使用server-mock与跨域问题

使用server-mock

```
$ npm install -g server-mock
```

此为全局安装npm

```
$ mock init
```

在任一位置初始化，生成3个文件

```
$ mock start
```

在命令行里会有回应

```
xmlhttp.readyState ==4 && xmlhttp.status == 200
```

4是第4步，200是网络请求，服务器没问题，有问题是500

```
$ npm i jquery
```

此为安装JQuery

server-mock使用方法：
https://www.npmjs.com/package/server-mock
github地址：
https://github.com/jirengu/server-mock

做网站流程：

（入口、页面的模板、路由的处理、一些静态文件）
1.申请域名、绑定ip
2.购买服务器（比如说阿里云），之后会得到用户名和密码，可以用自己终端登录
3.上传本地代码到服务器（上传方式很多，可以直接复制粘贴也可以git）
（页面模版，若愚建议前端来做，前端对其把控更好）

一个例子：输入后点击按钮获取结果

html:

```
<!DOCTYPE html>
<html>
```

```
<body>
    <p id="container"></p>
    <input type="text" id="music-name">
    <button id="btn">获取</button>
    <script>
        document.querySelector('#btn').addEventListener('click', function() {
                var musicName = document.querySelector('#music-name').value;
                var xhr = new XMLHttpRequest();
                xhr.open('get', '/getMusic?name=' + musicName, true);
                xhr.send();

                xhr.onreadystatechange = function() {
                    if (xhr.readyState === 4 && xhr.status === 200) {
                        document.querySelector('#container').innerText = xhr.re
sponseText;
                        console.log(xhr.responseText)
                    }
            }
        })
    </script>
</body>

</html>
```

router.js:

```
app.get('/getMusic', function(req, res) {
    console.log('ni hao');
    console.log(req.query);
    if (req.query.name === 'hunger') {
        res.send('nihao hunger');
    } else {
        res.send('who are you');
    }
});
```

一个例子：输入大段文字的处理（post）

html:

```
<!DOCTYPE html>
<html>

<body>
    <p id="container"></p>
    <input type="text" id="music-name">
```

```
    <button id="btn">获取</button>

    <textarea id="article" cols="30" rows="10"></textarea>
    <button id="btn2">提交文章</button>
    <script src=https://code.jquery.com/jquery-3.1.1.min.js></script>
    <script>
        $('#btn').addEventListener('click', function() {
            var musicName = document.querySelector('#music-name').value;
            var xhr = new XMLHttpRequest();
            xhr.open('get', '/getMusic?name=' + musicName, true);
            xhr.send();
            xhr.onreadystatechange = function() {
                if (xhr.readyState === 4 && xhr.status === 200) {
                    document.querySelector('#container').innerText = xhr.respon
seText;
                    console.log(xhr.responseText)
                }
            }
        })

        $('#btn2').addEventListener('click', function() {
            var article = $('#article').value;
                      var xhr = new XMLHttpRequest();
                      xhr.open('post', '/postArticle', true);
                      xhr.setRequestHeader("Content-type", "application/x-www-f
orm-urlencoded");

                      xhr.send("article=" + article);
                      xhr.onreadystatechange = function() {
                          if (xhr.readyState === 4 && xhr.status === 200) {
                              $('#container').innerText = xhr.responseText;
                              console.log(xhr.responseText)

            }}
            // ajax({
            //     url: '/postArticle',
            //     type: 'post',
            //     dataType: 'text',
            //     data: {
            //         article: article
            //     },
            //     success: function() {
            //         $('#container').innerText = xhr.responseText;
            //     },
            //     error: function() {
            //         console.log('error');
            //     }
            // })
```

```
        })

        function $(id) {
            return document.querySelector(id);
        }  //转化JQuery对象为DOM对象

        // function ajax(opts) {
        //     opts.success = opts.success || function() {};
        //     opts.error = opts.error || function() {};
        //     opts.type = opts.type || 'get';
        //     opts.dataType = opts.dataType || 'json';
        //     opts.data = opts.data || {};
        //     var dataStr = '';
        //     for (var key in opts.data) {
        //         dataStr += key + '=' + opts.data[key] + '&';
        //     }
        //
        //
        //     var xmlhttp = new XMLHttpRequest();
        //     xmlhttp.onreadystatechange = function() {
        //         if (xmlhttp.readyState === 4 ){
        //                 if (xmlhttp.status === 200) {
        //                     if (opts.dataType === 'text') {
        //                         opts.success(xmlhttp.responseText);
        //                     }
        //                     if (opts.dataType === 'json') {
        //                         var json = JSON.parse(xmlhttp.responseText);
        //                         opts.success(json);
        //                     }
        //                 } else {
        //                     opts.error();
        //                 }
        //             }
        //         };
        //         dataStr = dataStr.substr(0, dataStr.length - 1);
        //         if (opts.type.toLowerCase() === 'post') {
        //             xmlhttp.open(opts.type, opts.url, true);
        //             xmlhttp.setRequestHeader("Content-type", "application/x-
www-form-urlencoded");
        //             xmlhttp.send(dataStr);
        //         }
        //         if (opts.type.toLowerCase() === 'get') {
        //             xmlhttp.open(opts.type, opts.url + '?' + dataStr, true);
        //             xmlhttp.send();
        //         }
        //     }
    </script>
```

```
</body>

</html>
```

