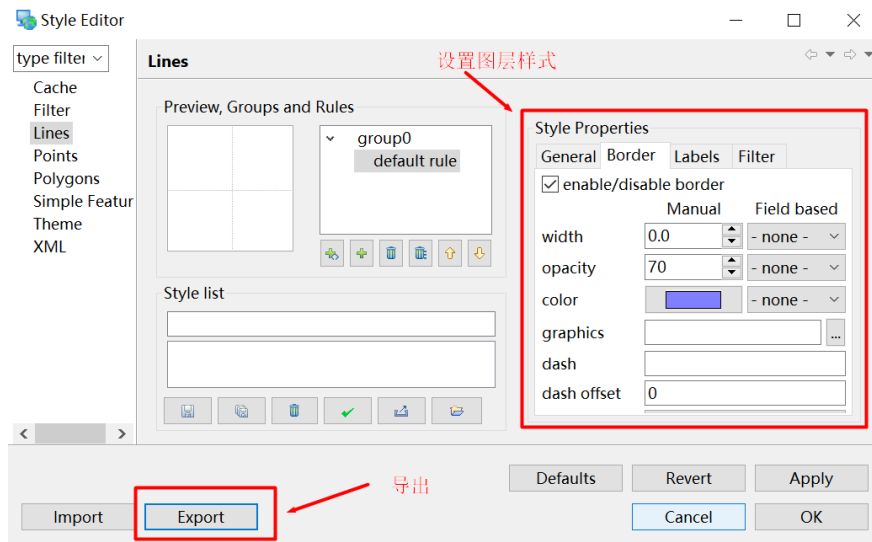


图 25 uDig 中设置图层样式

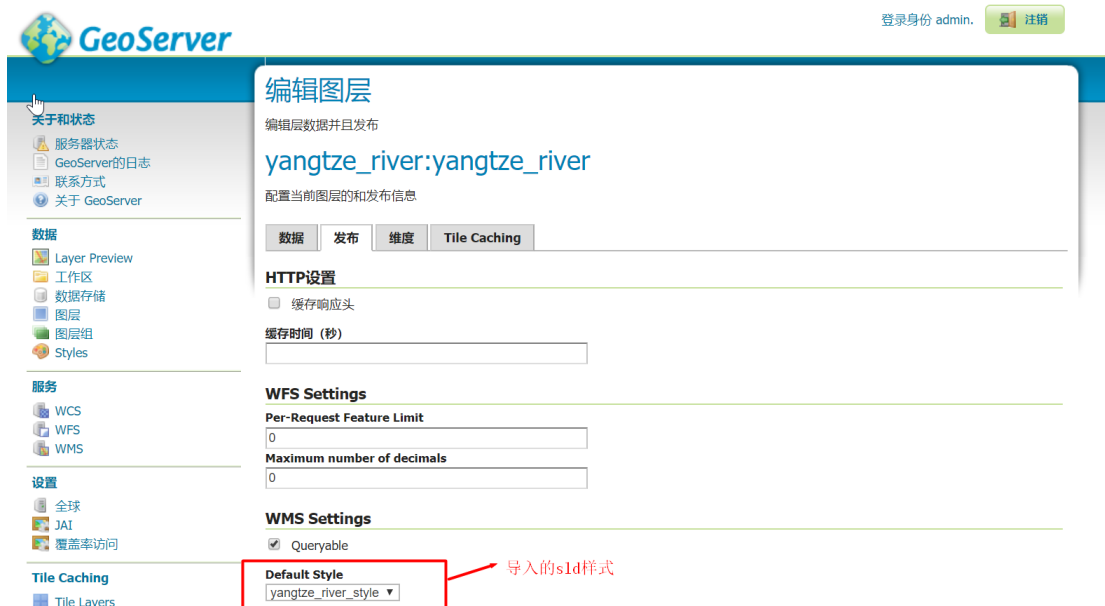


图 26 GeoServer 添加 style

(5) 图层预览:

在 GeoServer 中点击左侧栏“数据-Layer Preview”，进入地图预览功能界面；找到发布的 yangtze_river 图层，点击对应的 OpenLayers 进行预览。图层参数可在对应 url 中查看。

