

edoc2 SDK 开发手册

文档版本信息:

时间	版本	操作者	备注
2013-10-24	V1.0	安道金	

目 录

1	edoc2 API 接口	3
1.1	文件操作	3
1.2	文件夹操作	15
1.3	文件和文件夹混合	22
1.4	共享操作	23
1.5	文档权限	24
1.6	元数据操作(仅适用 ECM)	31
1.7	搜索	42
1.8	组织模块	44
1.9	日志操作	58
1.10	知识地图操作(仅适用 ECM)	59
1.11	系统管理操作	60
2	edoc2 业务组件(仅适用 ECM)	64
2.1	组织选择	64
2.2	文档选择	66
2.3	文件夹选择	66
2.4	文件上传	67
2.5	文件下载	68
2.6	文件预览	69
2.7	搜索接口	70
2.8	文件选择	71
2.9	文件夹选择	73
2.10	edoc2 主界面操作说明	75
3	edoc2 组织集成(仅适用 ECM)	83
3.1	组织集成配置文件修改	83
3.2	SqlConfig-template.xml 文件 sql 配置项说明	85
4	edoc2 SDK 集成工具介绍	99
4.1	edoc2SdkDeployTool 配置文件布置工具	99
5	附录	100
5.1	API 配置	100
5.2	API Demo(仅适用 ECM)	103
5.3	Java 调用	104

1 edoc2 API 接口

1.1 文件操作

1.1.1 上传文件

1.1.1.1 创建文件

接口形式	代理类
代理类	EDoc2.ApiClientManage.Upload.UploadFileApiProvider
方法	<code>public virtual ResultInt CreateFile(string token, int atFolderId, string uploadFileName, byte[] fileContent)</code>
参数说明	Token:用户凭证 atFolderId:文件夹编号 uploadFileName:上传的文件名 fileContent:文件二进制内容
返回值	Result=0 成功,ResultValue:文件编号
版本	4.7.0 以上、LiteV2

接口形式	代理类
代理类	EDoc2.ApiClientManage.Upload.UploadFileApiProvider
方法	<code>public virtual ResultInt CreateFile(string token, int atFolderId, string uploadFileName, Stream stream)</code>
参数说明	Token:用户凭证 atFolderId:文件夹编号 uploadFileName:上传的文件名 stream:文件流
返回值	Result=0 成功,ResultValue:文件编号
版本	4.7.0 以上、LiteV2

接口形式	代理类
代理类	EDoc2.ApiClientManage.Upload.UploadFileApiProvider
方法	<code>public virtual ResultInt CreateFile(string token, int atFolderId, string uploadFileName, long readSize, Stream stream, UploadFileBlockProcess blockHandler, UploadFileErrorProcess errorHandler)</code>
参数说明	Token:用户凭证 atFolderId:文件夹编号 uploadFileName:上传的文件名 readSize:一次性读取大小(吞吐数据量) stream:文件流 blockHandler:上传处理方法 errorHandler:错误处理方法
返回值	Result=0 成功,ResultValue:文件编号
版本	4.7.0 以上、LiteV2

1.1.1.2 更新文件

接口形式	代理类
代理类	EDoc2.ApiClientManage.Upload.UploadFileApiProvider
方法	publicvirtualint UpdateFile(string token, int fileId, byte[] fileContent)
参数说明	Token:用户凭证 fileId:文件编号, fileContent:文件内容
返回值	Result=0 成功,ResultValue:文件编号
版本	4.7.0 以上、 LiteV2

接口形式	代理类
代理类	EDoc2.ApiClientManage.Upload.UploadFileApiProvider
方法	publicvirtualint UpdateFile(string token, int fileId, Stream stream)
参数说明	Token:用户凭证 fileId:文件编号 stream:文件流
返回值	Result=0 成功,ResultValue:文件编号
版本	4.7.0 以上、 LiteV2

接口形式	代理类
代理类	EDoc2.ApiClientManage.Upload.UploadFileApiProvider
方法	publicvirtualint UpdateFile(string token, int fileId, long readSize, Stream stream, UploadFileBlockProcess blockHandler, UploadFileErrorProcess errorHandler)
参数说明	Token:用户凭证 atFolderId:文件夹编号 uploadFileName:上传的文件名 readSize:一次性读取大小(吞吐数据量) stream:文件流 blockHandler:上传处理方法 errorHandler:错误处理方法
返回值	Result=0 成功,ResultValue:文件编号
版本	4.7.0 以上、 LiteV2

1.1.1.3 根据文件路径创建或更新文件

接口形式	代理类
代理类	EDoc2.ApiClientManage.Upload.UploadFileApiProvider
方法	publicvirtualResultUploadFile CreateOrUpdateFileByFilePath(string token, int atFolderId, string uploadFileName, string sourceFilePath)
参数说明	Token:用户凭证 atFolderId:文件夹编号 uploadFileName:上传的文件名 sourceFilePath:源文件路径
返回值	Result=0 成功,

	ResultValue:文件编号 IsUpdateFileVersion:是否更新文件
备注	如果文件存在则更新版本
版本	4.7.0 以上、 LiteV2

接口形式	代理类
代理类	EDoc2.ApiClientManage.Upload.UploadFileApiProvider
方法	<code>public virtual Result UploadFile CreateOrUpdateFileByFilePath(string token, int atFolderId, string uploadFileName, long readSize, string sourceFilePath, UploadFileBlockProcess blockHandler, UploadFileErrorProcess errorHandler)</code>
参数说明	Token:用户凭证 atFolderId:文件夹编号 uploadFileName:上传的文件名 readSize:一次性读取大小(吞吐数据量) sourceFilePath:源文件路径 blockHandler:上传处理方法 errorHandler:错误处理方法
返回值	Result=0 成功, ResultValue:文件编号 IsUpdateFileVersion:是否更新文件
备注	如果文件存在则更新版本
版本	4.7.0 以上、 LiteV2

1.1.1.4 创建或更新文件

接口形式	代理类
代理类	EDoc2.ApiClientManage.Upload.UploadFileApiProvider
方法	<code>public virtual Result UploadFile CreateOrUpdateFile(string token, int atFolderId, string uploadFileName, byte[] fileContent)</code>
参数说明	Token:用户凭证 atFolderId:文件夹编号 uploadFileName:上传的文件名 fileContent:文件二进制内容
返回值	Result=0 成功, ResultValue:文件编号 IsUpdateFileVersion:是否更新文件
备注	如果文件存在则更新版本
版本	4.7.0 以上、 LiteV2

接口形式	代理类
代理类	EDoc2.ApiClientManage.Upload.UploadFileApiProvider
方法	<code>public virtual Result UploadFile CreateOrUpdateFile(string token, int atFolderId, string uploadFileName, long readSize, Stream stream)</code>
参数说明	Token:用户凭证 atFolderId:文件夹编号

	uploadFileName:上传的文件名 readSize:一次性读取大小(吞吐数据量) stream:文件流
返回值	Result=0 成功, ResultValue:文件编号 IsUpdateFileVersion:是否更新文件
备注	如果文件存在则更新版本
版本	4.7.0 以上、LiteV2

接口形式	代理类
代理类	EDoc2.ApiClientManage.Upload.UploadFileApiProvider
方法	<code>public virtual Result UploadFile CreateOrUpdateFile(string token, int atFolderId, string uploadFileName, long readSize, Stream stream, UploadFileBlockProcess blockHandler, UploadFileErrorProcess errorHandler)</code>
参数说明	Token:用户凭证 atFolderId:文件夹编号 uploadFileName:上传的文件名 readSize:一次性读取大小(吞吐数据量) stream:文件流 blockHandler:上传处理方法 errorHandler:错误处理方法
返回值	Result=0 成功, ResultValue:文件编号 IsUpdateFileVersion:是否更新文件
备注	如果文件存在则更新版本
版本	4.7.0 以上、LiteV2

1.1.1.5 通过接口上传文件示例

例：Web 上传

```
int folderId = int.Parse(txtAtFolderId.Text);
string fileName = upFile.FileName;
resDoc = new UploadFileApiProvider().CreateOrUpdateFile(txtToken.Text, folderId,
fileName, 1024 * 10, upFile.PostedFile.InputStream, delegate(UploadFileBlockArgs args)
{
    args.UploadedFileSize * 100 / args.SourceFileSize;
    //处理进度或其它操作
}, delegate(Exception error)
{
    //错误写日志或其它操作
});
```

版本 仅适用 ecm

1.1.1.6 通过 edoc2 页面上传方式

接口形式	Url 地址
Web地址	EDoc2应用地址/Document/File_UploadEx.aspx
参数说明	Token:用户凭证 folderid:文件夹编号 multifile:true false 是否多文件上传 uploadCallback:回调方法 proxy:代理页面
返回值	windows.uploadCallback(fileList):fileList:上传成功后文件列表
备注	通过页面上传文件
示例	<p>//上传文件</p> <pre>function uploadFile(args) { if (isNull(args) isNull(args["Token"]) isNull(args["EDoc2Url"]) isNull(args["DataFn"]) isNull(args["FolderId"])) { return; } if (isNull(args["Width"])) { args["Width"] = 750; } if (isNull(args["Height"])) { args["Height"] = 450; } if (isNull(args["Title"])) { args["Title"] = "Upload files"; } if (isNull(args["MultiFile"])) { args["MultiFile"] = true; } var url = args.EDoc2Url + "Document/File_UploadEx.aspx?folderid=" + args.FolderId.toString() + "&multifile=" + args.MultiFile.toString(); var win=openUploadWnd(url, args.Title, args.Width, args.Height); win.uploadCallback = function (fileList) { //文件列表处理 args.DataFn(win, fileList); }; }</pre> <p>需注意的是调用上传接口后的回调方法在EDoc2V4.8.0和4.8.0以下版本写法不一样 EDoc2 V4.8.0 以下版本:</p> <pre><script type="text/javascript"> var edocUrl = "http://localhost/edoc2v4/"; function showPage() { var url = edocUrl + "Document/File_UploadEx.aspx?folderid=20&multifile=true"; var win = openUploadWnd(url, 480, 360); //指定上传完成的回调函数 win.uploadCallback = function (filelist) { debugger; var dataRows; //遍历上传成功的附件信息 for (var i = 0; i < filelist.length; i++) { } win.close(); }; } function openUploadWnd(url, width, height) { return window.open(url, "_blank", "height=" + height + "px, width=" + width + "px"); } </script></pre> <p>EDoc2 V4.8.0 版本:</p>

