# 图形报表

1：了解ECharts

2：掌握ECharts实现会员数量折线图的实现过程

# ****第一章. 图形报表ECharts****

jfreechar（api编程）–>HighChart报表–>百度ECharts

### 【目标】

了解ECharts

### 【路径】

1：ECharts简介

2：5分钟上手ECharts

3：查看ECharts官方实例

### 【讲解】

## 1.1. ECharts简介

ECharts缩写来自Enterprise Charts，商业级数据图表，是百度的一个开源的使用JavaScript实现的数据可视化工具，可以流畅的运行在 PC 和移动设备上，兼容当前绝大部分浏览器（IE8/9/10/11，Chrome，Firefox，Safari等），底层依赖轻量级的矢量图形库 [ZRender](https://github.com/ecomfe/zrender)，提供直观、交互丰富、可高度个性化定制的数据可视化图表。

官网：<https://echarts.baidu.com/>

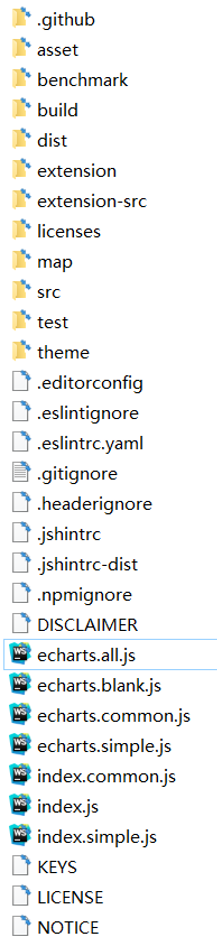
下载地址：<https://echarts.baidu.com/download.html>



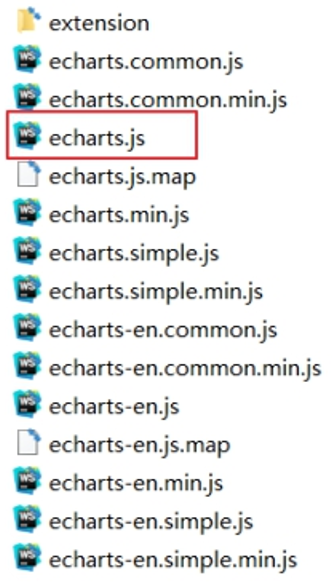
下载完成可以得到如下文件：



解压上面的zip文件：



我们只需要将dist目录下的echarts.js文件（已经压缩）引入到页面上就可以使用了

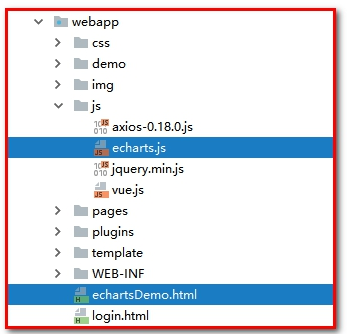


## 1.2. 5分钟上手ECharts

我们可以参考官方提供的5分钟上手ECharts文档感受一下ECharts的使用方式，地址如下：

<https://echarts.apache.org/zh/tutorial.html#5%20%E5%88%86%E9%92%9F%E4%B8%8A%E6%89%8B%20ECharts>

第一步：在meinian\_web中创建html页面并引入echarts.js文件



echartsDemo.html

<!DOCTYPE html>

<**html** **lang**="en">

<**head**>

<**meta** **charset**="UTF-8">

<**title**>Title</**title**>

</**head**>

<!-- 引入 ECharts 文件 -->

<**script** **src**="js/echarts.js"></**script**>

<**body**>

</**body**>

</**html**>

第二步：在页面中准备一个具备宽高的DOM容器。

<**body**>

<!-- 为 ECharts 准备一个具备大小（宽高）的 DOM -->

<**div** **id**="main" **style**="width: 600px;height:400px;"></**div**>

</**body**>

第三步：通过echarts.init方法初始化一个 echarts 实例并通过setOption方法生成一个简单的柱状图

<script type="text/javascript">

// 基于准备好的dom，初始化echarts实例

**var** myChart = echarts.init(document.getElementById('main'));

// 指定图表的配置项和数据

**var** option = {

title: {

text: 'ECharts 入门示例'