router.js

```
app.get('/getMusic', function(req, res) {
    console.log('ni hao');
    console.log(req.query);
    if (req.query.name === 'hunger') {
        res.send('nihao hunger');
    } else {
        res.send('who are you');
    }
});

app.post('/postArticle', function(req, res) {
    res.send(req.body.article.substr(0,10)+'...');
        console.log(req.body)
})
```

一个例子：点击换新闻

html:

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <title>news</title>
    <style>
        .container {
            width: 900px;
            margin: 0 auto;
        }
    </style>
</head>

<body>
    <div class="container">
        <ul class="news">
            <li>11</li>
            <li>22</li>
            <li>33</li>
        </ul>
```

```
        <button class="change">换一组</button>
    </div>



    <script>
        $('.change').addEventListener('click', function() {
            var xhr = new XMLHttpRequest();
            xhr.open('get', '/getNews', true);
            xhr.send();
            xhr.onreadystatechange = function() {
                if (xhr.readyState === 4 && xhr.status === 200) {appendHtml(JSO
N.parse(xhr.responseText))
                    console.log(JSON.parse(xhr.responseText));
                }
            }
        })

        function appendHtml(news) {
            var html = '';
            for (var i = 0; i < news.length; i++) {
                html += '<li>'+news[i]+'</li>';
            }$('.news').innerHTML=html;
        }


        function $(id) {
            return document.querySelector(id);
        }
    </script>
</body>

</html>
```

router.js

```
app.get('/getNews', function(req, res) {

    var news = ["aaa", "bbb", "ccc", "ddd", "eee"]
    var data = [];
    for (var i = 0; i < 3; i++) {
        var index = parseInt(Math.random() * news.length);
        data.push(news[index]);
    }



    res.send(data);
```

```
})
```

调试：前后端3约定

1.url
2.发送请求的参数的格式
3.数据响应的数据格式
如果以上3点都对，那应该是前端这边出错，要检查对数据的处理是否正确。

设置host

C盘 -> Windows -> System32 -> drivers -> etc -> hosts文件;
127.0.0.1 a.jrg.com #注释
127.0.0.1 b.jrg.com #注释

跨域禁止

内里改成

```
xhr.open('get', 'http://a.jrg.com:8080/getNews', true);
```

之后打开`a.jrg.com:8080`可访问；
打开`b.jrg.com:8080`被禁止；

跨域方法1: jsonp

1.本质上，在script里引用JQuery那句就是跨域。
2.原理是避开ajax，变成一般的JS文件引用
html:

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <title>news</title>
    <style>
        .container {
            width: 900px;
            margin: 0 auto;
        }
    </style>
</head>
```

```
<body>
    <div class="container">
        <ul class="news">
            <li>11</li>
            <li>22</li>
            <li>33</li>
        </ul>
        <button class="change">换一组</button>
    </div>



    <script>
        $('.change').addEventListener('click', function() {
            var script = document.createElement('script');
            script.src = 'http://a.jrg.com:8080/getNews?callback=appendHtml';
            document.head.appendChild(script);
            document.head.removeChild(script); //不然太多
        })

        function appendHtml(news) {
            var html = '';
            for (var i = 0; i < news.length; i++) {
                html += '<li>' + news[i] + '</li>';
            }
            console.log(html);
            $('.news').innerHTML = html;
        }


        function $(id) {
            return document.querySelector(id);
        }
    </script>
</body>

</html>
```

router.js

```
app.get('/getNews', function(req, res) {

    var news = ["aaa", "bbb", "ccc", "ddd", "eee"]
    var data = [];
    for (var i = 0; i < 3; i++) {
        var index = parseInt(Math.random() * news.length);
        data.push(news[index]);
```

```
        news.splice(index, 1);
    }
    var cb = req.query.callback;
    if (cb) {
        res.send(cb + '(' + JSON.stringify(data) + ')');
    } else {
        res.send(data); //更加通用
    }


})
```