	<pre><script type="text/javascript"> var edocUrl = "http://localhost/edoc2v4/"; var win; function showPage() { var url = edocUrl + "Document/File_UploadEx.aspx?folderid=20&multifile=true"; win = openUploadWnd(url, 480, 360); } //指定上传完毕的回调函数 uploadCallback = function (filelist) { debugger; var dataRows; //遍历上传成功的附件信息 for (var i = 0; i < filelist.length; i++) { } win.close(); }; function openUploadWnd(url, width, height) { return window.open(url, "_blank", "height=" + height + "px, width=" + width + "px, toolbar=no, menubar=no, scrollbars=no, resizable=no, location=no, status=no"); } </script></pre>
跨域上传示例	<pre>function openUploadWnd(url, title,width, height) { return window.open(url, title, "height=" + height + "px, width=" + width + "px, toolbar=no, menubar=no, scrollbars=no, resizable=no, location=no, status=no"); } //上传文件 function uploadFile(args) { if (isNull(args) isNull(args["Token"]) isNull(args["EDoc2Url"]) isNull(args["DataFn"]) isNull(args["FolderId"])) { return; } if (isNull(args["Width"])) { args["Width"] = 750; } if (isNull(args["Height"])) { args["Height"] = 450; } if (isNull(args["Title"])) { args["Title"] = "Upload files"; } if (isNull(args["MultiFile"])) { args["MultiFile"] = true; } var url = args.EDoc2Url + "Document/File_UploadEx.aspx?token=" + args.Token + "&folderid=" + args.FolderId.toString() + "&multifile=true&uploadCallback=parent.opener.uploadCallback&proxy=" + encodeURIComponent("http://michael/OMGWebSite/Proxy.aspx"); window.uploadCallback = function (fileList) { //文件列表处理 args.DataFn(window, fileList); } openUploadWnd(url, args.Title, args.Width, args.Height); }</pre>
版本	仅适用 ecm

1.1.2 获取文件信息

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public ResultEDocFileInfo GetFileInfoByFileId(string token, int fileId)</code>
参数说明	Token:用户凭证 fileId:文件编号
返回值	Result=0 成功,EDocFileInfo:文件信息
版本	4.7.0 以上、LiteV2

1.1.3 根据版本编号获取文件信息

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	publicResultEDocFileInfo GetFileInfoByFileVerId(string token, int fileVerId)
参数说明	Token:用户凭证 fileVerId:文件版本编号
返回值	Result=0 成功,EDocFileInfo[]:文件信息列表
版本	4.7.0 以上、LiteV2

1.1.4 获取指定文件夹下文件列表

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	publicResultEDocFileInfoList GetChildFileList(string token, int folderId)
参数说明	Token:用户凭证 folderId:文件夹编号
返回值	Result=0 成功,EDocFileInfo[]:文件信息列表
版本	4.7.0 以上、LiteV2
接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	publicResultEDocFileInfoList GetChildFileListExpand(string token, int folderId, bool isChild)
参数说明	Token:用户凭证 folderId:文件夹编号 isChild:是否加载子文件
返回值	Result=0 成功,EDocFileInfo[]:文件信息列表
版本	4.7.0 以上、LiteV2

1.1.5 获取回收站文件列表

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	publicResultEDocFileInfoList GetRecycleFileList(string token)
返回值	Result=0 成功,EDocFileInfo[]:文件信息列表
版本	4.7.0 以上、LiteV2

1.1.6 文件是否存在

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	publicResultFindFile ExistsFileByFileId(string token, int fileId)
参数	Token:用户凭证 fileId:文件编号
返回值	Result=0 成功,

	IsExists:是否存在, FindFileId:文件编号, FindFileParentFolderId:所在文件夹编号
接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	publicResultFindFile ExistsFileByFolder(string token, int atFolderId, string fileName)
参数	Token:用户凭证 atFolderId:文件夹编号, filename:文件名
返回值	Result=0 成功, IsExists:是否存在, FindFileId:文件编号, FindFileParentFolderId:所在文件夹编号
版本	4.7.0 以上、LiteV2

1.1.7 修改文件名

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	publicint ChanageFileName(string token, int fileId, string newFileName, string newRemark)
参数	Token:用户凭证 fileId:文件编号, newFileName:文件新命名, newRemark:文件新备注
返回值	Result=0 成功,
版本	4.7.0 以上、LiteV2

1.1.8 指派文件的创建者和修改者

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	publicint UpdateFileCreatorOrModifyOperator(string token, int oldUserId, bool isCreate, int newUserId, params int[] fileIds)
参数	Token:用户凭证 oldUserId:旧用户编号, isCreate:是创建者还是修改者 newUserId:新用户编号 fileIds:文件编号列表
返回值	Result=0 成功,
版本	4.7.0 以上、LiteV2

1.1.9 移动文件列表

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider

方法	<code>publicint MoveFileList(string token, int targetFolderId, bool dropPerms, paramsint[] fileIds)</code>
参数	Token:用户凭证 targetFolderId:目标文件夹编号, dropPerms:是否删除权限, fileIds:文件编号列表
返回值	Result=0 成功,
版本	4.7.0 以上、LiteV2

1.1.10 复制文件列表

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>publicint CopyFileList(string token, int targetFolderId, bool copyMeta, bool copyPerm, paramsint[] fileIds)</code>
参数	Token:用户凭证 targetFolderId:目标文件夹编号, copyMeta:是否复制元数据, copyPerm是否复制权限 fileIds:文件编号列表
返回值	Result=0 成功,
版本	4.7.0 以上、LiteV2

1.1.11 删除文件列表

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>publicint DeleteFileList(string token, bool intoRecycleBin, paramsint[] fileIds)</code>
参数	Token:用户凭证 intoRecycleBin:是否删除到回收站, fileIds:文件编号列表
返回值	Result=0 成功,
版本	4.7.0 以上、LiteV2

1.1.12 清空文件

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>publicint ClearFileList(string token, paramsint[] fileIds)</code>
参数	Token:用户凭证 fileIds:文件编号列表
返回值	Result=0 成功,
版本	4.7.0 以上、LiteV2

1.1.13 恢复文件列表

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public int RestoreFileList(string token, params int[] fileIds)</code>
参数	Token:用户凭证 fileIds:文件编号列表
返回值	Result=0 成功,
版本	4.7.0 以上、LiteV2

1.1.14 发布文件版本

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public int PublishFileVersion(string token, int fileId, int taskStarterId)</code>
参数	Token:用户凭证 fileId:文件编号 taskStarterId:发起人编号
返回值	Result=0 成功,
版本	4.7.0 以上、LiteV2

1.1.15 批量创建关联文件

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public int CreateRelationFile(string token, int fileId, params int[] relateFileIds)</code>
参数	Token:用户凭证 fileId:文件编号 relateFileIds:要关联的文件编号列表
返回值	Result=0 成功,
版本	4.7.0 以上、LiteV2

1.1.16 获取关联文件列表

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public int GetRelatedFiles(string token, int fileId)</code>
参数	Token:用户凭证 fileId:文件编号
返回值	Result=0 成功,EDocRelateFileInfo[] 关联文件信息
版本	4.7.0 以上、LiteV2

1.1.17 删除关联文件

接口形式	代理类
------	-----

代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public int RemoveFileRelation(string token, int fileId, int relateFileId)</code>
参数	Token:用户凭证 fileId:文件编号 relateFileId:已关联的文件编号
返回值	Result=0 成功,
版本	4.7.0 以上、LiteV2

1.1.18 文档预览

接口形式	Url 地址
Web地址	EDoc2应用地址/Preview.aspx
参数说明	Token:用户凭证 loginName:用户帐号 FileId:预览文件编号
返回值	windows.uploadCallback(fileList):fileList:上传成功后文件列表
示例	预览 Id 为 201 的文件的请求地址: EDoc2应用地址/ Preview.aspx? FileId =201&calendarCode=1 如 201 文件上一版本的版本 Id 为 198, 那么预览该版本文件请求地址为: EDoc2应用地址/ Preview.aspx? FileVerId =198&calendarCode=1
备注	通过页面预览文件
版本	仅适用 ecm

1.1.19 文档下载

接口形式	Url 地址
Web地址	EDoc2应用地址/Document/File_Download.aspx
参数说明	Token:用户凭证 loginName:用户帐号 File_Id:下载文件编号
示例	下载 Id 为 201 的文件的请求地址: EDoc2应用地址/ Document/File_Download.aspx?file_Id=201 如 201 文件上一版本的版本 Id 为 198, 那么下载该版本文件请求地址为: EDoc2应用地址/ Document/File_Download.aspx?ver_id=198
返回值	文件流
版本	仅适用 ecm

1.1.20 文件选择

接口形式	Url 地址
Web地址	EDoc2应用地址/AppExt/Common/SelectEDocFile.aspx
参数说明	Token:用户凭证 loginName:用户帐号 FileId:预览文件编号
返回值	例如用window.showModalDialog方式打开选择文档窗口, 那么操作结果将通过window.returnValue返回 Js代码片段: <code>var res = window.showModalDialog(url, "", "dialogWidth:750px; dialogHeight:450px;");</code>

	<p>返回值解析：</p> <pre> res:Json数组，选中的文档 res [0]._data :{ curVerId:"当前版本Id", curVerNumStr:"当前版本编号", id:"主键Id，唯一标识", metaTypeId:"元数据类型Id", metaTypeName:"元数据类型名称", name:"文档名称", path:"文档完全路径", size:"文档大小" }</pre>
示例	<pre> //选择文件 function selectFile(args) { if (isNull(args) isNull(args["Token"]) isNull(args["EDoc2Url"]) isNull(args["DataFn"])) { return; } //if(isNull(args["FolderId"])){ arg if (isNull(args["Width"])) { args["Width"] = 750; } if (isNull(args["Height"])) { args["Height"] = 450; } if (isNull(args["Title"])) { args["Title"] = "Upload files"; } var url = args.EDoc2Url + "AppExt/Common/SelectEDocFile.aspx?tkn=" + args.Token; var fileList = window.showModalDialog(url, args.Title, "dialogWidth:" + args.Width.toString() + "px; dialogHeight:" + args.Height.toString() + ";"); if (fileList != null&&fileList.length > 0) { args.DataFn(fileList); } } }</pre>
跨域示例 (参考Demo)	<pre> //选择文件 function selectFile(args) { if (isNull(args) isNull(args["Token"]) isNull(args["EDoc2Url"]) isNull(args["DataFn"])) { return; } if (isNull(args["Width"])) { args["Width"] = 750; } if (isNull(args["Height"])) { args["Height"] = 450; } if (isNull(args["Title"])) { args["Title"] = "Upload files"; } var url = "../SelectProxy.aspx?SelectType=File&tkn=" + args.Token + "&proxy=" + encodeURIComponent(getProjectUrl() + "Proxy.aspx"); var resValue = window.showModalDialog(url, window, "dialogWidth:" + args.Width.toString() + "px; dialogHeight:" + args.Height.toString() + ";"); if (resValue != null&& resValue.length > 0) { args.DataFn(resValue); } } }</pre>
版本	仅适用 ecm

