

3.13作业

1.结合上述所讲的echarts与后台交互数据,查看官网echarts:实例----官方实例,在其中找一个符合你身份echarts图形(与后台交互),除了今天讲的柱状图

1.1 数据 [

```
{value: 335, name: '直接访问'},  
{value: 310, name: '邮件营销'},  
{value: 234, name: '联盟广告'},  
{value: 135, name: '视频广告'},  
{value: 1548, name: '搜索引擎'}
```

]

1.2 ['直接访问', '邮件营销', '联盟广告', '视频广告', '搜索引擎']

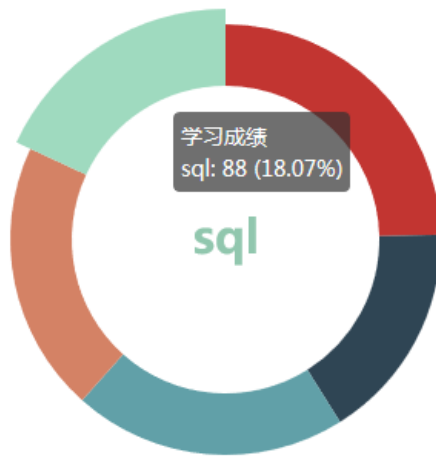
2.编写实体类直接写value对应数值,然后name对应名称即可

```
@Data  
public class News {  
    private String name;  
    private Integer value;  
    public News() {  
    }  
    public News(String name, Integer value) {  
        this.name = name;  
        this.value = value;  
    }  
}
```

3.用arrayList返回前端即可

```
response.setContentType("application/json;charset=utf-8;");  
List<News> list=new ArrayList<>();  
News news=new News( name: "直接访问", value: 335);  
News news1=new News( name: "邮件营销", value: 310);  
News news2=new News( name: "联盟广告", value: 234);  
News news3=new News( name: "视频广告", value: 135);  
News news4=new News( name: "搜索引擎", value: 1548);  
list.add(news);  
list.add(news1);  
list.add(news2);  
list.add(news3);  
list.add(news4);  
System.out.println(JSON.toJSONString(list));  
response.getWriter().write(JSON.toJSONString(list));  
}
```

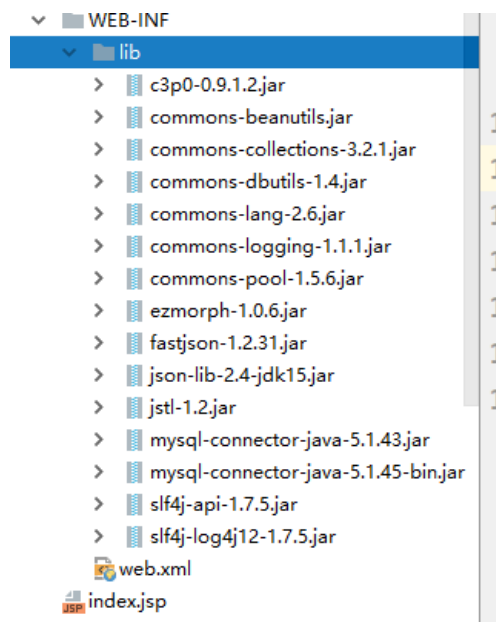
- 数学
- 语文
- 英语
- java
- sql



4.结果图

作业2 三级联动

1.先创表



2.创建项目,然后导入相应jar包

3.先写页面

4.编写c3p0的xml配置文件,填写配置信息

5.创建DBUtil的工具类

6.编写分别的实体类

7.在编写dao层分别获取省份,城市,乡镇的方法

```

@Override
public List<City> getCity(Integer pid) {
    List<City> list=null;
    String sql="select * from city where pid=? ";
    try {
        Object[] param={pid};
        list= queryRunner.query(sql,new BeanListHandler<City>(City.class));
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return list;
}

@Override
public List<Country> getCountry(Integer cid) {
    List<Country> list=null;
    String sql="select * from country where cid=?";
    try {
        Object[] param={cid};
        list= queryRunner.query(sql,new BeanListHandler<Country>(Country.class));
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return list;
}

```

8.在测试类中测试dao层方法

```

public class Test {
    public static void main(String[] args) {
        BizDao bizDao=new BizDaoImpl();
        List<Province> province = bizDao.getProvince();
        System.out.println(province.toString());
    }
}

```

Test > main()

```

Test (1) x
Fri Mar 13 22:22:04 CST 2020 WARN: Establishing SSL connection without server's identity verification is not recomme
Fri Mar 13 22:22:04 CST 2020 WARN: Establishing SSL connection without server's identity verification is not recomme
Fri Mar 13 22:22:04 CST 2020 WARN: Establishing SSL connection without server's identity verification is not recomme
[Province(pid=11, pname=北京), Province(pid=12, pname=天津), Province(pid=13, pname=河北), Province(pid=14, pname=山西
Fri Mar 13 22:22:04 CST 2020 WARN: Establishing SSL connection without server's identity verification is not recomme
Fri Mar 13 22:22:04 CST 2020 WARN: Establishing SSL connection without server's identity verification is not recomme

```

9.编写服务层调用dao层方法

```

* Create By WeiBin on 2020/3/13 22:23
*/
public class BizServiceImpl implements BizService {
    private BizDao bizDao=new BizDaoImpl();

    @Override
    public List<Province> getProvince() {
        return bizDao.getProvince();
    }

    @Override
    public List<City> getCity(Integer pid) {
        return bizDao.getCity(pid);
    }

    @Override
    public List<Country> getCountry(Integer cid) {
        return bizDao.getCountry(cid);
    }
}

```

10.编写servlet调用服务层

```

@javax.servlet.annotation.WebServlet(name = "ProvinceServlet",urlPatterns = {"/getProvince"})
public class ProvinceServlet extends javax.servlet.http.HttpServlet {
    protected void doPost(javax.servlet.http.HttpServletRequest request, javax.servlet.http.HttpServletResponse r
        response.setContentType("application/json;charset=utf-8");
        BizService bizService=new BizServiceImpl();
        List<Province> province = bizService.getProvince();
        PrintWriter writer = response.getWriter();
        writer.write(JSON.toJSONString(province));
        writer.flush();
        writer.close();
    }
}

```

11.在页面中调用servlet即可

```

$(function () {
    $("#province").change(function () {
        var pid=$("#province").val();
        $("#city").children().remove();
        province(pid);
    });
    $("#city").change(function () {
        var cid=$("#city").val();
        $("#country").children().remove();
        city(cid);
    });
});
// 绑定省份
function province(pid) {
    alert("进入了省份");
    $.getJSON(
        "getCity",
        {"pid":pid},
        function (data) {
            $("#country").append("<option value='-1'>请选择城市</option>");
            $(data).each(function (index,obj) {
                var cityId=obj.cid;
                var cityName=obj.cname;
                var option="<option value='"+cityId+"'>"+cityName+"</option>";
                $("#city").append(option);
            });
        }
    );
}

```

12.效果图

省份 城市 乡镇

请选择乡镇

江岸区

江汉区

硚口区

汉阳区

武昌区

青山区

洪山区

东西湖区

汉南区

蔡甸区

江夏区

黄陂区

新洲区

13.整体项目结构

