1. [**使用ajax提交form表单，包括ajax文件上传**](https://www.cnblogs.com/zhuxiaojie/p/4783939.html)
2. **前言**

使用ajax请求数据，很多人都会，比如说：

$.post(path,{data:data},function(data){

...

},"json");

又或者是这样的ajax

[复制代码](javascript:void(0);)

$.ajax({

url:"${pageContext.request.contextPath}/public/testupload",

type:"post",

data:{username:username},

success:function(data){

window.clearInterval(timer);

console.log("over..");

},

error:function(e){

alert("错误！！");

window.clearInterval(timer);

}

});

[复制代码](javascript:void(0);)

同样的，很多人也会。但是写的越多就越会发现，这样子虽然可以避免刷新页面，但是我们要写大量的js来到得数据：

var username = $("#username").val();

var password = $("#password").val();

...

如果数量少的话，那还没有什么，但是如果数据十分大的话，那就十分的麻烦，那有没有什么简单的方法呢？答案肯定有的！下面介绍两种方法，可以极大的提高开发人员的效率。

1. **方法**
2. **方法一：使用FormData对象**

FormData对象是html5的一个对象，目前的一些主流的浏览器都已经兼容。额，如果你说ie8什么的，那我们还是来谈谈今天的天气吧，我没听见。呵呵，开个玩笑，不支持FormData的，可以使用方法二，下面会介绍。接着说FormData，它是一个html5的javascript对象，非常的强大。

FormData可以凭空创建一个对象，然后往这个对象里面添加数据，然后直接提交，不需要写一行html代码，如下：

[复制代码](javascript:void(0);)

var form = new FormData();

form.append("username","zxj");

form.append("password",123456);

var req = new XMLHttpRequest();

req.open("post", "${pageContext.request.contextPath}/public/testupload", false);

req.send(form);

[复制代码](javascript:void(0);)

这样就可以向浏览器发送表单数据了，或者也可以使用Jquery这样发送：

[复制代码](javascript:void(0);)

var form = new FormData();

form.append("username","zxj");

form.append("password",123456);

　$.ajax({

url:"${pageContext.request.contextPath}/public/testupload",

type:"post",

data:form,

processData:false,

contentType:false,

success:function(data){

window.clearInterval(timer);

console.log("over..");

}

});

[复制代码](javascript:void(0);)

这样也可以直接发送数据到后台。

**你以为这就完了？不！这才刚开始呢！！**

其次FormData还支持直接从html中的表单生成数据，就是在html页面中已经有数据了，然后FormData可以直接把这个表单的数据写入这个对象，然后直接提交给后台

代码如下，先给出html代码:

[复制代码](javascript:void(0);)

<form id="tf">

<input type="file" name="img"/>

<input type="text" name="username"/>

<input type="button" value="提" onclick="test();"/>

.............

</form>

[复制代码](javascript:void(0);)

大家注意到没有，里面可是有文件的哦！

没错，FormData还支持困扰众多开发者已久的ajax的上传文件，以前我们上传文件，需要写一个表单直接刷新提交，但是这里不需要，下面给出提交代码：

[复制代码](javascript:void(0);)

function test(){

var form = new FormData(document.getElementById("tf"));

// var req = new XMLHttpRequest();

// req.open("post", "${pageContext.request.contextPath}/public/testupload", false);

// req.send(form);

$.ajax({

url:"${pageContext.request.contextPath}/public/testupload",

type:"post",

data:form,

processData:false,

contentType:false,

success:function(data){

window.clearInterval(timer);

console.log("over..");

},

error:function(e){

alert("错误！！");

window.clearInterval(timer);

}

});

get();//此处为上传文件的进度条

}

[复制代码](javascript:void(0);)

就是这么简单，使用FormData，在构造这个对象的时候，把表单的对象，作为一个参数放进去，就可以了，然后FormData，就会得到这个表单对象里面的所有的参数，甚至我们在表单中，都不需要声明enctype ="multipart/form-data" ，就可以直接提交。

使用FormData，第一是在提交表单的时候，不需要写大量的js来获得表单数据，直接把表单对象构造就行了。第二就是可以直接异步上传文件，简单牛逼爆了！

**注意：使用FormData提交的时候，大家会注意到表单提交的是request payload，具体有兴趣的同学可以自己百度，它不是之前的Form data提交的，所以后台也是要经过处理的，比如springMVC就需要配置**

<!-- 配置nultipartresolver,注意：id名必须这样写，不然会报错 -->

<bean id="multipartResolver" class="org.springframework.web.multipart.commons.CommonsMultipartResolver">

<property name="defaultEncoding" value="UTF-8"></property>

<property name="maxInMemorySize" value="10240000"></property>

</bean>

不然会接收不到数据，当然，后台的话，我们这里就先不管。

https://www.cnblogs.com/zhuxiaojie/p/4783939.html