当前接口

Edoc2应用地址/AppExt/Common/SelectEdocFile.aspx?loginName=[loginNmae]

Edoc2应用地址/AppExt/Common/SelectEdocFile.aspx?tkn=[tkn]

1.1.21 发布文件主版本(4.8.1)

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public int PublishFileMainVersion(string token, int fileId)</code>
参数	Token:用户凭证 fileId:文件编号
返回值	Result=0 成功,
版本	4.8.1

1.2 文件夹操作

1.2.1 获取文件夹信息

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public ResultEDocFolderInfo GetFolderInfoByFolderId(string token, int folderId)</code>
参数说明	Token:用户凭证 folderId:文件夹编号
返回值	Result=0 成功,EDocFolderInfo:文件夹信息
版本	4.7.0 以上、LiteV2

1.2.2 获取子文件夹列表

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public ResultEDocFolderInfoList GetChildFolderInfoListByFolderId(string token, int folderId)</code>
参数说明	Token:用户凭证 folderId:文件夹编号
返回值	Result=0 成功,EDocFolderInfo[]:文件夹列表信息
版本	4.7.0 以上、LiteV2

1.2.3 是否存在相同文件夹

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public ResultBool ExistsFolder(string token, int folderParentId, string folderName)</code>
参数说明	Token:用户凭证

	folderParentId:文件夹编号 folderName:文件夹名称
返回值	Result=0 不存在, 否则返回已经存在的文件夹 ID,ResultValue:true false 文件夹是否存在
版本	4.7.0 以上、LiteV2

1.2.4 创建文件夹

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public ResultInt CreateFolder(string token, int parentFolderId, string folderName, string folderCode, long folderMaxSize, long fileMaxSize, long alertSize, string remark, string forbiddenExts, int securityLevelId, DateTime createTime, DateTime modifyTime, string creatorLoginName, string modifierLoginName)</code>
参数说明	Token:用户凭证 parentFolderId:父文件夹编号 folderName:文件夹名称 folderCode:文件夹编码 folderMaxSize:文件夹最大尺寸 fileMaxSize:文件最大尺寸 alertSize:提示大小 remark:备注 forbiddenExts:禁止上传文件扩展名 securityLevelId:安全等级编号 createTime:创建时间 modifyTime:修改时间 creatorLoginName:创建人帐号 modifierLoginName:修改人帐号
返回值	Result=0 成功,ResultValue:文件夹编号
版本	4.7.0 以上、LiteV2

1.2.5 更新文件夹

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public int UpdateFolder(string token, int folderId, string folderName, string folderCode, string remark, long fileMaxSize, long folderMaxSize, long folderAlertSize, string forbiddenExts, int sortOrder, int securityLevelId)</code>
参数说明	Token:用户凭证 folderId:文件夹编号 folderName:文件夹名称 folderCode:文件夹编码 folderMaxSize:文件夹最大尺寸 fileMaxSize:文件最大尺寸 folderAlertSize:提示大小

	forbiddenExts:禁止上传文件扩展名 sortOrder:排序值 securityLevelId:安全等级编号
返回值	Result=0 成功,ResultValue:文件夹编号
版本	4.7.0 以上

1.2.6 修改文件夹名称

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	public int ChangeFolderName(string token, int folderId, string newName)
参数说明	Token:用户凭证 folderId:文件夹编号 newName:文件夹新命名
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.2.7 移动文件夹列表

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	public int MoveFolderList(string token, int targetFolderId, bool dropPerms, paramsint [] folderIds)
参数说明	Token:用户凭证 targetFolderId:目标文件夹 dropPerms:是否删除权限 folderIds:移动的文件夹编号列表
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.2.8 复制文件夹列表

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	public int CopyFolderList(string token, int targetFolderId, bool copyMeta, bool copyPerm, paramsint [] folderIds)
参数说明	Token:用户凭证 targetFolderId:目标文件夹 copyMeta:是否复制元数据 copyPerms:是否复制权限 folderIds:移动的文件夹编号列表
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.2.9 删除文件夹列表

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	public int DeleteFolderList(string token, bool intoRecycleBin, paramsint [] folderIds)
参数说明	Token:用户凭证 intoRecycleBin:是否删除回收站(true:存放到回收站, false:直接删除) folderIds:文件夹编号列表
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.2.10 获取回收站文件夹列表

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	public Result EDocFolderInfoList GetRecycleFolderList(string token)
参数说明	Token:用户凭证
返回值	Result=0 成功,EDocFolderInfo[]:文件夹列表信息
版本	4.7.0 以上、LiteV2

1.2.11 清空文件夹

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	public int ClearFolderList(string token, paramsint [] folderIds)
参数说明	Token:用户凭证 folderIds:文件夹编号列表 注:清空当前文件夹, 并删除当前文件夹
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.2.12 恢复文件夹列表

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	public int RestoreFolderList(string token, paramsint [] folderIds)
参数说明	Token:用户凭证 folderIds:文件夹编号列表
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.2.13 获取顶级公共文件夹列表

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	public Result EDocFolderInfoList GetTopPublicFolderList(string token)

参数说明	Token:用户凭证
返回值	Result=0 成功,EDocFolderInfo[]:文件夹列表信息
版本	4.7.0 以上、LiteV2

1.2.14 获取顶级个人文件夹列表

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public ResultEDocFolderInfoList GetTopPersonalFolderList(string token)</code>
参数说明	Token:用户凭证
返回值	Result=0 成功,EDocFolderInfo[]:文件夹列表信息
版本	4.7.0 以上、LiteV2

1.2.15 选择文件夹

接口形式	Url 地址
Web地址	EDoc2应用地址/AppExt/Common/SelectEDocFolder.aspx
参数说明	Token:用户凭证 loginName:用户帐号 FileId:预览文件编号
返回值	<p>例如用window.showModalDialog方式打开选择文件夹窗口，那么操作结果将通过window.returnValue返回</p> <p>Js代码片段：</p> <pre>var res = window.showModalDialog(url, "", "dialogWidth:750px; dialogHeight:450px;");</pre> <p>返回值解析：</p> <pre>res:Json数组，选中的文件夹 res [0]._data :{ alertSize:"", childFileCount:"该文件夹下文档个数", childFolderCount:"该文件夹下文件夹个数", code:"编号", createTime:"创建时间", creatorId:"创建人Id", creatorName:"创建人名称", enabledFolderTypeSet:"", folderType:"", forbiddenFileExtensions:"禁止上传的文档后缀名", id:"主键Id, 唯一标识", maxFileSize:"允许上传的最大文档大小", maxFolderSize:"", modifyTime:"更新时间", name:"名称", parentFolderId:"父文件夹Id",</pre>

	<pre>parentFolderName:"父文件夹名称", path:"文件夹路径", permission:"", remark:"备注", securityLevel:"", size:"大小", state:"状态", type:"类型" }</pre>
示例	<pre>//选择文件 function selectFile(args) { if (isNull(args) isNull(args["Token"]) isNull(args["EDoc2Url"]) isNull(args["DataFn"])) { return; } //if(isNull(args["FolderId"])){arg if (isNull(args["Width"])) { args["Width"] = 750; } if (isNull(args["Height"])) { args["Height"] = 450; } if (isNull(args["Title"])) { args["Title"] = "Upload files"; } var url = args.EDoc2Url + "AppExt/Common/SelectEDocFile.aspx?tkn=" + args.Token; var folderList = window.showModalDialog(url, args.Title, "dialogWidth:" + args.Width.toString() + "px; dialogHeight:" + args.Height.toString() + ";"); if (folderList != null && folderList.length > 0) { args.DataFn(folderList); } }</pre>
跨域示例 (参考Demo)	<pre>//选择文件夹 function selectEDoc2Folder(args) { if (isNull(args) isNull(args["Token"]) isNull(args["EDoc2Url"]) isNull(args["DataFn"])) { return; } if (isNull(args["Width"])) { args["Width"] = 750; } if (isNull(args["Height"])) { args["Height"] = 450; } if (isNull(args["Title"])) { args["Title"] = "Select Folder"; } var url = "../SelectProxy.aspx?SelectType=Folder&tkn=" + args.Token + "&proxy=" + encodeURIComponent(getProjectUrl() + "Proxy.aspx"); var resValue = window.showModalDialog(url, window, "dialogWidth:" + args.Width.toString() + "px; dialogHeight:" + args.Height.toString() + ";"); if (resValue != null && resValue.length > 0 && resValue[0] != null) { args.DataFn(resValue[0]); } }</pre>
版本	仅适用 ecm

1.2.16 设置和修改文件夹安全策略(4.8.1)

接口形式	代理类
------	-----

代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public int SetFolderSecurityStrategy(string token, int folderId, EDocFolderSecurityStrategy strategy)</code>
参数说明	<p>token:用户凭证 folderId:文件夹编号 strategy:文件夹安全策略</p> <p>EDocFolderSecurityStrategy strategy 属性说明</p> <p><code>bool Enabled</code> 是否启用安全策略 <code>DateTime StartTime</code>开始时间 <code>DateTime ExpiredTime</code>结束时间 <code>bool MustOnline</code>是否强制在线使用 <code>string IpRange</code>允许的IP范围 <code>ValidateType ValidateType</code>验证方式 <code>string MacAddress</code>允许的mac地址 <code>bool EnabledOutSend</code>是否允许外发文档 <code>bool EncryptSvrFile</code>是否加密服务器文件 <code>int CiphertextOutwardPolicy</code>密文外发策略</p>
返回值	Result=0 成功
版本	4.8.1

1.2.17 获取文件夹安全策略(4.8.1)

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public ResultEDocFolderSecurityStrategy GetFolderSecurityStrategy(string token, int folderId)</code>
参数说明	<p>token:用户凭证 folderId:文件夹编号</p>
返回值	<p>Result=0 成功</p> <p>EDocFolderSecurityStrategy : 文件夹安全策略对象</p> <p>EDocFolderSecurityStrategy strategy 对象属性说明</p> <p><code>bool Enabled</code> 是否启用安全策略 <code>DateTime StartTime</code>开始时间 <code>DateTime ExpiredTime</code>结束时间 <code>bool MustOnline</code>是否强制在线使用 <code>string IpRange</code>允许的IP范围 <code>ValidateType ValidateType</code>验证方式 <code>string MacAddress</code>允许的mac地址 <code>bool EnabledOutSend</code>是否允许外发文档 <code>bool EncryptSvrFile</code>是否加密服务器文件 <code>int CiphertextOutwardPolicy</code>密文外发策略</p>
版本	4.8.1