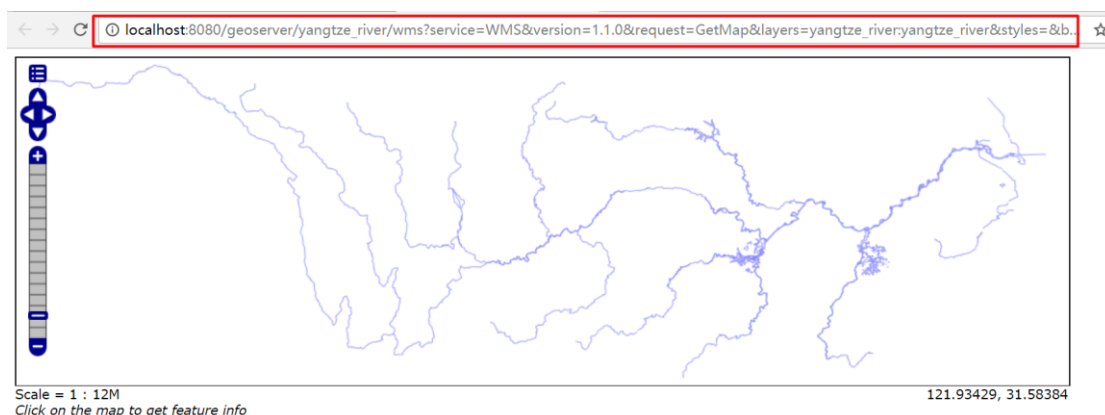


图 27 yangtze_river 图层预览

5.2 用户信息登录管理

数据库中新建普通用户权限表 **o_user** 及管理员权限表 **manager**，用于存储具有登录权限的用户信息；其逻辑数据结构为：o_user/manager (id, username, password)，物理数据结构如下。

	id [PK] integer	username character varying (254)	password character varying (254)
1	1	lydia	111

图 28 o_user/manager 物理数据结构

6 功能模块实现

6.1 登录校验

Cookie 与 Session 是 Web 程序中常用的会话跟踪技术：Cookie 通过在客户端记录信息确定用户身份，可永久存储，除非用户手动删除；Session 通过在服务器端记录信息确定用户身份，仅存储于服务器的一次会话，当用户会话结束（浏览器关闭）后，Session 将被销毁。

Java Cookie/Session 机制运行逻辑如下：用户第一次在登录界面（Login.jsp）输入用户名和密码并勾选“记住密码”时，若用户输入的用户名和密码在数据库中校验成功，则将相应的用户名和密码添加到 Cookie 和 Session；若校验不成功则刷新登录界面。用户再次打开登录界面时，从 Cookie 中获取保存的用户名和密码，填充到输入框中，余下 Session 校验流程同上。

主页面（User.jsp 和 Manager.jsp）由于设置了 Session 校验，未登录用户直接在网页中输入 User.jsp 或 Manager.jsp 的 url 将直接跳转到登录界面，登录校验成功后方可正常查看。

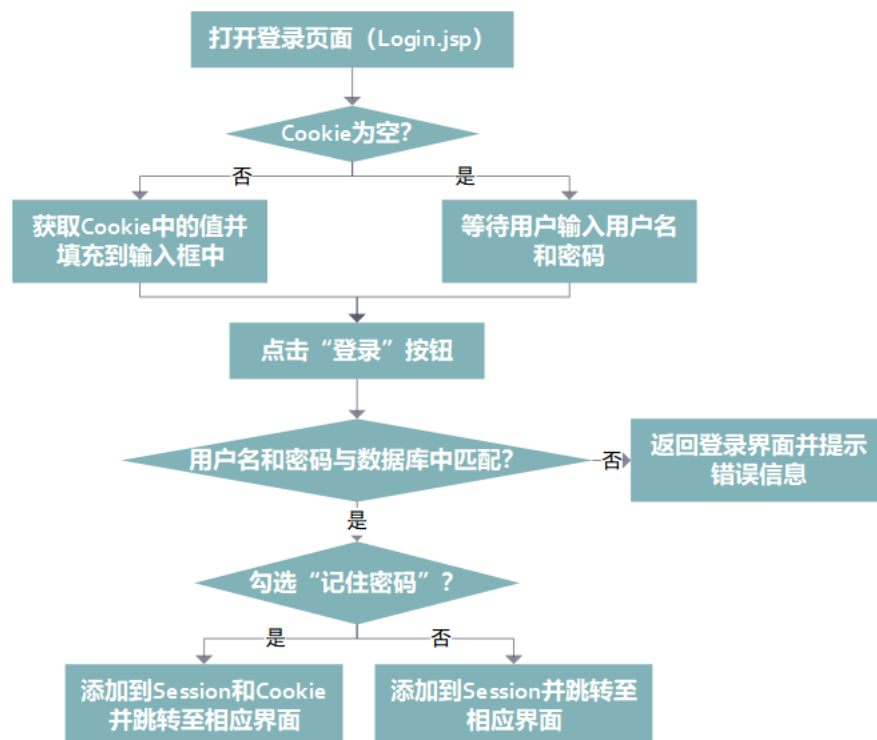


图 29 登录校验流程图

6.2 图层展示

OpenLayers 是一个专为 Web GIS 客户端开发提供的 JavaScript 类库包，用于实现标准格式发布的地图数据访问。利用 OpenLayers3 进行图层显示，包括 OSM、WMS、GeoJSON 图层；其中，OSM 图层为基础底图，WMS 图层为长江流域河流图层，GeoJSON 图层为水质监测站点图层。

(1) 引入 OpenLayers3 框架：

在官网（<http://openlayers.org/>）下载 OpenLayers3 开发包，并复制到工程 WebContent 目

录下；新建 user.jsp 和 manager.jsp，作为登录角色分别为普通用户和管理员的显示界面，并引入必要文件。

```
<link rel="stylesheet" href="css/ol.css" type="text/css"></link>
<script src="js/ol.js"></script>
```