http://localhost:8080/**可行**
a.jrg.com:8080可行
b.jrg.com:8080可行
本地亦可行

跨域方法2: CORS

原理是，相当于为某个网址颁发<访问许可证>

```
app.get('/getNews', function(req, res) {
    var news = ["aaa", "bbb", "ccc", "ddd", "eee"]
    var data = [];
    for (var i = 0; i < 3; i++) {
        var index = parseInt(Math.random() * news.length);
        data.push(news[index]);
        news.splice(index, 1);
    }
    res.header("Access-Control-Allow-Origin", "http://b.jrg.com:8080");
    // res.header("Access-Control-Allow-Origin", "*");
    res.send(data);
})
```

跨域方法3: 降域

a.html:

```
<html>

<head>
    <style>
        .ct {
            width: 910px;
            margin: auto;
        }
```

```
        .main {
            float: left;
            width: 450px;
            height: 300px;
            border: 1px solid #ccc;
        }

        .main input {
            margin: 20px;
            width: 200px;
        }

        .iframe {
            float: right;
        }

        iframe {
            width: 450px;
            height: 300px;
            border: 1px dashed #ccc;
        }
    </style>

</head>

<body cz-shortcut-listen="true">
    <div class="ct">
        <h1>使用降域实现跨域</h1>
        <div class="main">
            <input type="text" placeholder="http://a.jrg.com:8080/a.html">
        </div>

        <iframe src="http://a.jrg.com:8080/b.html" frameborder="0"></iframe>

    </div>



    <script>
        //URL: http://a.jrg.com:8080/a.html

        document.querySelector('.main input').addEventListener('input', functio
n() {
            console.log(this.value);
            console.log(frames[1]);
            window.frames[0].document.querySelector('input').value = this.value
;
        })
```

```
        // document.domain = "jrg.com"
    </script>



</body>

</html>
```

b.html:

```
<html><head><style>
    html,body{
        margin: 0;
    }
    input{
        margin: 20px;
        width: 200px;
    }
</style>

    </head><body cz-shortcut-listen="true"><input id="input" type="text" placeh
older="http://b.jrg.com:8080/b.html">
<script>

// URL: http://b.jrg.com:8080/b.html

document.querySelector('#input').addEventListener('input', function(){
    window.parent.document.querySelector('input').value = this.value;
})

// document.domain = 'jrg.com';

</script>
</body></html>
```

此时可行。但若a.html改成:

```
<iframe src="http://b.jrg.com:8080/b.html" frameborder="0"></iframe>
```

则不可行，除非激活a.html与b.html里的

```
 document.domain = "jrg.com"
```

跨域方法4: postMessage(window下的方法)

原理是使用一个专用通道Message，有点像CORS

a.html:

```
<html><head><style>
    .ct{
        width: 910px;
        margin: auto;
    }
    .main{
        float: left;
        width: 450px;
        height: 300px;
        border: 1px solid #ccc;
    }
    .main input{
        margin: 20px;
        width: 200px;
    }
    .iframe{
        float: right;
    }

    iframe{
        width: 450px;
        height: 300px;
        border: 1px dashed #ccc;
    }
</style>

</head><body cz-shortcut-listen="true"><div class="ct">
    <h1>使用postMessage实现跨域</h1>
    <div class="main">
        <input type="text" placeholder="http://a.jrg.com:8080/a.html">
    </div>

    <iframe src="http://localhost:8080/b.html" frameborder="0"></iframe>

</div>



<script>
//URL: http://a.jrg.com:8080/a.html

$('.main input').addEventListener('input', function(){
    console.log(this.value);
    window.frames[0].postMessage(this.value,'*');
})
```

```
window.addEventListener('message',function(e) {
        $('.main input').value = e.data
    console.log(e.data);
});




function $(id){
    return document.querySelector(id);
}

</script>
</body></html>
```

b.html:

```
<html><head><style>
    html,body{
        margin: 0;
    }
    input{
        margin: 20px;
        width: 200px;
    }
</style>

    </head><body cz-shortcut-listen="true"><input id="input" type="text" placeh
older="http://b.jrg.com:8080/b.html">
<script>

// URL: http://b.jrg.com:8080/b.html

$('#input').addEventListener('input', function(){
    window.parent.postMessage(this.value, '*');
})

window.addEventListener('message',function(e) {
        $('#input').value = e.data
    console.log(e.data);
});

function $(id){
    return document.querySelector(id);
}

</script>
```

```
</body></html>
```

这个方法一个优点是a.html里的iframe下的网址可以为:

```
http://localhost:8080/b.html
```

就比降域灵活