1.3 文件和文件夹混合

1.3.1 批量删除文件夹和文件

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public int DeleteFolderListAndFileList(string token, int[] folderIds, int[] fileIds, bool intoRecycleBin)</code>
参数说明	Token:用户凭证 intoRecycleBin:是否删除回收站 folderIds:文件夹编号列表 fileIds:文件编号列表
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.3.2 批量复制文件夹和文件

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public int CopyFolderListAndFileList(string token, int[] folderIds, int[] fileIds, int targetFolderId, bool copyMeta, bool copyPerm, bool copyCustEvents)</code>
参数说明	Token:用户凭证 folderIds:文件夹编号列表 fileIds:文件编号列表 copyMeta:是否复制元数据 copyPerm:是否复制权限 copyCustEvents:是否复制主动事件(Lite 版本只能设置为 false)
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.3.3 批量移动文件夹和文件

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public int MoveFolderListAndFileList(string token, int[] folderIds, int[] fileIds, int targetFolderId, bool dropPerms)</code>
参数说明	Token:用户凭证 folderIds:文件夹编号列表 fileIds:文件编号列表 targetFolderId:目标文件夹 dropPerms:是否删除权限
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.3.4 恢复文件夹列表和文件列表

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	public int RestoreFolderListAndFileList(string token, int[] folderIds, int[] fileIds)
参数说明	Token:用户凭证 folderIds:文件夹编号列表 fileIds:文件编号列表
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.3.5 清空文件夹列表和文件列表

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	public int Clear(string token, int[] folderIds, int[] fileIds)
参数说明	Token:用户凭证 folderIds:文件夹编号列表 fileIds:文件编号列表
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.3.6 清空所有文件夹和文件

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	public int ClearAll(string token)
参数说明	Token:用户凭证 注:清空回收站中的所有文件和文件夹
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.4 共享操作

1.4.1 获取共享给我的列表

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	public int GetAllShareEntry(string token, int pageIndex, int pageSize)
参数	Token:用户凭证 pageIndex:当前页 pageSize:分页大小
返回值	Result=0 成功,RealPageSize:实际分页大小,Count:总数,EDocShareInfo[]:当前页共享信息列表
版本	4.7.0 以上

1.4.2 获取我共享的列表

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public int GetMyShareEntry(string token, int pageIndex, int pageSize)</code>
参数	Token:用户凭证 pageIndex:当前页 pageSize:分页大小
返回值	Result=0 成功,RealPageSize:实际分页大小,Count:总数,EDocShareInfo[]:当前页共享信息列表
版本	4.7.0 以上

1.5 文档权限

1.5.1 权限相关类

1.5.1.1 成员类型

```
///<summary>
///用户
///</summary>
publicconstint MEMBER_USER = 1;
///<summary>
///部门
///</summary>
publicconstint MEMBER_DEPARTMENT = 2;
///<summary>
///职位
///</summary>
publicconstint MEMBER_POSITION = 4;
///<summary>
///用户组
///</summary>
publicconstint MEMBER_USER_GROUP = 8;
///<summary>
///联系人
///</summary>
publicconstint MEMBER_CONTACTER = 16;
```

1.5.1.2 基权值

请参考 EDoc2ApiConst 类中以 PERM_开头均为基权值。

1.5.2 文件权限

1.5.2.1 读取文件权限

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public ResultEDocPermissionEntryList ReadFilePermission(string token, int fileId)</code>
参数说明	Token:用户凭证 fileId:文件编号
返回值	Result=0 成功,EDocPermissionEntry[]权限列表
版本	4.7.0 以上、LiteV2

1.5.2.2 获取指定文件和用户的权限值

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public ResultInt GetFileUserPermission(string token, int fileId, int userId)</code>
参数说明	Token:用户凭证 fileId:文件编号 userId:用户编号
返回值	Result=0 成功, ResultValue 权限值
版本	4.7.0 以上、LiteV2

1.5.2.3 设置文件权限

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public int SetFilePermission(string token, int fileId, bool isMerge, params EDocPermissionEntry[] permList)</code>
参数说明	Token:用户凭证 fileId:文件编号 isMerge:权限存在是否将原有权限和设置权限合并 EDocPermissionEntry[]:权限列表
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public int SetFileCurrentPermission(string token, int fileId, bool recursive, bool isMerge, int memberId, int memberType, params int[] basicPermValues)</code>
参数说明	Token:用户凭证 fileId:文件编号 isMerge:权限存在是否将原有权限和设置权限合并 memberId:成员编号 memberType:成员类型 basicPermValues:当前文件夹基权列表

返回值	Result=0 成功
示例	例如:要设置当前文件夹给指定用户有显示, 新建文件夹权限 <code>new DocumentApiProvider().SetFileCurrentPermission(token, folderId, false, true, memberId, EDoc2ApiConst.MEMBER_USER, //成员类型用户 EDoc2ApiConst.PERM_DISPLAY, //显示基权 EDoc2ApiConst.PERM_DOWNLOAD) //下载文件基权</code>
版本	4.7.0 以上、LiteV2

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public int SetFileCurrentPermission(string token, int fileId, bool isMerge, int memberId, int memberType, DateTime startTime, DateTime expiredTime, params int[] basicPermFilesValues)</code>
参数说明	token: 用户凭证 fileId: 文件编号 isMerge: 是否合并权限 memberId: 成员编号 memberType: 成员类型 startTime: 开始时间 expiredTime: 结束时间 basicPermFilesValues: 基权列表
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public int SetFilePermission(string token, int fileId, bool isMerge, int memberId, int memberType, int[] basicPermFilesValues, int[] basicPermFileVersValues, DateTime startTime, DateTime expiredTime)</code>
参数说明	Token: 用户凭证 fileId: 文件编号 isMerge: 权限存在是否将原有权限和设置权限合并 memberId: 成员编号 memberType: 成员类型 basicPermFilesValues: 文件基权列表 basicPermFileVersValues: 历史文件基权列表 startTime: 开始时间 expiredTime: 结束时间
返回值	Result=0 成功

版本	4.7.0 以上、LiteV2
接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public int SetFilePermission(string token, int fileId, bool isMerge, int memberId, int memberType, int[] basicPermFilesValues, int[] basicPermFileVersValues)</code>
参数说明	Token:用户凭证 fileId:文件编号 isMerge:权限存在是否将原有权限和设置权限合并 memberId:成员编号 memberType:成员类型 basicPermFilesValues:文件基权列表 basicPermFileVersValues:历史文件基权列表
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.5.2.4 删除文件权限

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public int DeleteFilePermission(string token, int fileId, bool isRemove, params EDocPermissionEntry[] permList)</code>
参数说明	Token:用户凭证 fileId:文件编号 isRemove:权限存在是否只剥离指定的权限 EDocPermissionEntry[] permList: 要删除的文件权限列表
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public int DeleteFilePermission(string token, int fileId, bool isRemove, int memberId, int memberType, int[] basicPermFilesValues, int[] basicPermFileVersValues)</code>
参数说明	Token:用户凭证 fileId:文件编号 isRemove:权限存在是否只剥离指定的权限 memberId:成员编号 memberType:成员类型 basicPermFilesValues:文件基权列表 basicPermFileVersValues:历史文件基权列表
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.5.2.5 文件是否有权限

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	publicResultBool HavePermissionInFile(string token, int fileId, int userId, int permValue)
参数说明	Token:用户凭证 fileId:文件编号 userId:用户编号 permValue:基权值
返回值	Result=0 成功,ResultValue:true/false 是否有权限
版本	4.7.0 以上、LiteV2

1.5.3 文件夹权限

1.5.3.1 读取文件夹权限

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	publicResultEDocPermissionEntryList ReadFolderPermission(string token, int folderId)
参数说明	Token:用户凭证 folderId:文件夹编号
返回值	Result=0 成功,EDocPermissionEntry[]权限列表
版本	4.7.0 以上、LiteV2

1.5.3.2 获取指定文件夹和用户的权限信息

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	publicResultEDocPermissionEntry GetPermEntryInfoByUserId(string token, int folderId, int userId)
参数说明	Token:用户凭证 folderId:文件夹编号 userId:用户编号
返回值	Result 权限值 (0:无权限, 1:只读, 133569:读写, 43682241:完全控制) EDocPermissionEntry 权限信息
版本	4.7.0 以上、LiteV2

1.5.3.3 获取指定文件夹和用户的权限值

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	publicResultInt GetFolderUserPermission(string token, int folderId, int userId)
参数说明	Token:用户凭证 folderId:文件夹编号 userId:用户编号
返回值	Result=0 成功, ResultValue 权限值

版本	4.7.0 以上、LiteV2
----	-----------------

1.5.3.4 设置文件夹权限

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public int SetFolderCurrentPermission(string token, int folderId, bool recursive, bool isMerge, int memberId, int memberType, params int[] basicPermValues)</code>
参数说明	Token:用户凭证 folderId:文件夹编号 isMerge:权限存在是否将原有权限和设置权限合并 memberId:成员编号 memberType:成员类型 basicPermValues:当前文件夹基权列表
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public int SetFolderCurrentPermission(string token, int folderId, bool recursive, bool isMerge, int memberId, int memberType, DateTime startTime, DateTime expiredTime, params int[] basicPermValues)</code>
参数说明	Token:用户凭证 folderId:文件夹编号 isMerge:权限存在是否将原有权限和设置权限合并 memberId:成员编号 memberType:成员类型 startTime:开始时间 expiredTime:结束时间 basicPermValues:当前文件夹基权列表
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2
示例	例如:要设置当前文件夹给指定用户有显示, 新建文件夹权限 <code>new DocumentApiProvider().SetFolderCurrentPermission(token, folderId, false, true, memberId, EDoc2ApiConst.MEMBER_USER, //成员类型用户 EDoc2ApiConst.PERM_DISPLAY, //显示基权 EDoc2ApiConst.PERM_CREATE_FOLDER) //创建文件夹基权</code>

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider

方法	<code>public int SetFolderPermission(string token, int folderId, bool recursive, params EDocPermissionEntry[] permList)</code>
参数说明	Token:用户凭证 fileId:文件编号 isMerge:权限存在是否将原有权限和设置权限合并 EDocPermissionEntry[]:权限列表
返回值	Result=0 成功
版本	4.7.0 以上、 LiteV2