图 30 引入 OpenLayers3 框架

(2) 初始化地图容器：

初始化 ol.Map 对象并设置参数，主要包括 target（DOM 中地图容器的 id）、view（视图）和 controls（控件）。

在 view 中可设置地图投影、地图中心坐标点和初始化显示层级；这里将地图投影参数设置为“EPSG:4326”，即 WGS 84 坐标系。

在 controls 中添加全屏控件（FullScreen）和比例尺（ScaleLine）。

(3) 添加 OSM 图层：

新建 ol.layer.Tile 图层对象，设置 source 为 ol.source.OSM 对象，并添加到地图容器 map 中。

```
<!--初始化地图容器-->
var map = new ol.Map({
  layers: [
    new ol.layer.Tile({
      source: new ol.source.OSM()
    })
  ],
  target: 'map',
  view: new ol.View({
    projection: 'EPSG:4326',
    center: [99.52, 29.72],
    zoom: 5
  }),
  controls: ol.control.defaults().extend([
    new ol.control.FullScreen(), //全屏控件
    new ol.control.ScaleLine(), //比例尺
  ])
});
```

图 31 初始化地图容器并添加 OSM 图层

(4) 添加 WMS 图层：

新建 ol.layer.Tile 图层对象，设置 source 为 ol.source.TileWMS 对象；其中，参数 url 和 LAYERS 可由 4.1.4(5)中的 WMS 图层 url 获得。最后将图层添加到地图容器 map 中。

```
<!--添加WMS图层-->
var yangtze_river = new ol.layer.Tile({
  title: "yangtze_river",
  source: new ol.source.TileWMS({
    url: 'http://localhost:8080/geoserver/yangtze_river/wms',
    params: {'LAYERS': 'yangtze_river:yangtze_river'}
  })
});
map.addLayer(yangtze_river);
```

图 32 添加 WMS 图层

(5) 添加 GeoJSON 图层：

新建 ol.layer.Vector 图层对象，调用 StationDAO 类函数 getStationGeom()，得到包含所有站点坐标的 GeoJSON 对象；在前端设置点位样式，利用 OpenLayers 进行加载。最后将图层添加到地图容器 map 中。

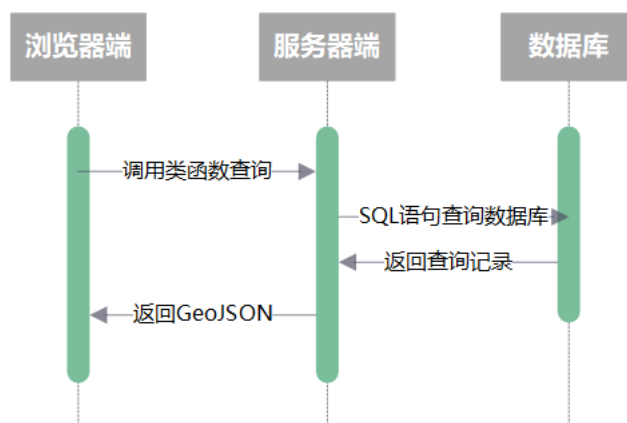


图 33 GeoJSON 图层显示时序图

6.3 数据库编辑

对 PostgreSQL 数据库的操作主要运用了 **JDBC API**。它是一个 Java API，可以访问任何类型的数据库的数据，尤其是存储在关系数据库中的数据。

在 ConnUtil 类中，设置数据驱动、数据库地址、PostgreSQL 的用户名和密码，获得数据库连接。在 StationDAO 类中，利用 INSERT、UPDATE、DELETE 语句，实现对站点的增删改。前后端通信运用表单和相应的 Servlet 完成。

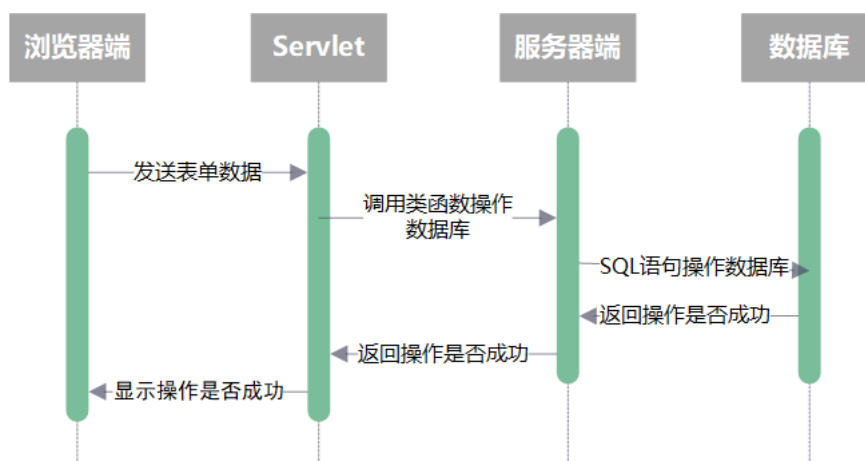


图 34 数据库编辑

6.4 站点查询

(1) 引入 jQuery 库:

jQuery 是一个通用的开源 JavaScript 函数库，可实现 ajax 异步请求方式。

在官网 (<http://jquery.com/download/>) 下载 jquery-3.3.1.min.js，复制到工程 WebContent/js 目录下，并在 user.jsp 中引入。

