

# 前端编码记录：通用模块(JS)

---

## 注意事项记录：

当使用webstorm编辑的时候一定要记得在settings->Languages & Frameworks->Node.js and NPM中勾选Coding assistance for Node.js

1. 首先我们在src中新建一个utils目录，其中存放通用模块

2. 我们需要在webpack.config.js中加入一个resolve选项用于快速对文件夹src中的位置进行别名引用

```
resolve : {  
  alias : {  
    utils  : __dirname + '/src/utils',  
    page   : __dirname + '/src/pages',  
    service : __dirname + '/src/service',  
    image  : __dirname + '/src/images',  
  }  
},
```

3. 之后便可以进行通用模块的编辑了，我们首先编辑通用的网络请求模块：

```
'use strict';  
  
var _patent = {  
  //网络请求  
  request : function(param){  
    $.ajax({  
      type   : param.method || 'get',  
      url    : param.url || "",  
      dataType : param.type || 'json',  
      data   : param.data || "",  
      success : function(res){  
        //请求成功  
        var _this = this;  
        if (0 === res.status){  
          //登录成功，若param的success是function类型就向服务器返回  
          data和msg对象
```

```

        typeof param.success === 'function' &&
param.success(res.data,res.msg);
    }
    else if(10 === res.status){
        //还没有登录，则执行登录函数
        _this.dologin();
    }
    else if(1 ===res.status){
        //返回错误，向服务器返回错误信息
        typeof param.error === 'function' && param.error(res.msg);
    }
},
error : function(err){
    //请求失败
    typeof param.error === 'function' && param.error(err.statusText);
}

});

},
dologin : function(){
    //执行登录页面的redirect
    window.location.href = './login.html?redirect=' +
encodeURIComponent(window.location.href);
}
};

module.exports = _patent;

```

要注意，最后的exports,一定是复数。

之后便可以require这个\_patent对象,并使用它进行请求操作：

```

_patent.request({
  url : 'http://happymmall.com/product/list.do?keyword=1',
  success: function(res){
    console.log(res);
  },
  error: function(err){
    console.log(err);
  }
});

```

如果使用这样的请求的话会出现跨域错误，所以要使用charles进行Http代理转移，也就是将不跨域的请求：<http://localhost:8088/product/list.do?keyword=1> 进行代理截取转移到 <http://happymmall.com/product/list.do?keyword=1>

## 4. charles的安装及操作

- 先安装指定版本的java

```
echo "deb http://ppa.launchpad.net/webupd8team/java/ubuntu xenial main"
| tee /etc/apt/sources.list.d/webupd8team-java.list

echo "deb-src http://ppa.launchpad.net/webupd8team/java/ubuntu xenial
main" | tee -a /etc/apt/sources.list.d/webupd8team-java.list
apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys
EEA14886

sudo apt-get update
sudo apt-get install oracle-java8-installer
```

