



黑马程序员™
www.itheima.com

传智播客旗下
高端IT教育品牌

jQuery 中的 Ajax

目录 Contents

- ◆ \$.ajax()
- ◆ \$.get、\$.post
- ◆ Todo案例
- ◆ 全局事件
- ◆ nprogress进度条插件

\$.ajax()方法概述

作用：发送Ajax请求。

```
$.ajax({  
    type: 'get',  
    url: 'http://www.example.com',  
    data: { name: 'zhangsan', age: '20' },  
    contentType: 'application/x-www-form-urlencoded',  
    beforeSend: function () {  
        return false  
    },  
    success: function (response) {},  
    error: function (xhr) {}  
});
```

```
{  
    data: 'name=zhangsan&age=20'  
}
```

```
{  
    contentType: 'application/json'  
}
```

```
JSON.stringify({name: 'zhangsan', age:  
    '20'})
```

\$.ajax()方法概述

作用：发送jsonp请求。

```
$.ajax({  
    url: 'http://www.example.com',  
    // 指定当前发送jsonp请求  
    dataType: 'jsonp',  
    // 修改callback参数名称  
    jsonp: 'cb',  
    // 指定函数名称  
    jsonCallback: 'fnName',  
    success: function (response) {}  
})
```

serialize方法

作用：将表单中的数据自动拼接成字符串类型的参数

```
var params = $('#form').serialize();  
// name=zhangsan&age=30
```

■ \$.get()、\$.post

\$.get()、\$.post()方法概述

作用：\$.get方法用于发送get请求，\$.post方法用于发送post请求。

```
$.get('http://www.example.com', {name: 'zhangsan', age: 30}, function (response) {})  
$.post('http://www.example.com', {name: 'lisi', age: 22}, function (response) {})
```

为todo数据库添加账号

1. 使用mongo命令进入mongodb数据库
2. 使用use admin命令进入到admin数据中
3. 使用db.auth('root' , 'root')命令登录数据库
4. 使用use todo命令切换到todo数据库
5. 使用db.createUser({user: 'itcast' , pwd: 'itcast' , roles: ['readWrite']})创建todo数据库账号
6. 使用exit命令退出mongodo数据库

展示任务列表

1. 准备一个放置任务列表的数组
2. 向服务器端发送请求，获取已存在的任务
3. 将已存在的任务存储在任务列表数组中
4. 通过模板引擎将任务列表数组中的任务显示在页面中

添加任务

1. 为文本框绑定键盘抬起事件，在事件处理函数中判断当前用户敲击的是否是回车键
2. 当用户敲击回车键的时候，判断用户在文本框中是否输入了任务名称
3. 向服务器端发送请求，将用户输入的任务名称添加到数据库中，同时将任务添加到任务数组中
4. 通过模板引擎将任务列表数组中的任务显示在页面中

删除任务

1. 为删除按钮添加点击事件
2. 在事件处理函数中获取到要删任务的id
3. 向服务器端发送请求，根据ID删除任务，同时将任务数组中的相同任务删除
4. 通过模板引擎将任务列表数组中的任务重新显示在页面中

更改任务状态

1. 为任务复选框添加onchange事件
2. 在事件处理函数中获取复选框是否选中
3. 向服务器端发送请求，将当前复选框的是否选中状态提交到服务器端
4. 将任务状态同时也更新到任务列表数组中
5. 通过模板引擎将任务列表数组中的任务重新显示在页面中并且根据任务是否完成为li元素添加completed类名

修改任务名称

1. 为任务名称外层的label标签添加双击事件，同时为当前任务外层的li标签添加editing类名，开启编辑状态
2. 将任务名称显示在文本框中并让文本框获取焦点
3. 当文本框离开焦点时，将用户在文本框中输入值提交到服务器端，并且将最新的任务名称更新到任务列表数组中
4. 使用模板引擎重新渲染页面中的任务列表。

计算未完成任务数量

1. 准备一个用于存储未完成任务数量的变量
2. 将未完成任务从任务数组中过滤出来
3. 将过滤结果数组的长度赋值给任务数量变量
4. 将结果更新到页面中

显示未完成任务

1. 为active按钮添加点击事件
2. 从任务列表数组中将未完成任务过滤出来
3. 使用模板引擎将过滤结果显示在页面中

清除已完成任务

1. 为clear completed按钮添加点击事件
2. 向服务器端发送请求将数据库中的已完成任务删除掉
3. 将任务列表中的已完成任务删除调用
4. 使用模板引擎将任务列表中的最后结果显示在页面中

全局事件

只要页面中有Ajax请求被发送，对应的全局事件就会被触发

```
.ajaxStart()    // 当请求开始发送时触发  
.ajaxComplete() // 当请求完成时触发
```


NProgress

官宣：纳米级进度条，使用逼真的滑动动画来告诉用户正在发生的事情！

```
<link rel='stylesheet' href='nprogress.css' />  
<script src='nprogress.js'></script>
```

```
NProgress.start(); // 进度条开始运动  
NProgress.done(); // 进度条结束运动
```



传智播客旗下高端IT教育品牌