},

tooltip: {},

legend: {

data:['销量']

},

xAxis: {

data: ["衬衫","羊毛衫","雪纺衫","裤子","高跟鞋","袜子"]

},

yAxis: {},

series: [{

name: '销量',

type: 'bar',

data: [5, 20, 36, 10, 10, 20]

}]

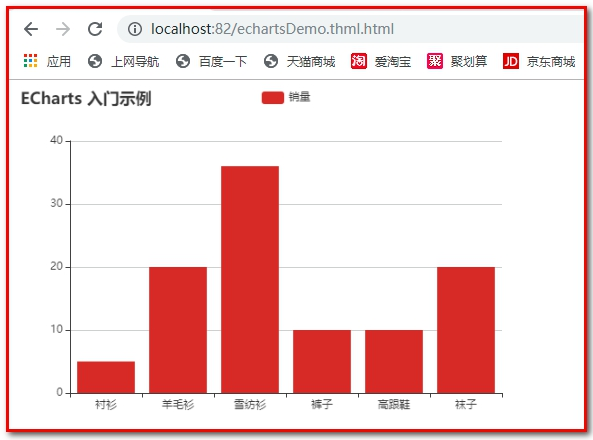
};

// 使用刚指定的配置项和数据显示图表。

myChart.setOption(option);

</**script**>

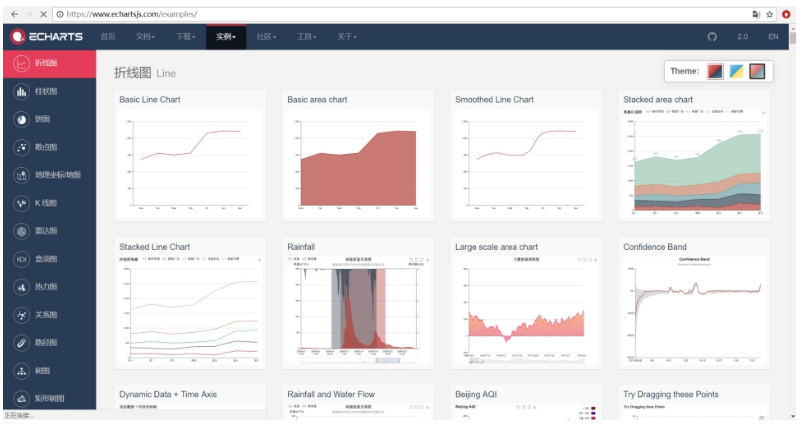
效果如下：



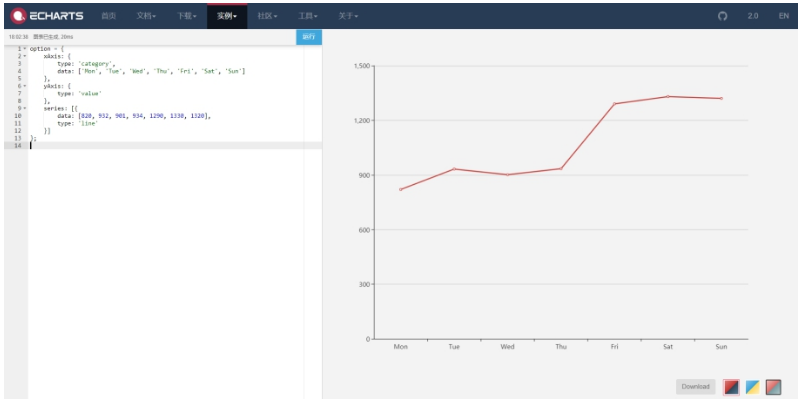
## 1.3. 查看ECharts官方实例

ECharts提供了很多官方实例，我们可以通过这些官方实例来查看展示效果和使用方法。

官方实例地址：<https://echarts.apache.org/examples/zh/index.html>



可以点击具体的一个图形会跳转到编辑页面，编辑页面左侧展示源码（js部分源码），右侧展示图表效果，如下：



要查看完整代码可以点击右下角的Download按钮将完整页面下载到本地。

通过官方案例我们可以发现，使用ECharts展示图表效果，关键点在于确定此图表所需的数据格式，然后按照此数据格式提供数据就可以了，我们无须关注效果是如何渲染出来的。

