**JavaScript 实例**

## JavaScript 读取文件

步骤：

1.在html中使用input元素获取目标文件路径。

将input元素中的type属性值为file；

<input id="filePicker" type="file" />

当使用元素选取了文件时：

a.其value属性由" "变为文件在本机中的路径

b.其files属性为为上传文件的集合，可使用[序号]指定文件： fp.files[0]

2.在js中使用FileReader()读取文件。

var reader = new FileReader(); //创建FileReader对象实例

reader.readAsText(fp.files[0], "utf-8"); //将file对象根据编码格式转化为字符串形式,默认格式为“utf-8”

reader.onload = function () { //FileReader实例下的onload事件将处理转换后的结果，此处使用匿名函数处理读取完成后的代码

var data = reader.result; //文件的内容将被读取在FileReader对象的result属性中

p.innerText = data; //最后对读取的文件内容进行处理

}

## Ajax+Django+JavaScript

使用$.ajax()函数

|  |  |
| --- | --- |
| Name | Value/Description |
| async | A Boolean value indicating whether the request should be handled asynchronous or not. Default is true |
| beforeSend(*xhr*) | A function to run before the request is sent |
| cache | A Boolean value indicating whether the browser should cache the requested pages. Default is true |
| complete(*xhr,status*) | A function to run when the request is finished (after success and error functions) |
| contentType | The content type used when sending data to the server. Default is: "application/x-www-form-urlencoded" |
| context | Specifies the "this" value for all AJAX related callback functions |
| data | Specifies data to be sent to the server |
| dataFilter(*data*,*type*) | A function used to handle the raw response data of the XMLHttpRequest |
| dataType | The data type expected of the server response. |
| error(*xhr,status,error*) | A function to run if the request fails. |
| global | A Boolean value specifying whether or not to trigger global AJAX event handles for the request. Default is true |
| ifModified | A Boolean value specifying whether a request is only successful if the response has changed since the last request. Default is: false. |
| jsonp | A string overriding the callback function in a jsonp request |
| jsonpCallback | Specifies a name for the callback function in a jsonp request |
| password | Specifies a password to be used in an HTTP access authentication request. |
| processData | A Boolean value specifying whether or not data sent with the request should be transformed into a query string. Default is true |
| scriptCharset | Specifies the charset for the request |
| success(*result,status,xhr*) | A function to be run when the request succeeds |
| timeout | The local timeout (in milliseconds) for the request |
| traditional | A Boolean value specifying whether or not to use the traditional style of param serialization |
| type | Specifies the type of request. (GET or POST) |
| url | Specifies the URL to send the request to. Default is the current page |
| username | Specifies a username to be used in an HTTP access authentication request |
| xhr | A function used for creating the XMLHttpRequest object |

在html页面上的JS代码：

$("#name").change(function () { //指定元素值变化时

var name = $(this).val(); //获取元素的值

$.ajax({

url: '', //异步请求的url

data: { 'name': name }, //提交的数据名和值

dataType: 'json', //数据类型

success: function (data) { //当请求成功后执行的代码，data参数为返回的数据名

if (data.is\_taken) {

alert(data.error\_message);

}

}

});

});

Django的Views.py

def member\_validate(request):

name = request.GET.get('name')

data = {'is\_taken':Member.objects.filter(name\_\_exact=name).exists() }

if data['is\_taken']: #判断检验结果是否为真

data['error\_message'] = 'A user with this username already exists!'

return JsonResponse(data)

$.ajax({

url: '{% url 'ajax\_tb\_list' %}, //异步请求的url

dataType: 'json', //数据类型

success: function (data) { //当请求成功后执行的代码

alert(data.db\_path);

}

});