

## Ajax 封装

**问题：** 发送一次请求代码过多，发送多次就会请求代码冗余重复

**解决方案：** 将请求代码封装到函数中，发请求时调用函数即可

如下：

```
ajax({
  type: 'get',
  url: 'http://www.example.com',
  success: function(data){
    console.log(data)
  }
})
```

### 1. 简单封装

```
function ajax (options) {
  var xhr = new XMLHttpRequest()
  xhr.open(options.type,options.url)
  xhr.send()
  xhr.onreadystatechange = function(){
    if (xhr.readyState==4 && xhr.status==200){
      options.success(xhr.responseText)
    }
  }
}
```

### 2. 请求参数

因为 get 请求与 post 请求传递参数不同，因此我们根据不同的请求方式修改代码

```
function ajax (options) {
  var xhr = new XMLHttpRequest()
  var parmas = ''
  for(var atr in options.data){
    parmas += atr + '=' + options.data[atr] + '&'
  }
  parmas = parmas.substr(0,parmas.length-1) // 去掉最后一个&
  if (options.type == 'get'){
    options.url = options.url + '?' + parmas
  }
  xhr.open(options.type,options.url)
  if (options.type == 'post'){
    xhr.setRequestHeader('Content-Type','application/x-www-form-urlencoded')
  }
  xhr.send(parmas)
```

```

    }else{
        xhr.send()
    }

    xhr.onreadystatechange = function(){
        if (xhr.readyState==4 && xhr.status==200){
            options.success(xhr.responseText)
        }
    }
}

```

### 3. 请求失败处理

```

function ajax (options) {
    var xhr = new XMLHttpRequest()
    var parmas = ''
    for(var atr in options.data){
        parmas += atr + '=' + options.data[atr] + '&'
    }
    parmas = parmas.substr(0,parmas.length-1) // 去掉最后一个&
    if (options.type == 'get'){
        options.url = options.url + '?' + parmas
    }
    xhr.open(options.type,options.url)
    if (options.type == 'post'){
        xhr.setRequestHeader('Content-Type','application/x-www-form-urlencoded')
        xhr.send(parmas)
    }else{
        xhr.send()
    }

    xhr.onreadystatechange = function(){
        if (xhr.readyState==4 && xhr.status==200){
            options.success(xhr.responseText,xhr)
        }else{
            options.error(xhr.responseText,xhr)
        }
    }
}

```

### 4. 返回数据的处理

```

function ajax (options) {
    var xhr = new XMLHttpRequest()
    var parmas = ''
    for(var atr in options.data){
        parmas += atr + '=' + options.data[atr] + '&'
    }
}

```

```

params = params.substr(0,params.length-1) // 去掉最后一个&
if (options.type == 'get'){
    options.url = options.url + '?' + params
}
xhr.open(options.type,options.url)
if (options.type == 'post'){
    xhr.setRequestHeader('Content-Type','application/x-www-form-urlencoded')
    xhr.send(params)
}else{
    xhr.send()
}

xhr.onreadystatechange = function(){
    if (xhr.readyState==4 && xhr.status==200){
        var contentType = xhr.getResponseHeader('Content-Type')
        if (contentType.includes('application/json')){
            respText = JSON.parse(xhr.responseText)
            options.success(respText)
        }else{
            options.success(xhr.responseText)
        }
    }else{
        options.error(xhr.responseText,xhr)
    }
}
}
}

```

## 5. 是否异步

```

function ajax (options) {
    var xhr = new XMLHttpRequest()
    var parmas = ''
    for(var atr in options.data){
        params += atr + '=' + options.data[atr] + '&'
    }
    params = params.substr(0,params.length-1) // 去掉最后一个&
    if (options.type == 'get'){
        options.url = options.url + '?' + params
    }
    xhr.open(options.type,options.url,options.async)
    if (options.type == 'post'){
        xhr.setRequestHeader('Content-Type','application/x-www-form-urlencoded')
        xhr.send(params)
    }else{
        xhr.send()
    }
}

```

```
xhr.onreadystatechange = function(){
    if (xhr.readyState==4 && xhr.status==200){
        var contentType = xhr.getResponseHeader('Content-Type')
        if (contentType.includes('application/json')){
            respText = JSON.parse(xhr.responseText)
            options.success(respText)
        }else{
            options.success(xhr.responseText)
        }
    }else{
        options.error(xhr.responseText,xhr)
    }
}
}
```

## 6. 设置默认

```
function ajax (options) {
    var defaults ={
        type:'get',
        url:'',
        async:true,
        data:{}
        success:function(data){
        },
        error:function(data,xhr){
        }
    }
    Object.assign(defaults,options)
    var xhr = new XMLHttpRequest()
    var parmas = ''
    for(var atr in defaults.data){
        params += atr + '=' + defaults.data[atr] + '&'
    }
    params = params.substr(0,params.length-1) // 去掉最后一个&
    if (defaults.type == 'get'){
        defaults.url = defaults.url + '?' + params
    }
    xhr.open(defaults.type,defaults.url)
    if (defaults.type == 'post'){
        xhr.setRequestHeader('Content-Type','application/x-www-form-urlencoded')
        xhr.send(params)
    }else{
        xhr.send()
    }

    xhr.onreadystatechange = function(){
```

```
if (xhr.readyState==4 && xhr.status==200){  
    var contentType = xhr.getResponseHeader('Content-Type')  
    if (contentType.includes('application/json')){  
        respText = JSON.parse(xhr.responseText)  
        defaults.success(respText)  
    }else{  
        defaults.success(xhr.responseText)  
    }  
    }else{  
        defaults.error(xhr.responseText,xhr)  
    }  
}  
}
```