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>SetFolderPermission(string token, int folderId, bool recursive, bool isMerge, int memberId, int memberType, int[] basicPermissionValues, int[] basicPermFoldersValues, int[] basicPermFilesValues, int[] basicPermFileVersValues)</code>
参数说明	Token:用户凭证 fileId:文件编号 isMerge:权限存在是否将原有权限和设置权限合并 memberId:成员编号 memberType:成员类型 basicPermPermissionValues:当前文件夹基权列表 basicPermFoldersValues:子文件夹基权列表 basicPermFilesValues:文件基权列表 basicPermFileVersValues:历史文件基权列表
返回值	Result=0 成功
版本	4.7.0 以上、 LiteV2

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>SetFolderPermission(string token, int folderId, bool recursive, bool isMerge, int memberId, int memberType, int[] basicPermissionValues, int[] basicPermFoldersValues, int[] basicPermFilesValues, int[] basicPermFileVersValues, DateTime startTime, DateTime expiredTime)</code>
参数说明	Token:用户凭证 fileId:文件编号 isMerge:权限存在是否将原有权限和设置权限合并 memberId:成员编号 memberType:成员类型 basicPermPermissionValues:当前文件夹基权列表 basicPermFoldersValues:子文件夹基权列表 basicPermFilesValues:文件基权列表 basicPermFileVersValues:历史文件基权列表 startTime:开始时间 expiredTime:结束时间
返回值	Result=0 成功
版本	4.7.0 以上、 LiteV2

1.5.3.5 删除文件夹权限

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public int DeleteFolderPermission(string token, int folderId, bool recursive, bool isRemove, int memberId, int memberType, int[] basicPermissionValues, int[] basicPermFoldersValues, int[] basicPermFilesValues, int[] basicPermFileVersValues)</code>
参数说明	Token:用户凭证 fileId:文件编号 isRemove:权限存在是否只剥离指定的权限 memberId:成员编号 memberType:成员类型 basicPermPermissionValues:当前文件夹基权列表 basicPermFoldersValues:子文件夹基权列表 basicPermFilesValues:文件基权列表 basicPermFileVersValues:历史文件基权列表
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.5.3.6 文件夹是否有权限

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public ResultBool HavePermissionInFolder(string token, int folderId, int userId, int permValue)</code>
参数说明	Token:用户凭证 folderId:文件夹编号 userId:用户编号 permValue:基权值
返回值	Result=0 成功,ResultValue:true false 是否有权限
版本	4.7.0 以上、LiteV2

1.6 元数据操作(仅适用 ECM)

1.6.1 元数据枚举列表

```
public enum MetaRange
{
    ALL = 32,
    FORM = 2,
    ORG = 4,
    OTHER = 16,
    WF = 8,
    DOC = 1,
```

}

1.6.2 元数据类型操作

1.6.2.1 获取所有元数据类型列表

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	public ResultMetaTypeInfoList GetAllMetaType(string token)
参数说明	Token:用户凭证
返回值	Result=0 成功,EDocMetaTypeInfo[]元数据类型信息列表
版本	4.7.0 以上

1.6.2.2 创建元数据类型

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	public ResultInt CreateMetaType(string token, EDocMetaTypeInfo metaTypeInfo, int metaRange)
参数说明	Token:用户凭证 metaTypeInfo:元数据类型信息 metaRange:范围
返回值	Result=0 成功,ResultValue 元数据类型编号
版本	4.7.0 以上

1.6.2.3 获取元数据类型编号

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	public int SearchMetaTypeId(string token, string typeName)
参数说明	Token:用户凭证 typeName:类型名称
返回值	元数据类型编号
版本	4.7.0 以上

1.6.2.4 获取元数据类型

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	public ResultEDocMetaTypeInfo GetMetaTypeInfoByCode(string token, string metaTypeCode)
参数说明	Token:用户凭证 metaTypeCode:元数据类型编码
返回值	Result=0 成功,EDocMetaTypeInfo:元数据类型信息
版本	4.7.0 以上

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public ResultEDocMetaTypeInfo GetMetaTypeInfoByName(string token,string metaTypeCode)</code>
参数说明	Token:用户凭证 metaTypeCode:元数据类型编码
返回值	Result=0 成功,EDocMetaTypeInfo:元数据类型信息
版本	4.7.0 以上

1.6.3 元数据属性操作

1.6.3.1 根据元数据属性编号获取配置信息

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public ResultEDocMetaAttrConfigInfo GetMetaAttrConfigByAttrId(string token,int attrId)</code>
参数说明	Token:用户凭证 attrId:属性编号
返回值	Result=0 成功,EDocMetaAttrConfigInfo 元数据配置信息
版本	4.7.0 以上

1.6.3.2 根据元数据属性编号设置配置信息

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public int SetMetaAttrConfigByAttrId(string token,int attrId,EDocMetaAttrConfigInfo cfgInfo)</code>
参数说明	Token:用户凭证 attrId:属性编号 EDocMetaAttrConfigInfo:属性配置信息
返回值	Result=0 成功
版本	4.7.0 以上

1.6.3.3 根据属性编码获取属性信息

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public ResultEDocMetaAttrInfo GetMetaAttrInfoByAttrCode(string token,int metaTypeId,string attrCode)</code>
参数说明	Token:用户凭证 metaTypeId:元数据类型编号 attrCode:属性编码
返回值	Result=0 成功,EDocMetaAttrInfo:元数据属性信息
版本	4.7.0 以上

1.6.3.4 根据属性名称获取属性信息

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public ResultEDocMetaAttrInfo GetMetaAttrInfoByAttrName(string token,int metaTypeId,string attrCode)</code>
参数说明	Token:用户凭证 metaTypeId:元数据类型编号 attrName:属性名称
返回值	Result=0 成功,EDocMetaAttrInfo:元数据属性信息

1.6.3.5 更新属性值

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public int UpdateByAttrId(string token,int attrId, int metaTypeId,string newAttrValue,string whereAttrValue)</code>
参数说明	Token:用户凭证 attrId:元数据属性编号 metaTypeId:元数据类型编号 newAttrValue:新属性值 whereAttrValue:条件值(空:无条件更新)
返回值	Result=0 成功
版本	4.7.0 以上

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public int UpdateByAttrId(string token,int attrId, int metaTypeId,string newAttrValue,string whereAttrValue,int metaRange)</code>
参数说明	Token:用户凭证 attrId:元数据属性编号 metaTypeId:元数据类型编号 newAttrValue:新属性值 whereAttrValue:条件值(空:无条件更新) metaRange:范围
返回值	Result=0 成功
版本	4.7.0 以上

1.6.4 元数据与文件夹操作

1.6.4.1 清空文件夹元数据

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public int ClearFolderMeta(string token,int folderId)</code>
参数说明	Token:用户凭证

	folderId: 文件编号
返回值	Result=0 成功
版本	4.7.0 以上

1.6.4.2 获取文件夹元数据列表

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public ResultMeta GetFolderMetaList(string token,int folderId)</code>
参数说明	Token: 用户凭证 folderId: 文件编号
返回值	Result=0 成功, EDocMetaObjectTypeInfo[] 元数据列表
版本	4.7.0 以上

1.6.4.3 针对文件夹添加元数据信息

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public int AddFolderMetaByMetaTypeName(string token,int folderId,string metaTypeName,bool blUpdateStrategy,params MetaItemInfo[] items)</code>
参数说明	Token: 用户凭证 folderId: 文件夹编号 metaTypeName: 元数据类型名称 blUpdateStrategy: 是否更新策略 MetaItemInfo[]: 元数据信息列表
返回值	Result=0 成功
版本	4.7.0 以上

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public int AddFolderMetaByMetaTypeId(string token,int folderId,string metaTypeId,bool blUpdateStrategy,params MetaItemInfo[] items)</code>
参数说明	Token: 用户凭证 folderId: 文件夹编号 metaTypeId: 元数据类型编号 blUpdateStrategy: 是否更新策略 MetaItemInfo[]: 元数据信息列表
返回值	Result=0 成功
版本	4.7.0 以上

1.6.5 元数据与文件操作

1.6.5.1 清空文件元数据

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public int ClearFileMeta(string token,int fileId)</code>
参数说明	Token:用户凭证 fileId:文件编号
返回值	Result=0 成功
版本	4.7.0 以上

1.6.5.2 根据类型名称删除指定文件元数据

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public int DeleteFileMetaByTypeNames(string token,int fileId,params string[] metaTypeNames)</code>
参数说明	Token:用户凭证 fileId:文件编号 metaTypeNames:元数据类型名称列表
返回值	Result=0 成功
版本	4.7.0 以上

1.6.5.3 根据类型编号删除指定文件元数据

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public int DeleteFileMetaByTypeIds(string token,int fileId,params string[] metaTypeIds)</code>
参数说明	Token:用户凭证 fileId:文件编号 metaTypeIds:元数据类型编号列表
返回值	Result=0 成功
版本	4.7.0 以上

1.6.5.4 获取文件元数据信息列表

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public ResultMeta GetFileMetaList(string token,int fileId)</code>
参数说明	Token:用户凭证 fileId:文件编号
返回值	Result=0 成功,EDocMetaObjectTypeInfo[]元数据列表
版本	4.7.0 以上

1.6.5.5 文件添加元数据信息

接口形式	代理类
------	-----

代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public int AddFileMetaByMetaTypeName(string token, int fileId, string metaTypeName, params MetaItemInfo[] items)</code>
参数说明	Token:用户凭证 fileId:文件编号 metaTypeName:元数据类型名称 MetaItemInfo[]:元数据信息列表
返回值	Result=0 成功
版本	4.7.0 以上

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public int AddFileMetaByMetaTypeId(string token, int fileId, int metaTypeId, params MetaItemInfo[] items)</code>
参数说明	Token:用户凭证 fileId:文件编号 metaTypeId:元数据类型编号 MetaItemInfo[]:元数据信息列表
返回值	Result=0 成功
版本	4.7.0 以上

1.6.5.6 根据元数据获取文件列表

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public EDocSearchResultInfoList GetFileListByMeta(string token, int objType, string[] attCode, string[] attValue)</code>
参数说明	Token:用户凭证 objType: 元数据类型 attCode []:属性 code attValue []:属性值
返回值	Result=0 成功 <code>List<EDocFileInfo></code> :文件列表
版本	4.8.0

1.6.5.7 根据元数据获取文件列表（支持所有元数据类型查询 4.8.1）

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public EDocSearchResultInfoList GetFileListByMetaComplex(string token, int typeId, List<string> attCode, List<string> attValue)</code>
参数说明	Token:用户凭证 typeId: 元数据类型 attCode []:属性 code attValue []:属性值