在实际应用中，我们要展示的数据往往存储在数据库中，所以我们可以发送ajax请求获取数据库中的数据并转为图表所需的数据即可。

### 【小结】

1：ECharts简介

* ECharts百度开发的一套商业报表工具(库)

2：5分钟上手ECharts

3：查看ECharts官方实例

学习方法

* 看官网, 关注案例，无需关注api，根据项目需求，选择合适报表

# ****第二章. 会员数量折线图****

### 【目标】

使用ECharts实现注册会员数量折线图

### 【路径】

1：需求分析

2：前台代码

（1）导入ECharts库

（2）参考官方实例导入折线图

3：后台代码

（1）ReportController类

（2）MemberService服务接口

（3）MemberServiceImpl服务实现类

（4）MemberDao接口

（5）Mapper映射文件（MemberDao.xml）

### 【讲解】

## 2.1. 需求分析

会员信息是旅游机构的核心数据，其会员数量和增长数量可以反映出机构的部分运营情况。通过折线图可以直观的反映出会员数量的增长趋势。

本章节我们需要展示过去一年时间内每个月的会员总数据量。展示效果如下图：



导入sql语句

**DROP** **TABLE** **IF** **EXISTS** `t\_member`;

**CREATE** **TABLE** `t\_member` (

`id` int(11) **NOT** NULL AUTO\_INCREMENT,

`fileNumber` varchar(32) **DEFAULT** NULL,

`name` varchar(32) **DEFAULT** NULL,

`sex` varchar(8) **DEFAULT** NULL,

`idCard` varchar(18) **DEFAULT** NULL,

`phoneNumber` varchar(11) **DEFAULT** NULL,

`regTime` date **DEFAULT** NULL,

`password` varchar(32) **DEFAULT** NULL,

`email` varchar(32) **DEFAULT** NULL,

`birthday` date **DEFAULT** NULL,

`remark` varchar(128) **DEFAULT** NULL,

**PRIMARY** **KEY** (`id`)

) **ENGINE**=**InnoDB** AUTO\_INCREMENT=93 **DEFAULT** **CHARSET**=utf8;

**INSERT** **INTO** `t\_member` **VALUES** ('82', null, '小明', '1', '123456789000999999', '18544279942', '2019-03-08', null, null, null, null);

**INSERT** **INTO** `t\_member` **VALUES** ('83', null, '美丽', '1', '132333333333333', '13412345678', '2019-03-11', null, null, null, null);

**INSERT** **INTO** `t\_member` **VALUES** ('84', null, '小星星', null, null, '18343569942', '2019-03-13', null, null, null, null);

**INSERT** **INTO** `t\_member` **VALUES** ('85', null, '小强', null, null, 18344249942, '2019-03-06', null, null, null, null);

**INSERT** **INTO** `t\_member` **VALUES** ('86', null, '红康', null, null, 18544579942, '2019-04-04', null, null, null, null);

**INSERT** **INTO** `t\_member` **VALUES** ('87', null, '大海', null, null, 18544274442, '2019-02-06', null, null, null, null);

**INSERT** **INTO** `t\_member` **VALUES** ('88', null, '小婷', null, null, 18544269942, '2019-04-10', null, null, null, null);

**INSERT** **INTO** `t\_member` **VALUES** ('89', null, '晓晓', null, null, 18544339942, '2019-12-01', null, null, null, null);

**INSERT** **INTO** `t\_member` **VALUES** ('90', null, '纯生', null, null, 18544669942, '2019-12-02', null, null, null, null);

**INSERT** **INTO** `t\_member` **VALUES** ('91', null, '闰土', null, null, 18544779942, '2019-02-01', null, null, null, null);

**INSERT** **INTO** `t\_member` **VALUES** ('92', null, '波波', '2', '234234145432121345', '18019286521', '2019-10-19', null, null, null, null);

需要的sql：

#2019年05月31日之前注册会员的人数(11)

**SELECT** **COUNT**(id) **FROM** t\_member **WHERE** regTime <= '2019-05-31'

#2019年06月31日之前注册会员的人数(12)