(2) 条件查询:

在侧边栏输入待查询的站点名称，并将其设置为 ajax 的 data 参数，用 post 方式发送请求；在 QueryServlet 中接收参数，调用服务器端 StationDAO 类中的 getQueryResult 函数，并

将查询结果重写为 JSON 格式返回；最后在浏览器端将 ajax 返回结果显示在侧边栏中。

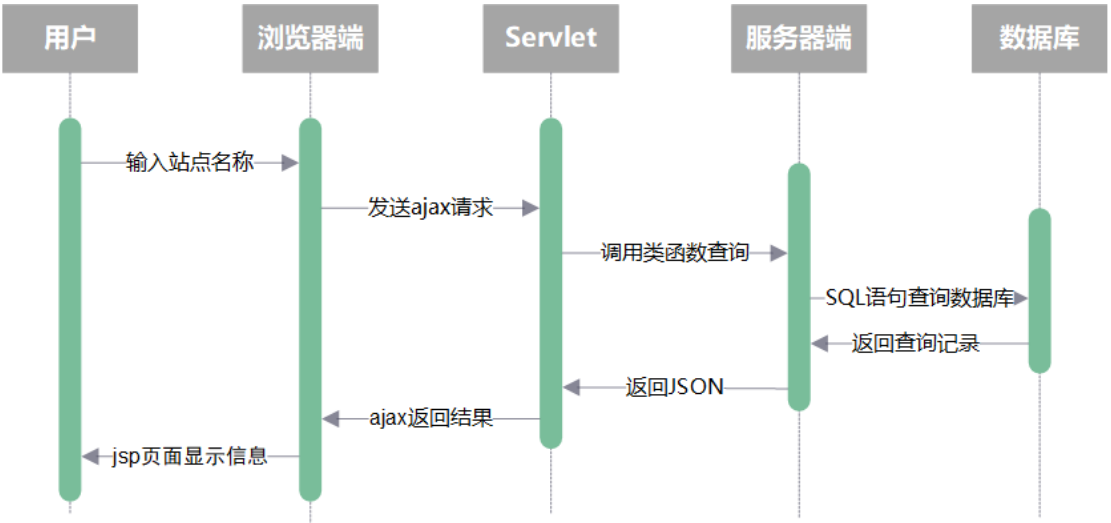


图 35 条件查询时序图

(3) 点击查询：

注册点击事件，在用户点击地图时捕捉点击位置的经纬度，并将其设置为 ajax 的 data 参数，用 post 方式发送请求；在 ClickServlet 中接收参数，调用服务器端 StationDAO 类中的 getClickResult 函数，并将查询结果重写为 JSON 格式返回；最后在浏览器端将 ajax 返回结果显示在侧边栏中。

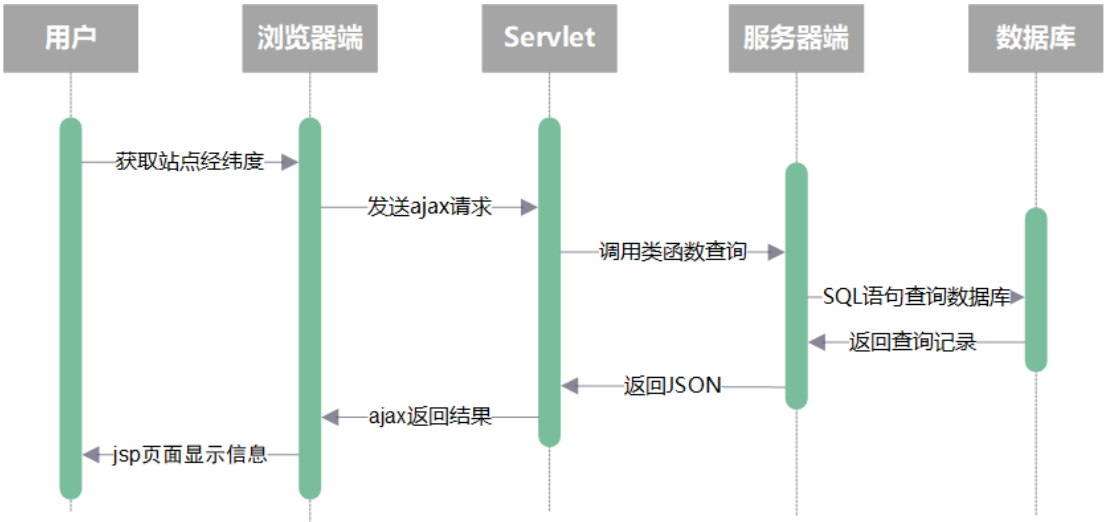


图 36 点击查询时序图

6.5 水质参数统计图表展示

(1) 引入 ECharts 库：

ECharts 是百度地图开发的一款基于 HTML5 的图形库，支持饼图、环形图、折线图、面积图、柱形图、条形图等。

在官网 (<http://echarts.baidu.com>) 下载 echarts.js, 复制到工程 WebContent/js 目录下, 并在 user.jsp 中引入。