- 然后安装charles

```
wget -q -O - https://www.charlesproxy.com/packages/apt/PublicKey | sudo
apt-key add

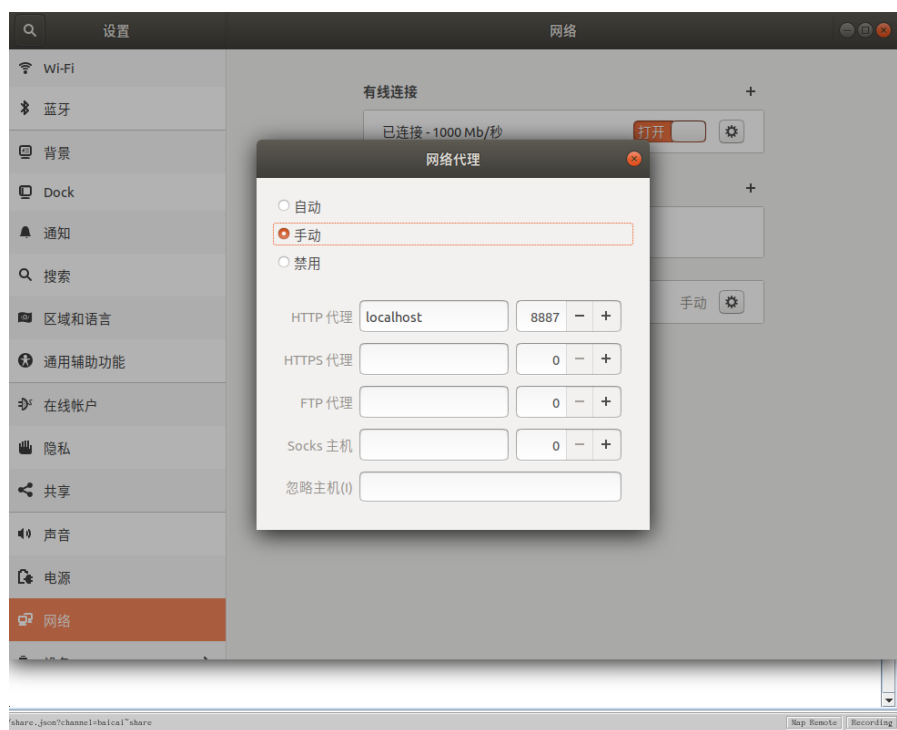
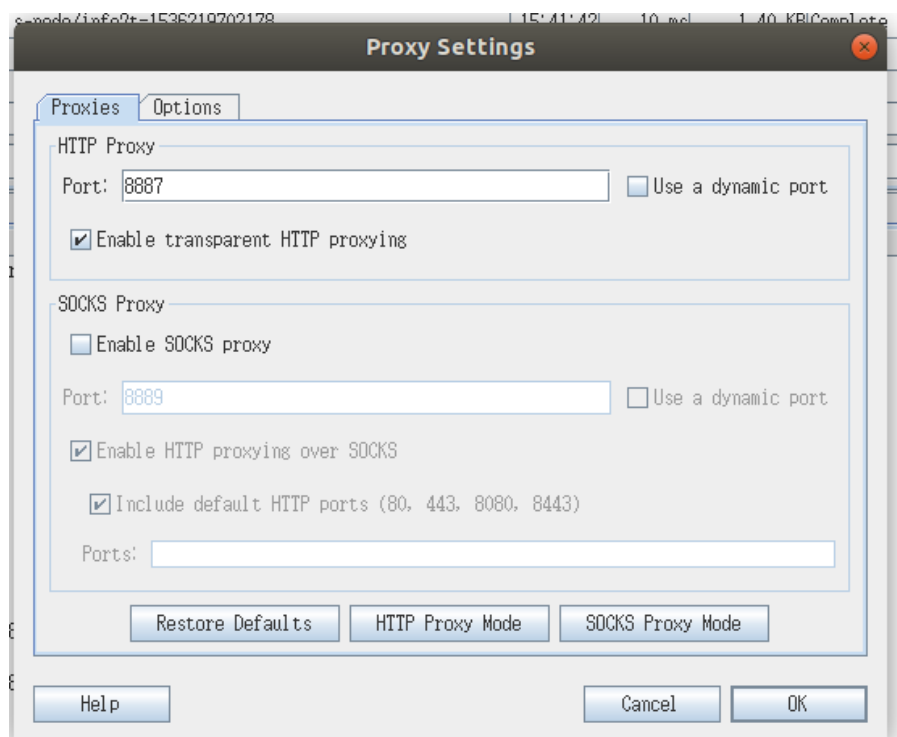
sudo sh -c 'echo deb https://www.charlesproxy.com/packages/apt/ charles-
proxy main > /etc/apt/sources.list.d/charles.list'

sudo apt-get update

sudo apt-get install charles-proxy
```

- 之后执行：

- 之后配置ubuntu的代理使其能监听到本地的http请求



之后就可以看到截取请求了。

- 之后右键->Map Remote

**Edit Mapping**

Map From

Protocol:

Host:

Port:

Path:

Query:

Map To

Protocol:

Host:

Port:

Path:

Query:

☐ Preserve host in header fields

To map from a path and its subdirectories you must end the path with a \*. To map an entire host leave the path blank.

设置好了映射。

## 5. 公用模块-获取服务器的地址

如果想得到真实的地址，我们提供一个获取的方法

```
var conf = {  
  serverHost : "",  
};  
var _patent = {  
  getServerUrl : function(path){  
    return conf.serverHost + path;  
  },  
}
```

## 6. 公用模块-获取url参数

获取url中指定某项的参数值

```

var _patent = {
  // 获取url参数
  getUrlParam : function(name){
    // happymall.com/product/list?keyword=XXX&page=1
    var reg = new RegExp('(' + name + '=[^&]*)([&|$)');
    var result = window.location.search.substr(1).match(reg);
    return result ? decodeURIComponent(result[2]) : null;
  },
}

```

## 7. 公用模块-渲染html模板

渲染模板，就是将得到的html模板与服务器传来的数据相结合生成html。我们使用hogan插件进行模板渲染。

首先安装hogan

```
npm install hogan --save
```

之后在patent.js中实现

```

var Hogan = require('hogan.js');
var _patent = {
  // 渲染html模板
  renderHtml : function(htmlTemplate, data){
    var template = Hogan.compile(htmlTemplate);
    var result = template.render(data);
    return result;
  },
}

```

## 8. 验证功能

从网上寻找相应的正则表达式即可配置：

```

var _patent = {
  // 字段的验证，支持非空，手机，邮箱
  validate : function(value,type){
    var value = $.trim(value);
    // 非空验证
    if ('require' === type){
      return !!value;
    }
    // 手机号验证
    if ('phone' === type){

```

```
        return /^1\d{10}$/.test(value);
    }
    // 邮箱格式验证
    if ('email' === type){
        return /^[a-zA-Z0-9_-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9]{2,6}$/.test(value);
    }
    },
}
```