

# 学生选课功能接口说明：

调用逻辑：A系统服务器post请求

```
@RequestMapping(value="/getStudent")  
public static String chooseClass(HttpServletRequest request, HttpServletResponse response)
```

调用示例：

```
String url="http://127.0.0.1:8081/course/getStudent?  
fileName=studentA.xml&type1=C&type2=B";  
String filename="src/main/resources/static/Cxml/studentC.xml";  
upFile(url,filename);
```

其中 filename是文件名，方便我这边存储临时文件。

type1是选课系统的类型，type2是被选课系统的类型

upFile这个方法我贴到下面：

```
public static void upFile(String urlS, String filename) throws IOException {  
    DataInputStream in = null;  
    OutputStream out = null;  
    HttpURLConnection conn = null;  
    JSONObject responseTxt = null;  
    InputStream ins = null;  
    ByteArrayOutputStream outputStream = null;  
    try {  
        //"http://localhost:8081/mes-boot-doc/test/fileupload?fileName=shafei.xls"  
        URL url = new URL(urlS);  
  
        conn = (HttpURLConnection) url.openConnection();  
        // 发送POST请求必须设置如下两行  
        conn.setDoOutput(true);  
        conn.setUseCaches(false);  
        conn.setRequestMethod("POST");  
        conn.setRequestProperty("Content-Type", "text/html");  
        conn.setRequestProperty("Cache-Control", "no-cache");  
        conn.setRequestProperty("Charset", "UTF-8");  
        conn.connect();  
        conn.setConnectTimeout(10000);  
        out = conn.getOutputStream();  
  
        File file = new File(filename);  
        in = new DataInputStream(new FileInputStream(file));  
  
        int bytes = 0;  
        byte[] buffer = new byte[1024];  
        while ((bytes = in.read(buffer)) != -1) {  
            out.write(buffer, 0, bytes);  
        }  
        out.flush();  
  
        // 返回流
```

```

        if (conn.getResponseCode() == HttpURLConnection.HTTP_OK) {
            System.out.println(conn.getResponseCode());
            ins = conn.getInputStream();
            outputStream = new ByteArrayOutputStream();
            byte[] data = new byte[1024];
            int count = -1;
            while ((count = ins.read(data, 0, 1024)) != -1) {
                outputStream.write(data, 0, count);
            }
            data = null;
            responseTxt = JSONObject.parseObject(new String(outputStream
                .toByteArray(), "UTF-8"));
        }
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        if (in != null) {
            in.close();
        }
        if (out != null) {
            out.close();
        }
        if (ins != null) {
            ins.close();
        }
        if (outStream != null) {
            outStream.close();
        }
        if (conn != null) {
            conn.disconnect();
        }
    }
}
}

```

以上是A系统要做的事情

在集成服务器chooseClass方法被A系统调用后，chooseClass完成对传来的xml文件的验证和转换为B系统的xml格式

集成端chooseClass调用B服务器的一个方法，传去这个Bxml还有原系统的类型

```

String chooseurl=url2+"/course/receiveAstudentAndReponse";
postFileto2(chooseurl,fileType2);

```

这是在chooseClass最后调用的，调用B服务器对应的receiveAstudentAndResponse方法，这个方法是需要你写的，就是接受一个学生xml，然后插入看看是否成功。尝试插入后自己那边发个get请求给集成服务器，调用集成服务器

```

@RequestMapping(value="/sendchooseResult")
public static String chooseResult(HttpServletRequest request, HttpServletResponse response) {
    String url="http://112.74.24.222:11222/";
}

```

这个方法调用格式是：在get的url里添加? result=success

如果成功就是result=success 失败就是result=failure

然后我这个集成服务器收到你这个get请求，向A服务器反馈选课结果这里你也要写一个接受get请求的方法让我调用，格式和B服务器向集成服务器反馈一样，在url里接受result参数，告诉我接口标识就行。

```
String chooseresult=request.getParameter( s: "result");
String system=request.getParameter( s: "type");
```

这样就可以获得选课是否成功的信息，然后这个方法再发贻给前端。

## 请求分享课程功能说明

A系统调用

```
@RequestMapping(value="share")
public static String share(HttpServletRequest request, HttpServletResponse response)
```

方法，在url参数type里表明自己系统的类型

然后集成服务器share方法开始运作

调用

```
public static void requestforshare(String urlS) throws IOException
{
    URL url = new URL(urlS);
    HttpURLConnection conn = null;
    conn = (HttpURLConnection) url.openConnection();
    conn.connect();
}
```

这个方法向某个服务器get请求课程分享，这个方法调用B系统的某个方法，这是B系统需要实现的方法

在B系统的这个方法里，要取出自己课程xml，然后post调用集成服务器的

```
@RequestMapping(value="receiveShare")
public static void receiveShareFileandSend(HttpServletRequest request, HttpServletResponse response)
```

方法，这个方法的调用格式是post 在url里加入type1，type2参数表明自己系统类型，和xml目标系统的类型。加入fileName参数方便集成服务器存储。

在这个方法中，集成服务器接受来自B系统的xml，验证转换后向A系统发送（从b系统type2参数可以知道往哪个系统发送），发送也是通过post，A系统作为原来请求BC系统share的源，就收到了来自集成服务器的B系统share，收到后展示到前端，然后如果C系统也通过集成服务器发了，也进行展示。也即是说share请求是只想集成服务器请求一次，但接受是响应两次。