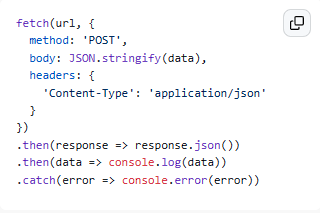
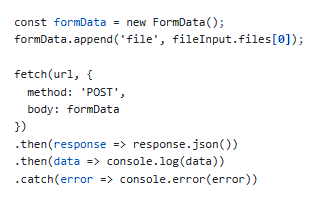
用fetch传递post请求：



其中，url是请求的地址，data是要发送的数据，headers设置了请求头的Content-Type为JSON格式。

要使用Fetch传递FormData文件，可以像下面这样编写代码：



其中，formData是一个FormData对象，可以使用append()方法将文件添加到FormData中。在Fetch请求中，将method设置为POST，body设置为FormData对象即可。

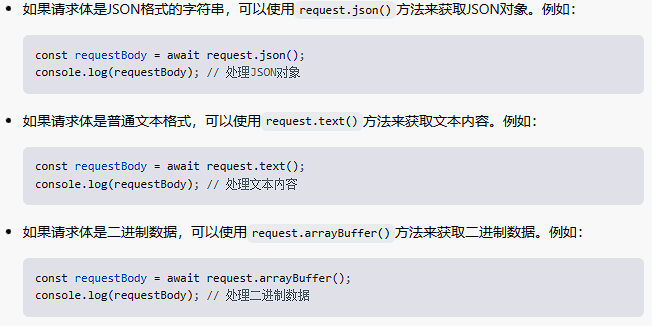
需要注意的是，如果要上传多个文件，可以使用formData.append()方法多次添加文件到FormData中。同时，不能再代码中设置请求头的类型为’multipart/form-data’，因为这样会失去浏览器为它自动生成的随机分隔符boundary，这样会造成后端接受不到文件的报错。

在Service Worker中拦截Fetch请求并获取请求体里的内容，可以使用Fetch Event和Request对象的clone()方法。代码示例：



我们使用Fetch Event监听所有的Fetch请求，然后使用handleRequest()函数处理请求。在handleRequest()函数中，我们首先使用clone()方法克隆请求对象，然后使用formData()方法获取请求体内容。接下来，我们可以对请求体内容进行处理，并返回响应。

需要注意的是，如果请求体内容不是FormData对象，可以使用text()、json()等方法获取请求体内容。另外，如果请求方法不是POST或PUT等带有请求体的方法，那么请求体内容为空。



如果 POST 请求中传递的是 FormData 文件，你可以使用 FormData API 读取请求主体内容。以下是一个示例代码：



要获得 File 对象中的内容，可以使用 File 对象提供的 FileReader API。以下是使用 FileReader 读取 File 对象内容的示例代码：