返回值	Result=0 成功 List<EDocFileInfo> :文件列表
版本	4.8.1

1.6.6 元数据表单操作

1.6.6.1 创建表单

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	public ResultInt CreateFormByTypeName(string token, int parentFolderId, string formName, string formCode, string formRemark, string metaTypeName)
参数说明	Token:用户凭证 parentFolderId:父文件夹编号 formName:表单名称 formCode:表单编码 formRemark:备注 metaTypeName:元数据类型名称
返回值	Result=0 成功,Result Value:表单编号
版本	4.7.0 以上

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	public ResultInt CreateFormByTypeCode(string token, int parentFolderId, string formName, string formCode, string formRemark, string metaTypeCode)
参数说明	Token:用户凭证 parentFolderId:父文件夹编号 formName:表单名称 formCode:表单编码 formRemark:备注 metaTypeCode:元数据类型编码
返回值	Result=0 成功,Result Value:表单编号
版本	4.7.0 以上

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	public ResultInt CreateFormById(string token, int parentFolderId, string formName, string formCode, string formRemark, int metaTypeId)
参数说明	Token:用户凭证 parentFolderId:父文件夹编号 formName:表单名称 formCode:表单编码 formRemark:备注 metaTypeId:元数据类型编号
返回值	Result=0 成功,Result Value:表单编号

版本	4.7.0 以上
----	----------

1.6.6.2 删除表单

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public int DeleteForm(string token,int formId)</code>
参数说明	Token:用户凭证 formId:表单编号
返回值	Result=0 成功
版本	4.7.0 以上

1.6.6.3 创建类别

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public ResultInt CreateFormCate(string token,int parentFormId,string formCateName)</code>
参数说明	Token:用户凭证 parentFormId:父表单编号 formCateName:表单类别名称
返回值	Result=0 成功,ResultValue:类别编号
版本	4.7.0 以上

1.6.6.4 删除类别

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public int DeleteFormCate(string token,int cateId)</code>
参数说明	Token:用户凭证 cateId:类别编号
返回值	Result=0 成功
版本	4.7.0 以上

1.6.6.5 创建表单记录

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public ResultInt CreateFormRecordByTypeName(string token,string typeName,int formId,params MetaItemInfo[] items)</code>
参数说明	Token:用户凭证 typeName:类型名称 formId:表单编号 MetaItemInfo[]:元数据信息列表
返回值	Result=0 成功
版本	4.7.0 以上

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public ResultInt CreateFormRecordByTypeCode(string token,string typeCode,int formId,params MetaItemInfo[] items)</code>
参数说明	Token:用户凭证 typeCode:类型编码 formId:表单编号 MetaItemInfo[]:元数据信息列表
返回值	Result=0 成功
版本	4.7.0 以上

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public ResultInt CreateFormRecordByTypeId(string token,string typeId,int formId,params MetaItemInfo[] items)</code>
参数说明	Token:用户凭证 typeId:类型编号 formId:表单编号 MetaItemInfo[]:元数据信息列表
返回值	Result=0 成功
版本	4.7.0 以上

1.6.6.6 更新表单记录

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public ResultInt UpdateRecordVersionByTypeId(string token,int typeId,int formId,params MetaItemInfo[] items)</code>
参数说明	Token:用户凭证 typeId:类型编号 formId:表单编号 MetaItemInfo[]:元数据信息列表
返回值	Result=0 成功
版本	4.7.0 以上

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public ResultInt UpdateRecordVersionByTypeName(string token,string typeName,int formId,params MetaItemInfo[] items)</code>
参数说明	Token:用户凭证 typeName:类型名称 formId:表单编号 MetaItemInfo[]:元数据信息列表
返回值	Result=0 成功

版本	4.7.0 以上
接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public ResultInt UpdateRecordVersionByTypeCode(string token,string typeCode,int formId,params MetaItemInfo[] items)</code>
参数说明	Token:用户凭证 typeCode:类型编码 formId:表单编号 MetaItemInfo[]:元数据信息列表
返回值	Result=0 成功
版本	4.7.0 以上

1.6.6.7 删除记录

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public int DeleteFormRecord(string token,int recordId)</code>
参数说明	Token:用户凭证 recordId:记录编号
返回值	Result=0 成功
版本	4.7.0 以上

1.6.6.8 获取表单记录

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public ResultEDocFormRecordInfo GetFormRecord(string token, int recordId)</code>
参数说明	Token:用户凭证 recordId:记录编号
返回值	Result=0 成功,EDocFormRecordInfo:表单记录
版本	4.7.0 以上

接口形式	代理类
代理类	EDoc2.ApiClientManage.Meta.EDoc2MetaApi
方法	<code>public ResultEDocFormOrCateInfo GetRecordListByFormOrCateId(string token, int pageSize, int pageNum, int formIdOrCateId)</code>
参数说明	Token:用户凭证 pageSize:页面大小 pageNum:当前页 formIdOrCateId:表单编号或类别编号
返回值	Result=0 成功,EDocFormRecordInfo:表单记录
版本	4.7.0 以上

1.7 搜索

1.7.1 简单文档查询

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.EDoc2DocumentApi
方法	<code>public ResultEDocSearchListInfo EasyDocSearch(string token, int folderId, string searchString, string argsXml, string isChildren)</code>
参数说明	token:用户凭证 folderId:文件夹编号 searchString:查询文本 argsXml:定制的参数 isChildren:是否查询子文件夹
返回值	Result=0 成功,EndNumber:结束数值,Info:查询结果
版本	4.7.0 以上、LiteV2

1.7.2 复杂文档查询

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.EDoc2DocumentApi
方法	<code>public ResultEDocSearchListInfo DocSearch(string token, int isSafeClient, string searchXml, int docViewId, string docViewStr, string argsXml, int startNum, int pageSize)</code>
参数说明	<p>token:用户凭证 isSafeClient:是否安全 searchXml:查询描述 docViewId:视图编号 docViewStr:视图名称 argsXml:定制的参数 startNum:开始数值 pageSize:页面大小</p> <p>searchXml 参考格式:</p> <pre><?xml version="1.0" encoding="utf-8"?> <Search> <SearchCondition> <SearchString><![CDATA[FileName:(中兴) FileContent:(中兴)]]></SearchString> <SearchType>QuickFile</SearchType> <SearchHistory>False</SearchHistory> <SearchSort>default</SearchSort> </SearchCondition> <conditions> <condition> <id>FileName</id> <type>0</type> <compare></compare></pre>

	<pre> <relation>0</relation> <value1><![CDATA[中兴]]></value1> <value2><![CDATA[]]></value2> </condition> <condition> <id>FileContent</id> <type>0</type> <compare></compare> <relation>0</relation> <value1><![CDATA[中兴]]></value1> <value2><![CDATA[]]></value2> </condition> </conditions> </Search> </pre>
返回值	Result=0 成功,EndNumber:结束数值,Info:查询结果
版本	4.7.0 以上、LiteV2

1.7.3 高级搜索中的基本(文件)搜索 4.8.1

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.EDoc2DocumentApi
方法	<code>public EDocSearchResultInfoList SearchFile(string token, EDocSearchFileWhereInfo fileWhereInfo)</code>
参数说明	<p>token:用户凭证</p> <p>EDocSearchFileWhereInfo fileWhereInfo 搜索条件 参数说明:</p> <p><code>int FileModifyOperator</code>修改人ID</p> <p><code>int FileCreateOperator</code>创建人ID</p> <p><code>string FileContent</code>文件包含的内容</p> <p><code>string FileName</code>文件名</p> <p><code>string FileTag</code>文件标签</p> <p><code>string FileCreateTime</code>创建时间 (格式:yyyy-MM-dd/yyyy-MM-dd 或 yyyy-MM-dd)</p> <p><code>string FileModifyTime</code>修改时间 (格式:yyyy-MM-dd/yyyy-MM-dd 或 yyyy-MM-dd)</p> <p><code>string FileExtName</code>文件后缀名</p> <p><code>int FilePath</code>所在文件夹ID</p> <p><code>bool SubFolder</code>包含子文件夹</p> <p><code>bool SearchHistory</code>包含历史版本</p>
返回值	<p>Result=0 成功</p> <p><code>List<EDocFileInfo></code>:文件列表</p>
版本	4.8.1、LiteV2

1.8 组织模块

1.8.1 组织操作

1.8.1.1 组织选择

接口形式	Url 地址
接口地址	EDoc2应用地址/ AppExt/Common/SelectOrgnization.aspx
参数说明	<p>userTree:成员树 deptTree:部门树 userGroupTree:用户组树</p> <p>以上三个参数的值传递的都是 json 格式字符串，包含以下三个属性 show:是否显示 true/false multiSelect:是否多选 true/false current:是否默认选中 true/false （这个属性对 deptTree 无效）</p>
示例	<p>显示成员树、部门树、用户组树，并且都是可多选，默认选中成员树</p> <p>EDoc2 应用地址/AppExt/Common/SelectOrgnization.aspx ?userTree={show:true,multiSelect:true,current:true} &deptTree={show:true,multiSelect:true} &userGroupTree={show:true,multiSelect:true}</p>
返回值	<p>例如用window.showModalDialog方式打开组织窗口，那么操作结果将通过 window.returnValue返回</p> <p>Js代码片段： <code>var res = window.showModalDialog(url, "", "dialogWidth:750px; dialogHeight:450px;");</code></p> <p>返回值解析： <code>res.users:Json数组，选中的成员或组织</code> <code>res.users [0]._data :{</code> <code>birthday:"生日",</code> <code>deptId:"部门Id",</code> <code>deptName:"部门名称",</code> <code>email:"邮箱",</code> <code>expiredTime:"过期时间",</code> <code>faxNumber:"传真号码",</code> <code>gender:"",</code> <code>id:"主键Id, 唯一标识",</code> <code>loginName:"登录名称",</code> <code>mobile:"手机",</code> <code>sortOrder:"排序值",</code> <code>telephone:"电话",</code> <code>userCode:"用户编码",</code> <code>userRealName:"真实名称",</code> <code>userState:"用户状态"}</code></p>

	<pre> res.depts:Json数组, 选中的部门 res.depts [0]._data :{ code:"编号", id:"主键Id, 唯一标识", level:"", name:"名称", parentId:"父级部门Id", parentName:"父级部门名称", positionId:"对应职位Id", remark:"备注", sortOrder:"排序值"} res.userGroups:Json数组, 选中的用户组 res.userGroups [0]._data :{ id:"主键Id, 唯一标识", isSystem:"是否系统内置用户组", name:"名称", parentId:"父级用户组Id", remark:"备注"} </pre>
版本	仅适用 ecm