(2) 设置参数, 初始化图表:

将条件查询和点击查询的 ajax 请求结果中的 PH、DO、CODMn、NH₃-N 值作为绘图参数, 并设置坐标轴标签; 最后将图表放入显示查询结果的侧边栏中。

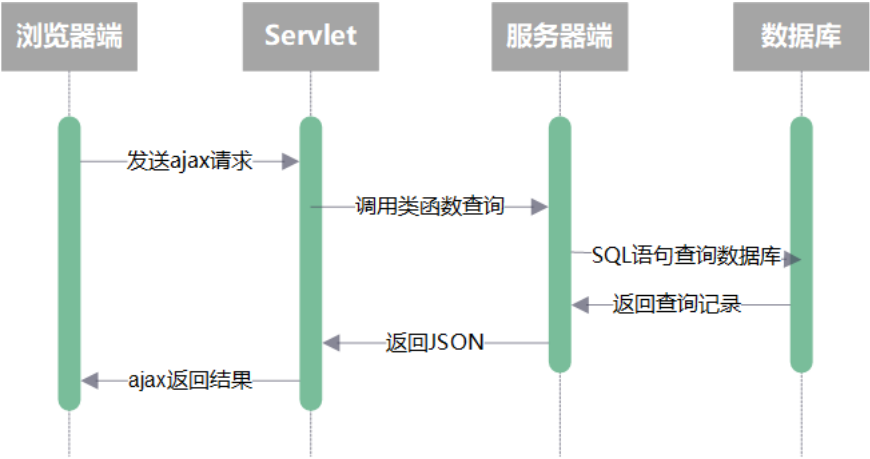


图 37 统计图表显示时序图

6.6 水质分类热力图展示

新建 ol.layer.Heatmap 图层对象, 根据下拉框所选定的值不同调用不同的 StationDAO 类函数 (getHeatmap1()/getHeatmap2()/getHeatmap3()), 得到包含相应水质类型站点坐标的 GeoJSON 对象, 并将图层添加到地图容器 map 中。

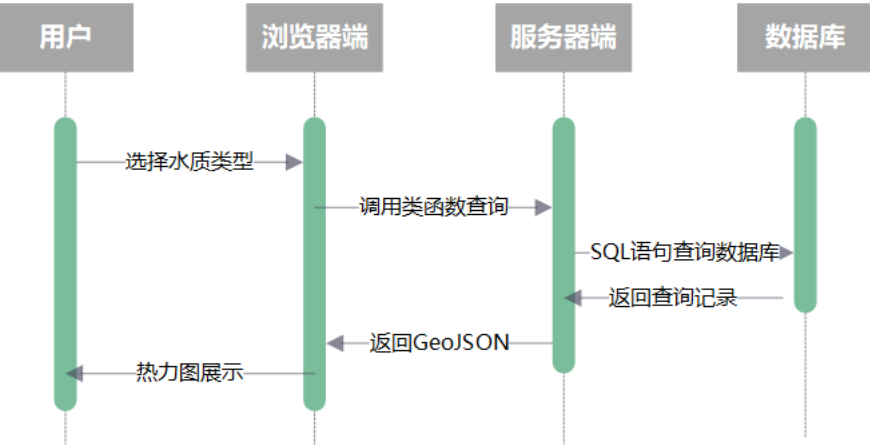


图 38 水质分类热力图展示

7 成果展示

7.1 登录校验



图 39 登录校验

7.2 图层展示

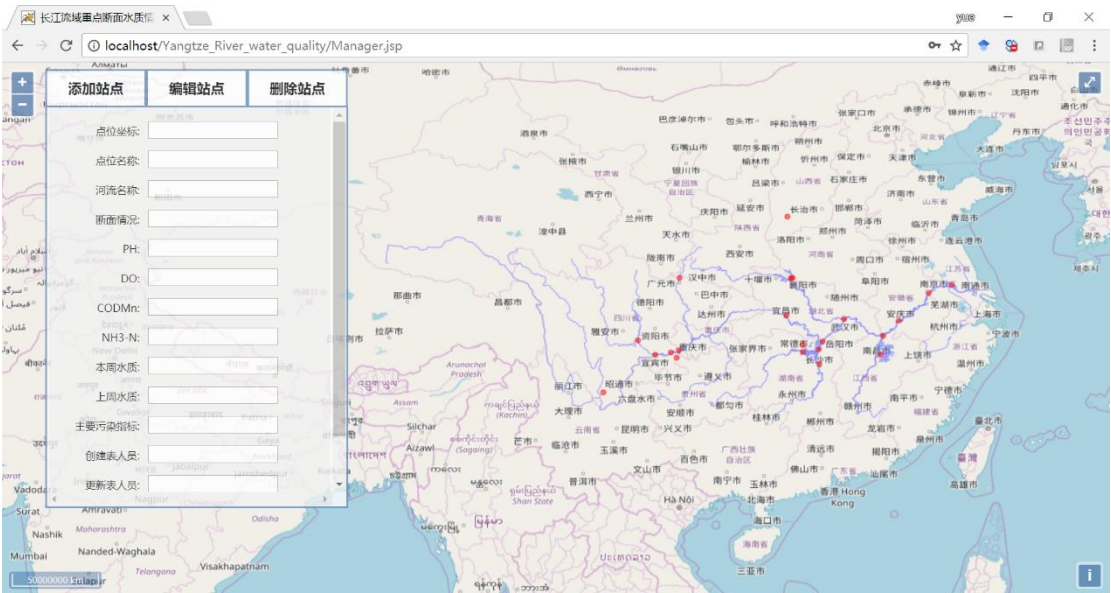


图 40 图层展示，包括 OSM、WMS、GeoJSON

7.3 数据库编辑

(1) 添加站点:

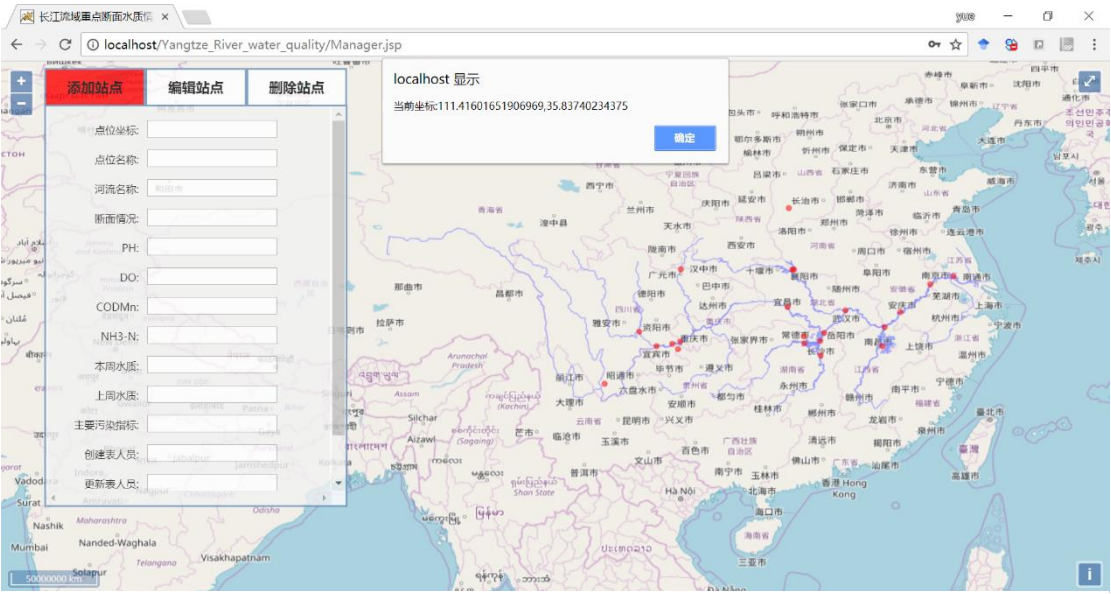


图 41 添加站点

(2) 编辑站点:

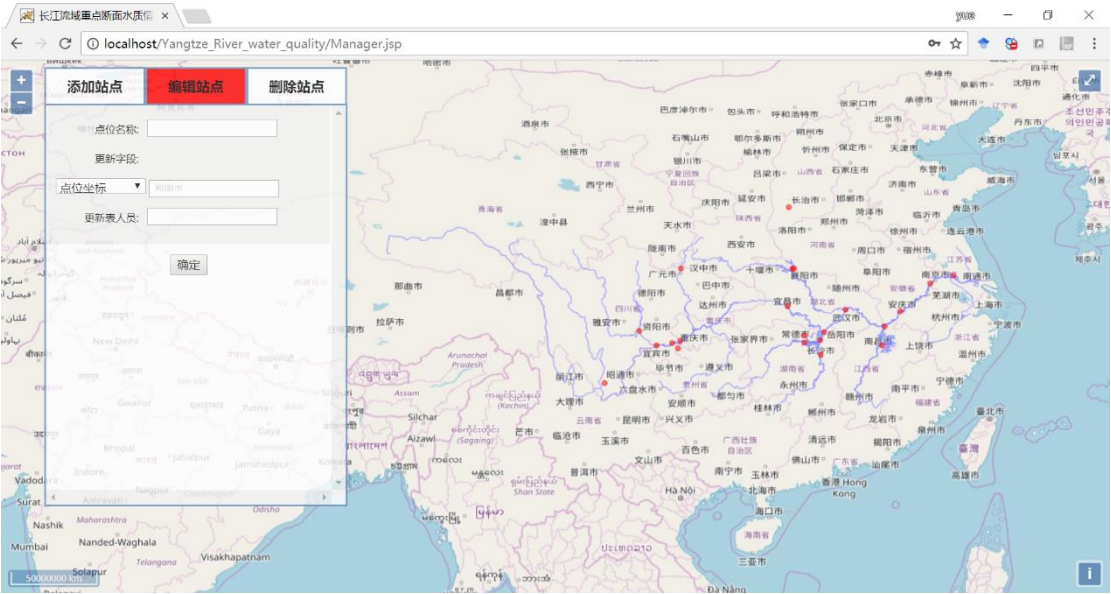


图 42 编辑站点

(3) 删除站点:

