|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 注入和读取  数据分类 | 浏览器端注入到request | | | | | 服务器端读取request | | | | 服务器端注入到response | | 浏览区端读取response |
|  | | **注入方式** | **跨域请求** | **请求方式** | **注入到request对象 ctx.request ctx.request.xxx都返回对象类型** | | **获取方式** | **Koa获取数据所需要中间件** | **读取request对象 ctx.request ctx.request.xxx都返回对象类型** | **Koa注入body对象到response  注: ctx.response.body=== ctx.body===ctx.state === 浏览器端response.data  ctx.response.body可以返回任何类型** | | Axios读取response |
| **Koa原生** | **+Response中间件** |
| headers数据 | | Ajax | √ | 5种 | **头文件:**  头文件携带自定义数据,用于后端做程序员做判断 可跨域? 类型改? 已登录?  第一种附加头文件:扩展axios  axios.defaults.headers.common['Authorization'] = 'AUTH\_TOKEN'; axios.defaults.headers.post['Content-Type']='application/x-www-form-urlencoded';  第二种附加头文件:拦截器  axios.interceptors.request() | | 5种 | 自带 | ctx.request.header //等于ctx.req.headers | ctx.response.body = {goods:’data’,   code: 0,  msg:’msg’ } ctx.response.status=201 | ctx.state.code=0  ctx.state.msg=’msg’  ctx.state.goods=[]  ctx.status=201 | .then(response=>{response.data.msg}) |
| query[英]问号,字符串数据 | | Ajax | √ | Get (默认) | axios.get(‘/a’,{params:{id:1}}) | | router.get(‘/a’,midware) | 自带 | ctx.request.query |
| axios.get('/a?name=小李') //字符串不带引号,若带引号会错误连引号一起搜索 | |
| Json数据 | | ajax | √ | Post | axios.post(‘/a’, {a:1}） //不可传送数组,例如axios.post(‘/a’, [1,2,3]) ,因为ctx.request.body必须是对象类型的 | | router.post(‘/a’,midware) | koa-bodyparser | ctx.request.body  //这里不能省去request |
| delete | axios.delete('http://127.0.0.1:3000/api/request/', {data: {id: 1}}) | | router.del(‘/a’,midware) |
| put | axios.put('http://localhost:3000/api/request',{a:1}) | | router.put(‘/a’,midware) |
| patch | axios.patch('http://localhost:3000/api/request',{a:1}) | | router.patch(‘/a’,midware) |
| 表单对象FormData数据 | | ajax | √ | 5种 | 第三种附加头文件:  axios.post(‘/a’,{img:Filetype(….)},{ headers:{ ‘Content-Type’:‘multipart/form-data’}}) | | 5种 | koa-body | ctx.request.files |
| Xml数据 | | ajax | √ | 5种 | axios.post(‘/a’, xmlString,{ headers: {'Content-Type': 'text/xml'}}） | | 5种 | koa-xml-body(在前) koa-bodyparser和koa-body冲突 | ctx.request.body |
| params 字符串数据 | | ajax | √ | 5种 | axios.post(`/delCategory/${row.\_id}`) | | 5种 | 自带 | ctx.params  //params不是request的key,不是简写,也返回对象 |
| Html 请求 | | 打开页面 localhost:3000/api | X | get | 默认携带头文件,头文件里有cookie | | Get | 自带 | ctx.request.header.cookie (只有此模式下可以获得cookie) | 只有此模式下可以直接后端写入cookie      //1.1后端设置ｃｏｏｋｉｅ:此方法仅用于ｈｔｔｐ请求localhost:3000/api时,后端向浏览器cookie设置toekn      ctx.cookies.set(        'cid',        'token\_code\_xxx',        {          domain: 'localhost',  // 写cookie所在的域名          path: '/',       // 写cookie所在的路径          maxAge: 10 \* 60 \* 1000, // cookie有效时长          expires: new Date('2017-02-15'),  // cookie失效时间          httpOnly: false,  // 是否只用于http请求中获取          overwrite: false  // 是否允许重写        }      )      //1.2 koa版后端获取cookie此方法仅用于获取ｈｔｔｐ请求localhost:3000/api时获取request.cookie里的token      console.log(`ctx.cookies.get('username')------`)      console.log(ctx.cookies.get('username')) |  |
| Html提交 | | Html-action(重绘) | X | Post | <form action="http://localhost:3000/api/request-file/" method="post" enctype="multipart/form-data">  </form> | | post | koa-body | ctx.request.body |  |  |

|  |  |
| --- | --- |
| **request** | **response** |
|  |  |