**[利用iframe+from表单实现跨域上传文件](https://www.cnblogs.com/shanligang/p/5789459.html)**

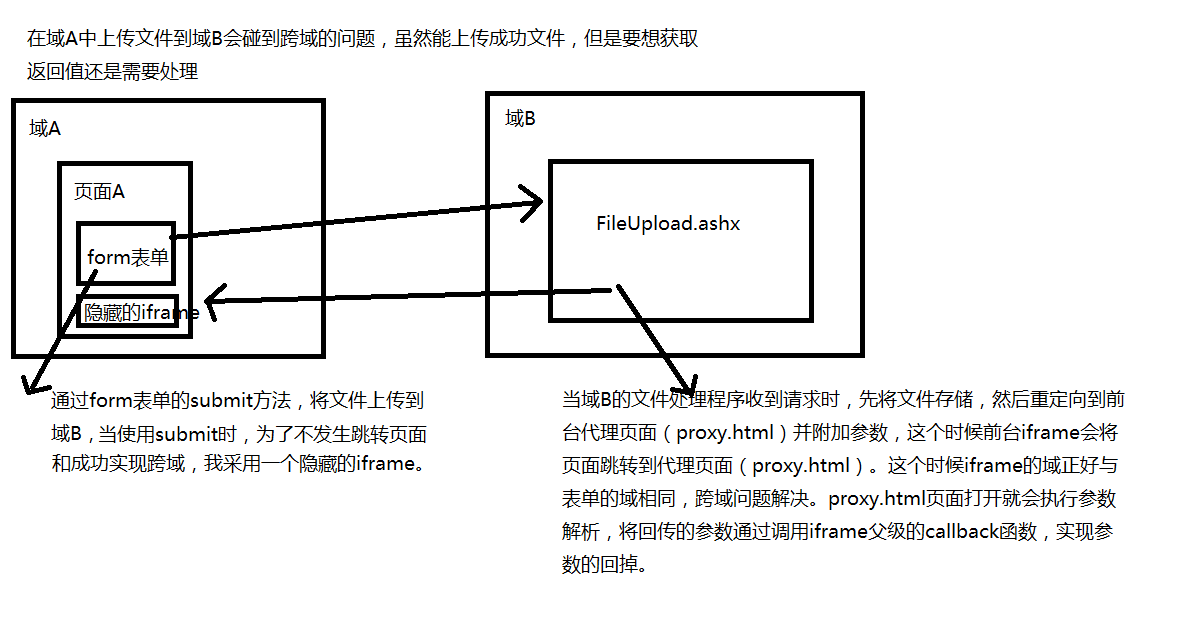
一、需要的材料

客户端需要一个HTML页面A其中包含一个iframe和一个form表单，一个页面B（我称之为客户端代理）里面包含对返回参数的处理；

服务端需要一个asp.net的一般处理程序用来处理上传文件并返回值。

二、原理图

有图才有真相，哈哈哈



三、客户端代码实现

1、页面A的实现

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31 | <!DOCTYPE html>  <html>        <head>          <meta charset="UTF-8">          <title>A页面</title>          <script src="http://libs.baidu.com/jquery/2.0.0/jquery.min.js"></script>      </head>        <body>          <iframe name='hidden\_frame' id="hidden\_frame" style='display:none'></iframe>          <form id="fileform" method="post" enctype="multipart/form-data" target="hidden\_frame">              <input type="file" name="fileUpload" />          </form>          <button id="submitbtn">开始上传</button>          <script type="text/javascript">              function callback(msg) {                  //回调函数                  alert(msg);              }          </script>          <script type="text/javascript">              $("#submitbtn").click(function() {                  var callbackurl = window.location.href.substring(0, window.location.href.lastIndexOf('/')) + "proxy.html"; //获取代理文件路径                  $("#fileform").attr("action", "FileHandler.ashx?callbackurl=" + callbackurl);                  $("#fileform").submit();              })          </script>      </body>    </html> |

2.代理页面实现 proxy.html 为了方便调用，我将该页面放在了与A页面同一目录下，也可以不同目录，但必须是同域

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | <!DOCTYPE html>  <html>        <head>          <meta charset="UTF-8">          <title>代理文件</title>      </head>      <body>          <script type="text/javascript">              var rs = window.location.search.split('?').slice(1);              window.parent.callback(rs.toString().split('=').slice(1));//调用父页面的方法          </script>      </body>  </html> |

三、服务端实现

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | public void ProcessRequest(HttpContext context)  {      context.Response.ContentType = "text/html";      HttpServerUtility server = context.Server;      HttpRequest request = context.Request;      HttpResponse response = context.Response;      string callbackurl = context.Request["callbackurl"];      HttpPostedFile file = context.Request.Files[0];      if (file.ContentLength > 0)      {          string extName = Path.GetExtension(file.FileName);          string fileName = Guid.NewGuid().ToString();          string fullName = fileName + extName;            string imageFilter = ".jpg|.png|.gif|.ico";// 随便模拟几个图片类型          if (imageFilter.Contains(extName.ToLower()))          {              string phyFilePath = server.MapPath("~/Upload/Image/") + fullName;              file.SaveAs(phyFilePath);              context.Response.Redirect(callbackurl + "?msg='上传成功'")          }      }  } |

四、该方法的优缺点以及适用范围

优点：没有兼容性问题，在常见的浏览器中都是适用的；

缺点：返回数据最大支持2KB，对于较大的数据范围建议使用CORS方式跨域

适用范围：上传文件，返回值只是一些短信息比如返回上传正确与否。

分类: [JavaScript](http://www.cnblogs.com/shanligang/category/602837.html)