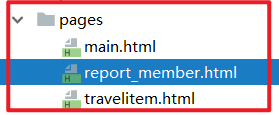
**SELECT** **COUNT**(id) **FROM** t\_member **WHERE** regTime <= '2019-06-31'

#2019年07月31日之前注册会员的人数(16)

**SELECT** **COUNT**(id) **FROM** t\_member **WHERE** regTime <= '2019-07-31'

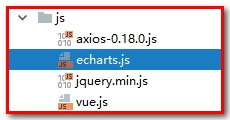
## 2.2. 前台代码

会员数量折线图对应的页面为/pages/report\_member.html。



### 2.2.1. ****导入ECharts库****

第一步：将echarts.js文件复制到health\_web工程中



第二步：在report\_member.html页面引入echarts.js文件

<**script** **src**="../js/echarts.js"></**script**>

### 2.2.2. 参照官方实例导入折线图

1：定义

<**div** **class**="app-container">

<**div** **class**="box">

<!-- 为 ECharts 准备一个具备大小（宽高）的 DOM -->

<**div** **id**="chart1" **style**="height:600px;"></**div**>

</**div**>

</**div**>

2：定义script

<script type="text/javascript">

// 基于准备好的dom，初始化echarts实例

**var** myChart1 = echarts.init(document.getElementById('chart1'));

/\*\*

\*返回Result(flag,message,data):data=Map<String,Object>

\* map：key value

\* months List<String> (["2018-12","2019-01","2019-02","2019-03","2019-04","2019-05"])

\* memberCount List<Integer> ([5, 20, 36, 40, 50, 60])

\*/

// 使用刚指定的配置项和数据显示图表。

//myChart.setOption(option);

axios.get("/report/getMemberReport.do").then((res)=>{

myChart1.setOption(

{

title: {

text: '会员数量'

},

tooltip: {},

legend: {

data:['会员数量']

},

xAxis: {

data: res.data.data.months

},

yAxis: {

type:'value'

},

series: [{

name: '会员数量',

type: 'line',

data: res.data.data.memberCount

}]

});

});

</**script**>

根据折现图对数据格式的要求，我们发送ajax请求，服务端需要返回如下格式的数据：

其中months和memberCount可以使用hashmap封装

返回Result：data、flag、message

其中data，是map集合

map集合的key：months； map集合的值：List

map集合的key：memberCount； map集合的值：List

{

"**data**":{

"**months**":["2019-01","2019-02","2019-03","2019-04"],

"**memberCount**":[3,4,8,10]

},

"**flag**":true,

"**message**":"获取会员统计数据成功"

}

## 2.3. 后台代码

### 2.3.1. ****Controller****

在meinian\_web工程中创建ReportController并提供getMemberReport方法

**package** com.atguigu.controller;

**import** com.alibaba.dubbo.config.annotation.Reference;

**import** com.atguigu.constant.MessageConstant;

**import** com.atguigu.entity.Result;

**import** com.atguigu.service.MemberService;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RestController;

**import** java.text.SimpleDateFormat;

**import** java.util.\*;

@RestController

@RequestMapping("/report")

**public** **class** **ReportController** {

@Reference

**private** MemberService memberService;

/\*\*

\* 会员数量统计

\* @return

\*/

@RequestMapping("/getMemberReport")

**public** Result **getMemberReport**() {

// 获取日历对象

Calendar calendar = Calendar.getInstance();

//根据当前时间，获取前12个月的日历(当前日历2020-02，12个月前，日历时间2019-03)

//第一个参数，日历字段

//第二个参数，要添加到字段中的日期或时间

calendar.add(Calendar.MONTH,-12);

List<String> list = **new** ArrayList<String>();

**for**(**int** i=0;i<12;i++){

//第一个参数是月份 2018-7

//第二个参数是月份+1个月

calendar.add(Calendar.MONTH,1);

list.add(**new** SimpleDateFormat("yyyy-MM").format(calendar.getTime()));

}

Map<String,Object> map = **new** HashMap<String,Object>();

// 把过去12个月的日期存储到map里面

map.put("months",list);

// 查询所有的会员

List<Integer> memberCount = memberService.findMemberCountByMonth(list);