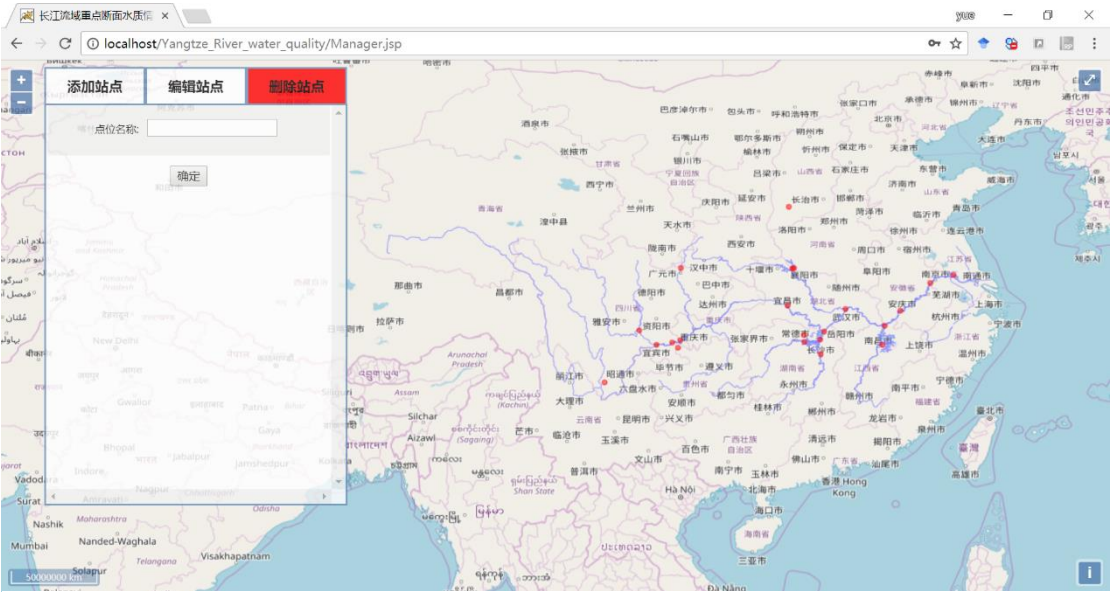


图 43 删除站点

7.4 站点查询与水质参数统计图表展示

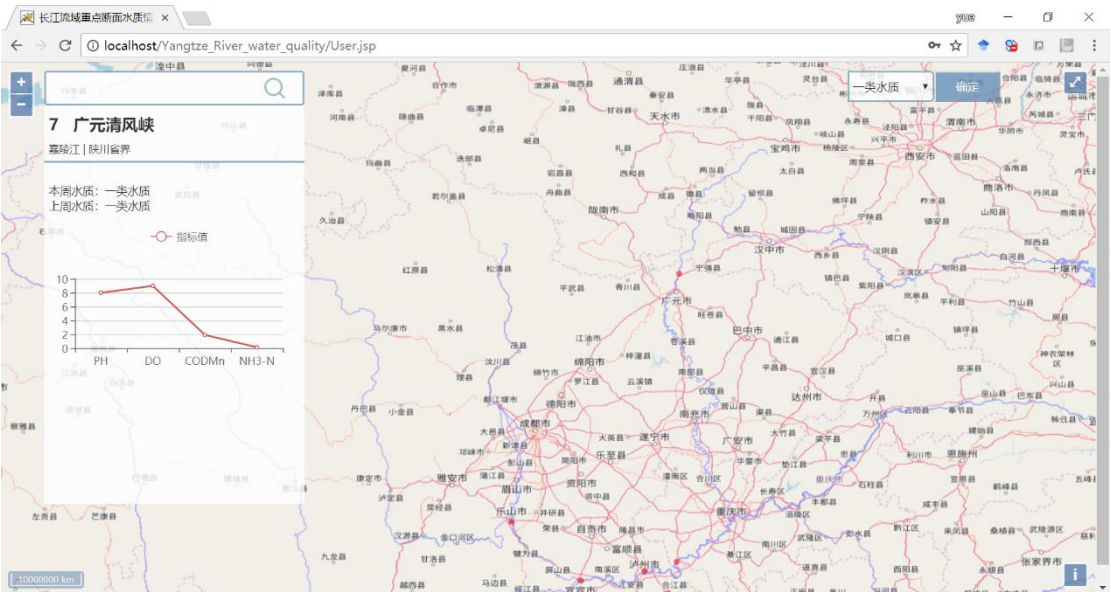


图 44 站点查询及水质参数统计图表展示

7.5 水质分类热力图展示

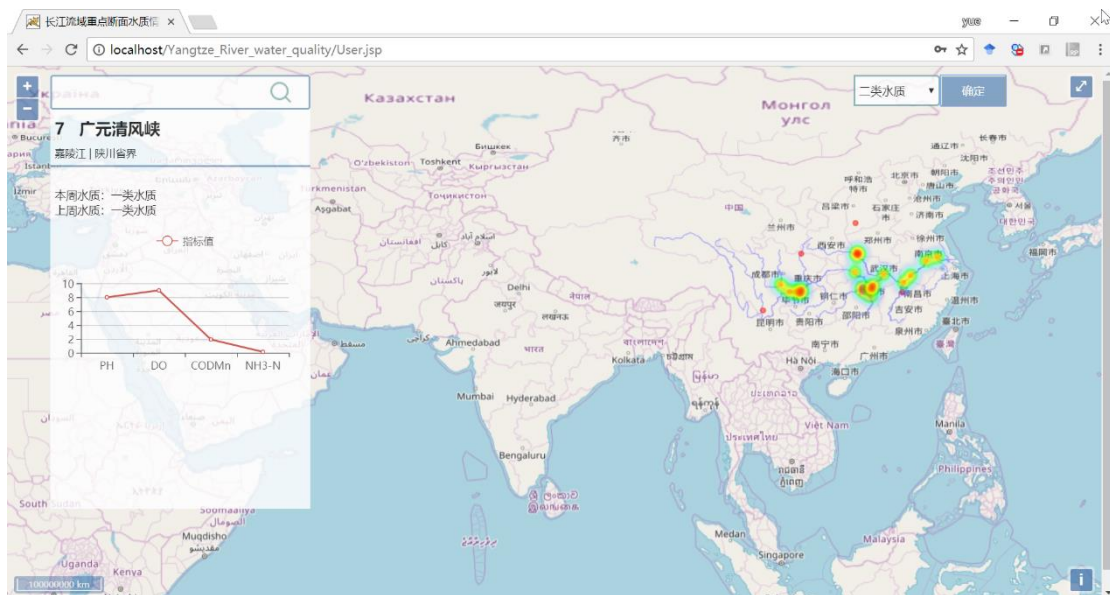


图 45 水质分类热力图展示

8 问题及解决办法

(1) 配置环境中的各种报错：

一定要注意版本的对应问题，包括 Tomcat 与 JDK，PostgreSQL 与 PostGIS。还要注意各种软件之间 64bit 或 32bit 应统一；这里使用的全为 32bit。

(2) 在 Eclipse 中运行程序时显示 Tomcat 端口被占用：

安装 Tomcat 时，为避免端口与 GeoServer 自带的服务器端口相冲突，将端口改为 80。

(3) 在 Web 工程下未找到 classes 文件夹及 web.xml 配置文件：

Eclipse 默认生成 classes 文件夹的路径不在 WebContent 中，且默认不生成 web.xml 文件，故在创建工程时应手动修改 classes 路径，并勾选生成 web.xml 文件选项。

(4) 配置 Servlet 时在 web.xml 文件中配置 Servlet 的路径和类名报错：

Eclipse 使用向导创建 Servlet 时，已自动在文件头添加 `@WebServlet("[Servlet 路径]")`，故不能再在 web.xml 中再次配置。

(5) 在 Servlet 中使用 `request.getParameter()` 能接收到字符串，但无法在数据库中进行查询：