1.8.2 用户操作

1.8.2.1 用户登录和注销操作

1.8.2.1.1 用户登录

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	<code>publicResultString UserLogin(string userLoginName,string password)</code>
参数说明	userLoginName:用户帐号 password:密码
返回值	Result=0 成功,ResultValue:Token 用户凭证
版本	4.7.0 以上、LiteV2

1.8.2.1.2 用户注销

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	<code>publicint UserLogout(string token)</code>
参数说明	Token:用户凭证
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.8.2.1.3 用户模拟登录

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	<code>publicResultString UserLoginImpersonateByUserId(int userId, string ipAddress)</code>
参数说明	userId:用户编号

	ipAddress:登录地址
返回值	Result=0 成功,ResultValue:Token 用户凭证
版本	4.7.0 以上、 LiteV2

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	<code>publicResultString UserLoginImpersonateByUserLoginName(string userLoginName, string ipAddress)</code>
参数说明	userLoginName:用户帐号 ipAddress:登录地址
返回值	Result=0 成功,ResultValue:Token 用户凭证
版本	4.7.0 以上、 LiteV2

1.8.2.1.4 用户模拟注销

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	<code>publicvoid UserLoginUndoImpersonate(string token)</code>
参数说明	Token:用户凭证
返回值	Result=0 成功
版本	4.7.0 以上、 LiteV2

1.8.2.2 获取用户信息

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	<code>publicResultEDocUserInfo GetUserInfoByToken(string token)</code>
参数说明	Token:用户凭证
返回值	Result=0 成功,EDocUserInfo:用户信息
版本	4.7.0 以上、 LiteV2

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	<code>publicResultEDocUserInfo GetUserInfoByLoginName(string token, string loginName)</code>
参数说明	Token:用户凭证 loginName:用户账号
返回值	Result=0 成功,EDocUserInfo:用户信息
版本	4.7.0 以上、 LiteV2

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	<code>publicResultEDocUserInfo GetUserInfoById(string token, int userId)</code>
参数说明	Token:用户凭证 userId:用户编号
返回值	Result=0 成功,EDocUserInfo:用户信息
版本	4.7.0 以上、 LiteV2

1.8.2.3 设置用户状态

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	public int ChangeUserStateById(string token, int userId, int state)
参数说明	Token:用户凭证 userId:用户编号 state:用户状态
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	public int ChangeUserStateByLoginName(string token, string loginName, int state)
参数说明	Token:用户凭证 loginName:用户账号 state:用户状态
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.8.2.4 创建用户

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	public Result EDocUserInfo CreateUser(string token, EDocUserInfo userInfo)
参数说明	Token:用户凭证 userInfo:用户信息
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	public int CreateUserByDeptId(string token, EDocUserInfo userInfo)
参数说明	Token:用户凭证 userInfo:用户信息
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	public int CreateUserByDeptCode(string token, string deptCode, EDocUserInfo userInfo)
参数说明	Token:用户凭证 userInfo:用户信息
返回值	Result=0 成功

版本	4.7.0 以上
----	----------

1.8.2.5 修改用户

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	public int UpdateUser(string token, EDocUserInfo userInfo)
参数说明	Token:用户凭证 userInfo:用户信息
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.8.2.6 删除用户

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	public int DeleteUserById(string token, int userId)
参数说明	Token:用户凭证 userId:用户编号
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	public int DeleteUserByLoginName(string token, string loginName)
参数说明	Token:用户凭证 loginName:用户帐号
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.8.2.7 批量删除用户

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	public int DeleteUserListById(string token, int [] userIds)
参数说明	Token:用户凭证 userIds:用户编号列表
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.8.2.8 用户修改密码

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	public int ChangeUserPassword (string token, string newPassword)
参数说明	Token:用户凭证

	newPassword:新密码
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	publicint ChangeUserPassword(string userLoginName, string newPassword)
参数说明	userLoginName:用户帐号 newPassword:新密码
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.8.2.9 是否登录

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	publicbool IsLogin(string token)
参数说明	Token:用户凭证
返回值	True False
版本	4.7.0 以上、LiteV2

1.8.2.10 根据用户编号获取此用户所在职位下所有子职位的人员信息(V1.2 4.8.0)

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	public ResultEDocUserInfoList GetChildUserListByUserId (string token, int userId)
参数说明	Token:用户凭证 userId: 用户编号
返回值	Result=0 成功, ResultEDocUserInfoList: 用户编号
版本	4.8.0 版本

1.8.3 部门操作

1.8.3.1 获取部门信息

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	publicResultEDocDepartmentInfo GetDepartmentInfoById(string token, int deptId)
参数说明	Token:用户凭证 deptId:部门编号
返回值	Result=0 成功,EDocDepartmentInfo:部门信息
版本	4.7.0 以上、LiteV2

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	publicResultEDocDepartmentInfo GetDepartmentInfoByCode(string token, string deptCode)

参数说明	Token:用户凭证 deptCode:部门编码
返回值	Result=0 成功,EDocDepartmentInfo:部门信息
版本	4.7.0 以上、LiteV2

1.8.3.2 获取子部门列表

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	publicResultEDocDepartmentInfoList GetChildDepartments(string token, int deptId)
参数说明	Token:用户凭证 deptId:部门编号
返回值	Result=0 成功,EDocDepartmentInfo[]:部门信息列表
版本	4.7.0 以上、LiteV2

1.8.3.3 获取顶级部门

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	publicResultEDocDepartmentInfo GetTopDepartmentInfo(string token)
参数说明	Token:用户凭证
返回值	Result=0 成功,EDocDepartmentInfo:部门信息
版本	4.7.0 以上、LiteV2

1.8.3.4 创建部门

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	publicResultEDocDepartmentInfo CreateDepartment(string token, EDocDepartmentInfo deptInfo)
参数说明	Token:用户凭证 deptInfo:部门信息
返回值	Result=0 成功,EDocDepartmentInfo:部门信息
版本	4.7.0 以上、LiteV2

1.8.3.5 更新部门

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	publicint UpdateDepartment(string token, EDocDepartmentInfo deptInfo)
参数说明	Token:用户凭证 deptInfo:部门信息
返回值	Result=0 成功,EDocDepartmentInfo:部门信息
版本	4.7.0 以上、LiteV2

1.8.3.6 删除部门

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	public int DeleteDepartmentByCode(string token, string deptCode)
参数说明	Token:用户凭证 deptCode:部门编码
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	public int DeleteDepartmentById(string token, int deptId)
参数说明	Token:用户凭证 deptId:部门编号
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.8.3.7 批量删除部门列表

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	public int DeleteDepartmentListById(string token, params int[] deptIds)
参数说明	Token:用户凭证 deptIds:部门编号列表
返回值	Result=0 成功 注:在删除部门的时候, 当前部门下不能存在用户
版本	4.7.0 以上、LiteV2

1.8.3.8 判断用户是否在指定的部门中(4.8.1)

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	public bool IsUserInDepartment(string userId, string departmentId, bool recursive)
参数说明	userId:用户凭证 departmentId:部门 ID recursive: 是否包含子部门
返回值	Result=True 成功
版本	4.8.1 版本

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	public bool IsUserInDepartment(int userIdentityId, int departmentIdentityId, bool recursive)
参数说明	userIdentityId:用户 IdentityId

	departmentIdentityId:部门 ID recursive: 是否包含子部门
返回值	Result=True 成功
版本	4.8.1 版本

1.8.4 职位操作

1.8.4.1 获取职位信息

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	publicResultEDocPositionInfo GetPositionInfoById(string token, int positionId)
参数说明	Token:用户凭证 positionId:职位编号
返回值	Result=0 成功,EDocPositionInfo:职位信息
版本	4.7.0 以上、LiteV2

1.8.4.2 获取顶级职位

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	publicResultEDocPositionInfo GetTopPositionInfo(string token)
参数说明	Token:用户凭证
返回值	Result=0 成功,EDocPositionInfo:职位信息
版本	4.7.0 以上、LiteV2

1.8.4.3 获取子职位列表

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	publicResultEDocPositionInfoList GetChildPositionList(string token, int positionId)
参数说明	Token:用户凭证 positionId:职位编号
返回值	Result=0 成功,EDocPositionInfo[]:职位信息列表
版本	4.7.0 以上、LiteV2

1.8.4.4 创建职位

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	publicResultEDocPositionInfo CreatePosition(string token, EDocPositionInfo positionInfo)
参数说明	Token:用户凭证 positionInfo:职位信息
返回值	Result=0 成功,EDocPositionInfo:职位信息
版本	4.7.0 以上、LiteV2

1.8.4.5 修改职位

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	publicint UpdatePosition(string token, EDocPositionInfo positionInfo)
参数说明	Token:用户凭证 positionInfo:职位信息
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.8.4.6 删除职位

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	publicint DeletePositionById(string token, int positionId)
参数说明	Token:用户凭证 positionId:职位编号
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.8.4.7 批量删除职位

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	publicint DeletePositionListById(string token, paramsint[] positionIds)
参数说明	Token:用户凭证 positionIds:职位编号列表
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.8.4.8 添加用户插入到职位

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	publicint AddUsersIntoPosition(string token, int positionId, paramsint[] userIds)
参数说明	Token:用户凭证 positionId:职位编号 userIds:用户编号列表
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.8.4.9 移除用户所在的职位

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	publicint RemovePositionListByUser(string token, int userId)

参数说明	Token:用户凭证 userId:用户编号
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.8.4.10 根据用户编号获取所在职位列表(4.8.0)

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	<code>public ResultEDocPositionInfoList GetChildPositionListByUserId(string token, int userId)</code>
参数说明	Token:用户凭证 userId:用户编号
返回值	Result=0 成功,EDocPositionInfo[]:职位信息列表
版本	4.8.0 版本

1.8.4.11 添加用户到职位中(4.8.0)

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	<code>public int AddUserToPosition(string token, int userId, int positionId, bool isMain)</code>
参数说明	Token:用户凭证 userId:用户编号 positionId:职位编号 isMain:是否主职
返回值	Result=0 成功
版本	4.8.0 版本

1.8.4.12 获取用户的主职位信息(4.8.1)

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	<code>public ResultEDocPositionInfo GetMainPositionOfUser(string userId)</code>
参数说明	userId:用户编号
返回值	Result=True 成功
版本	4.8.1 版本

1.8.4.13 判断用户是否在指定的职位下(4.8.1)

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	<code>public bool IsUserInPosition(string userId, string positionId, bool recursive)</code>
参数说明	userId:用户编号 positionId: 职位 ID recursive: 是否包含子职位
返回值	Result=True 成功