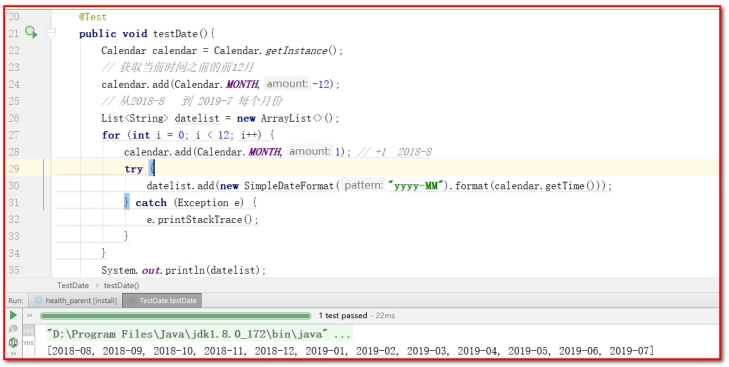
map.put("memberCount",memberCount);

**return** **new** **Result**(**true**, MessageConstant.GET\_MEMBER\_NUMBER\_REPORT\_SUCCESS,map);

}

}

计算日期，可以使用test方法测试一下



### 2.3.2. 服务接口

在MemberService服务接口中扩展方法findMemberCountByMonth

List<Integer> **findMemberCountByMonth**(List<String> monthsList);

### 2.3.3. 服务实现类

在MemberServiceImpl服务实现类中实现findMemberCountByMonth方法

**package** com.atguigu.service.impl;

**import** com.alibaba.dubbo.config.annotation.Service;

**import** com.atguigu.dao.MemberDao;

**import** com.atguigu.service.MemberService;

**import** com.atguigu.utils.DateUtils;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.transaction.annotation.Transactional;

**import** java.util.ArrayList;

**import** java.util.List;

@Service(interfaceClass = MemberService.class)

@Transactional

**public** **class** **MemberServiceImpl** **implements** **MemberService** {

@Autowired

**private** MemberDao memberDao;

// 根据月份统计会员数量

**public** List<Integer> **findMemberCountByMonth**(List<String> monthsList) {

List<Integer> memeberCountList = **new** ArrayList<Integer>();

**if**(monthsList!=**null** && monthsList.size()>0){

**for** (String months : monthsList) {

//String regTime = months+"-31";

// 获取指定月份的最后一天

String regTime = DateUtils.getLastDayOfMonth(months);

// 迭代过去12个月，每个月注册会员的数量，根据注册日期查询

Integer memeberCount = memberDao.findMemberCountBeforeDate(regTime);

memeberCountList.add(memeberCount);

}

}

**return** memeberCountList;

}

### 2.3.4. ****Dao接口****

在MemberDao接口中扩展方法findMemberCountBeforeDate

**package** com.atguigu.dao;

**public** **interface** **MemberDao** {

**public** Integer **findMemberCountBeforeDate**(String date);

}

### 2.3.5. Mapper映射文件

在MemberDao.xml映射文件中提供SQL语句

<?xml version="1.0" encoding="UTF-8" ?>

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"

"http://mybatis.org/dtd/mybatis-3-mapper.dtd" >

<**mapper** **namespace**="com.atguigu.dao.MemberDao">

<!--根据日期统计会员数，统计指定日期之前的会员数-->

<**select** **id**="findMemberCountBeforeDate" **parameterType**="string" **resultType**="int">

select count(id) from t\_member where regTime &lt;= #{value}

</**select**>

</**mapper**>

注意：在xml文件中 ， <号需要转义”&lt;”

>号需要转义”&gt;”

&号需要专业 ”&amp;”

测试

1：修改main.html，跳转到report\_member.html页面，显示图表



2：导入数据

3：测试效果



### 【小结】

1：需求分析

2：前台页面

（1）导入ECharts库

（2）参考官方实例导入折线图案例（折线图、柱状图、饼图…），参考API完成开发

3：后台代码

（1）ReportController类

（2）MemberService服务接口

（3）MemberServiceImpl服务实现类

（4）MemberDao接口

（5）Mapper映射文件（MemberDao.xml）