在接收参数时应使用强制类型转换将参数转换为 String，即改为 `(String) request.getParameter()`。

9 创新与贡献

(1) 实用性较高：

能结合 WebGIS 相关知识解决实际问题，相关功能的融合也较为恰当、合理，应用。

(2) 界面友好:

侧边栏设置为半透明，控件布局合理，在一定程度上方便用户操作、使用地图。

10 优化与改进

(1) 数据库编辑:

添加站点时，现可以点击图上任意位置弹框显示其经纬度，然后用户再将该数值填入输入框内；可以改进为点击图上任意位置添加标注点，并直接将该坐标作为 `ajax` 参数，无需用户手动输入。

编辑站点时，现为直接输入站点名称对其属性进行修改，优化方案应为：点击图上站点，在侧边栏显示其现有信息，用户可在此基础上对其属性进行修改。

(2) 站点查询:

目前条件查询仅实现对属性信息的输出，未在图上对相应点位进行标记，优化方案为：将输出的 `geom` 字段转换为坐标点形式，并新增一个 `marker` 图层对点位进行标注；同时，将地图中心点重新设定为该点位坐标，并修改缩放层级，实现对点位的突出展示。

(3) 水质参数统计图表展示:

由于此处并不是展示某一水质参数的时序变化，故采用折线图并不合适；应改为条形图，并标注参数值。

(4) 水质分类热力图展示:

由于研究的水体是呈线状分布的，故直接调用 `OpenLayers` 中的函数创建热力图（面状分布）不甚合适；后期有条件的话可以尝试结合水质扩散模型对热力图进行优化。

(5) 补充功能:

可引入 `turf.js`，利用其中缓冲区分析功能求取不同断面上站点的监测有效范围，并添加城市图层（可在 `OSM` 上获取），得到各个站点监测覆盖的城市。该功能可实现水质监测站点盲区的判断，对优化站点布局有一定指导作用。

(6) 成果部署到云空间:

由于时间关系，目前仅完成了云控件环境的配置，后期应将成果打包部署到云空间。