参考链接：http://www.jianshu.com/p/df464b26ae58

1.ajax请求的方法

```
function dorequest(data,url,successcaalback,failcallback){
    $.ajax({
        data:data,
        url:'http://localhost:3000/'+url,
        type:'GET',
        success:function(result){
            successcaalback(result)
        },
        error:function(result){
            failcallback(result)
        }
    })
}
```

```
function dorequest(data,url,successcaalback,failcallback){
        $.ajax({
            data:data,
            url:'http://localhost:3000/'+url,
            type:'GET',
            success:function(result){
                successcaalback(result)
            },
            error:function(result){
                failcallback(result)
            }
        })
    }
```

2..axios的请求方法

```
function dorequest(data,url,successcaalback,failcallback){
    axios.get('http://localhost:3000/'+url,{params:data})
      .then(function(result){
        successcaalback(result)
      })
      .catch(function(err){
        failcallback(err)
      });
}
function doPost(data,url,successcaalback,failcallback){
    axios.post('http://localhost:3000/'+url,{params:data})
      .then(function(result){
        successcaalback(result)
      })
      .catch(function(err){
        failcallback(err)
      });
}
```

```
function dorequest(data,url,successcaalback,failcallback){
        axios.get('http://localhost:3000/'+url,{params:data})
          .then(function(result){
            successcaalback(result)
          })
          .catch(function(err){
            failcallback(err)
          });
}
    function doPost(data,url,successcaalback,failcallback){
        axios.post('http://localhost:3000/'+url,{params:data})
          .then(function(result){
            successcaalback(result)
          })
          .catch(function(err){
            failcallback(err)
          });
}
```

返回的数据格式类似于

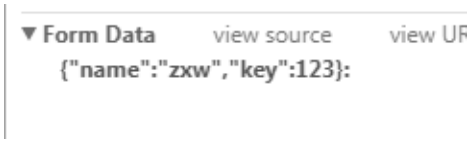
```
axios.get('/user/12345')
  .then(function(res){
    console.log(res.data);
    console.log(res.status);
    console.log(res.statusText);
    console.log(res.headers);
    console.log(res.config);
  })
```

3.axios的post请求

axios 在post时，会遇到参数解析的问题，一般在express4.0中，如果直接写json格式的data，

```
}
function doPost(data,url,successcaalback,failc
    axios.post(host+url, data,{headers : {'Cor
      .then(function(result){
```

会出现参数不好解析的情况，实际传递的参数如下：



服务器接受的参数如下：

```
OPTIONS /login 200 2.027 ms - 4
{ '{"params":{"name":"zxw","key":123}}' : '' }
```

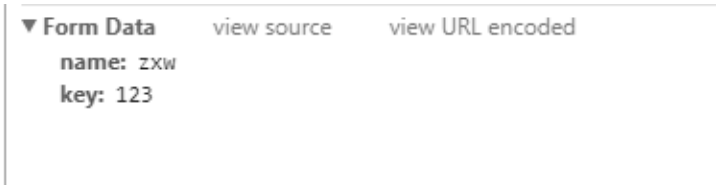
此时，应该使用axios自带的一个qs模块对参数进行解析步骤如下，

1.引入qs，

2，data的格式变为qs.stringify(data)

```
<script>
  var qs = require('qs');
  var host = 'http://localhost:3000/';
  function doPost(data,url,successcaalback,failcallback){
      axios.post(host+url, qs.stringify(data),{headers : {'Content-Type' : 'applicati
        .then(function(result){
          successcaalback(result)
        })
        .catch(function(err){
          failcallback(err)
```

此时浏览器发送的参数格式如下：



服务器端接受的参数如下

```
{ name: 'zxw', key: '123' }
1
```

直接使用req.body.name 获取参数。