版本	4.8.1 版本
接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	<code>public bool IsUserInPosition(int userIdentityId, int positionIdentityId, bool recursive)</code>
参数说明	userIdentityId: 用户编号 positionIdentityId: 职位 ID recursive: 是否包含子职位
返回值	Result=True 成功
版本	4.8.1 版本

1.8.5 工作组操作

1.8.5.1 获取工作组

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	<code>public ResultEDocGroupInfoList GetTopGroups(string token)</code>
参数说明	Token: 用户凭证
返回值	Result=0 成功, EDocGroupInfo: 工作组信息
版本	4.7.0 以上、LiteV2

1.8.5.2 获取顶级工作组

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	<code>public ResultEDocGroupInfo GetGroupInfoById(string token, int groupId)</code>
参数说明	Token: 用户凭证 groupId: 工作组编号
返回值	Result=0 成功, EDocGroupInfo[]: 工作组信息列表
版本	4.7.0 以上、LiteV2

1.8.5.3 获取用户的工作组列表

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	<code>public ResultEDocGroupInfoList GetGroupsOfUser(string token, int userId)</code>
参数说明	Token: 用户凭证 userId: 用户编号
返回值	Result=0 成功, ResultValue: 工作组编号
版本	4.7.0 以上、LiteV2

1.8.5.4 获取工作组的用户列表

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi

方法	<code>publicResultEDocUserInfoList GetChildUserListInGroup(string token,int groupId)</code>
参数说明	Token:用户凭证 groupId:工作组编号
返回值	Result=0 成功,UserInfoList:用户列表信息
版本	4.7.0 以上、LiteV2

1.8.5.5 创建工作组

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	<code>publicResultInt CreateGroup(string token,int parentGroupId,string name,string remark)</code>
参数说明	Token:用户凭证 parentGroupId:父工作组编号 name:工作组名称 remark:备注
返回值	Result=0 成功,ResultValue:工作组编号
版本	4.7.0 以上、LiteV2

1.8.5.6 修改工作组

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	<code>publicint ChangeGroup(string token, int groupId, string name, string remark, int parentGroupId)</code>
参数说明	Token:用户凭证 groupId:工作组编号 name:工作组名称 remark:备注 parentGroupId:父工作组编号
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.8.5.7 删除工作组

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	<code>publicint DeleteGroup(string token,int groupId)</code>
参数说明	Token:用户凭证 groupId:工作组编号
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.8.5.8 添加用户或工作组

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi

方法	<code>public int AddUsersOrGroupIntoGroup(string token, int groupId, List<int> userIds, List<int> groupIds)</code>
参数说明	Token:用户凭证 groupId:工作组编号 userIds:用户编号列表 groupIds:工作组编号列表
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.8.5.9 移除用户或工作组

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	<code>public int RemoveUsersOrGroupFromGroup(string token, int groupId, List<int> userIds, List<int> groupIds)</code>
参数说明	Token:用户凭证 groupId:工作组编号 userIds:用户编号列表 groupIds:工作组编号列表
返回值	Result=0 成功
版本	4.7.0 以上、LiteV2

1.8.5.10 判断用户是否在指定的用户组中(4.8.1)

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	<code>public bool IsUserInUserGroup(string userId, string groupId, bool recursive)</code>
参数说明	userId:用户编号 groupId:工作组编号 recursive: 是否包含嵌套用户组
返回值	Result=True 成功
版本	4.8.1 版本

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	<code>public bool IsUserInUserGroup(int userIdentityId, int groupId, bool recursive)</code>
参数说明	userIdentityId:用户编号 groupId:工作组编号 recursive: 是否包含嵌套用户组
返回值	Result=True 成功
版本	4.8.1 版本

1.8.5.11 查询系统用户组 Id(4.8.1)

接口形式	代理类
------	-----

代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	<code>public string GetAdministratorGroupId()</code>
参数说明	
返回值	Result=用户组 Id 成功
版本	4.8.1 版本

1.8.6 获取用户功能权限列表

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	<code>public ResultEDocUserFunctionInfoList GetFunPermAll(string token)</code>
参数说明	Token:用户凭证
返回值	Result=0 成功,EDocUserFunctionInfo[]:用户功能权限列表
版本	4.7.0 以上、LiteV2

接口形式	代理类
代理类	EDoc2.ApiClientManage.Organization.EDoc2OrganizationApi
方法	<code>public ResultEDocUserFunctionInfoList GetUserFunctionByUserId(string token,int userId)</code>
参数说明	Token:用户凭证 userId:用户编号
返回值	Result=0 成功,EDocUserFunctionInfo[]:用户功能权限列表
版本	4.8.0 以下版本

1.8.7 从外部进入 edoc2 主页面

Edoc2应用地址/default.aspx ?tkn=[tkn]

1.9 日志操作

1.9.1 查询操作日志(4.8.1)

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public ResultEDocLogOperationInfoList GetLogOperationByCondition(string token, EDocLogConditionInfo logConditionInfo)</code>
参数说明	<p>token: 用户凭证</p> <p>logConditionInfo: 搜索条件集合</p> <p>EDocLogConditionInfo logConditionInfo对象属性说明</p> <p>int UserId操作人ID</p> <p>string UserRealName操作人ID</p> <p>string OptType操作类型</p> <p>DateTime OptTimeStart开始时间</p> <p>DateTime OptTimeEnd结束时间</p>

	<code>int</code> PageNum总页数 <code>int</code> PageSize每页数量 <code>int</code> DestId目标编号 <code>string</code> DestName目标名称 <code>int</code> OptId操作ID <code>int</code> SourceId操作源对象ID <code>string</code> SourceName操作源对象名称 <code>int</code> OptSourceParentId操作源父对象ID <code>string</code> OptSourceParentName操作源父对象名称
返回值	Result=0 成功, PageCount: 总页数 TotalCount: 总记录数 <code>LogOperationInfo []</code> : 操作日志信息列表
版本	4.8.1

1.10 知识地图操作(仅适用 ECM)

1.10.1 根据父节点 ID 获取所有子节点(4.8.1)

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public ResultEDocMapNodeEntityList GetMapChildNodes(string token, int mnParentId)</code>
参数说明	token: 用户凭证 mnParentId: 父节点ID
返回值	Result=0 成功, <code>EDocMapNodeEntity[]</code> : 节点信息列表
版本	4.8.1

1.10.2 获取当前用户的所有节点(4.8.1)

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public ResultEDocMapNodeEntityList GetMapNodeByCurrentUser(string token)</code>
参数说明	token: 用户凭证
返回值	Result=0 成功, <code>EDocMapNodeEntity[]</code> : 节点信息列表
版本	4.8.1

1.10.3 获取顶层节点(4.8.1)

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public ResultEDocMapNodeEntityList GetMapTopNode(string token)</code>
参数说明	token: 用户凭证
返回值	Result=0 成功,

	EDocMapNodeEntity[]: 节点信息列表
版本	4.8.1

1.10.4 根据节点 ID 获取节点信息(4.8.1)

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	public ResultEDocMapNodeEntity GetMapNodeById(string token, int mnId)
参数说明	token: 用户凭证 mnId: 节点ID
返回值	Result=0 成功, EDocMapNodeEntity: 节点信息
版本	4.8.1

1.10.5 根据节点 ID 获取节点路径(4.8.1)

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	public ResultString GetMapNodePathById(string token, int mnId)
参数说明	token: 用户凭证 mnId: 节点ID
返回值	Result=0 成功, ResultValue: 节点路径
版本	4.8.1

1.11 系统管理操作

1.11.1 系统基本设置(设置实例配置信息)4.8.1

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	public int SetInstanceConfig(string token, EDocInstanceConfigInfo configuration)
参数说明	token: 用户凭证 configuration: 实例配置信息对象 EDocInstanceConfigInfo configuration 对象属性说明 string DefaultPreviewMode 系统默认的预览方式 string ArchiveDbServerName 归档数据库服务器名 string ArchiveDbName 归档数据库名称 string ArchiveDbUser 归档数据库登录名 string ArchiveDbPwd 归档数据库密码 string Language 默认语言 string TimeZone 默认时区 string Theme 默认主题 int DateFormat 默认日期格式 int TimeFormat 默认时间格式

int PortalStartPageId Portal起始页ID
 int DocViewMode默认视图
 int DocListPageSize列表视图页大小
 int DocSummaryPageSize摘要视图页大小
 int DocIconPageSize图标视图页大小
 int SearchPageSize搜索页大小
 bool ShowFolderTree是否显示文件夹树
 int DefaultPublishDay默认发布天数
 string DefaultUserPassword默认密码
 string SmtServer SMTP服务器地址
 string SmtEmail邮件地址
 string SmtUserName邮件用户名
 string SmtPassword邮件密码
 int SmtPort邮件发送端口号
 bool SmtUseSSL邮件发送使用SSL加密
 bool EachFileRelate是否相互关联
 bool ListNonePermItem显示没有权限的项目
 bool SearchNonePermItem在搜索结果中显示没有权限的项目
 string LoginValidateType登录验证的类型
 string LoginADValidateAddress域验证地址
 string LoginADValidateAuto是否自动登录
 string LoginADDomainName登录域名称
 string LoginValidateImage登录验证图片
 bool EnableFolderMetaData是否启用文件夹元数据
 int SessionTimeOut会话过期时间（分钟）
 string PublishExternalAddress外部地址
 string MessageServerAddress消息推送地址
 string FolderSortOrder获取文件夹树排序
 bool FolderSortDesc是否文件夹倒序排列，默认false，升序
 string DepartmentSortOrder部门排序
 bool DepartmentSortDesc部门是否倒序排列，默认false，升序
 bool ReLoginStrategy重复登录策略
 string VdiverVersion vdriver版本
 string WokflowType流程类型
 string WfUltDomainName ultimus流程的域名
 string WfShowSameIncident是否显示相同流程实例的不同步骤
 string DefaultTab默认页面导航Tab
 bool EnableMultiTargetUpload是否支持多目标上传
 bool SecEncryptNewAutoCAD是否加密新建的AutoCAD文件
 bool SecEncryptAllLocalAutoCAD自动加密本地所有AutoCAD文件
 bool IsAllowAnonymityComment是否允许匿名评论
 string ExtJsFiles扩展Js文件
 string EnabledPrinterModels启用打印机模型
 string EnabledPrintScreen屏幕截图
 bool GrayFileCanNotPreview将不能预览的文件置灰

	<p><code>bool</code> ProhibitUserPurgeFile禁止普通用户清除文件</p> <p><code>string</code> ContentSafeEngine启用内容安全引擎</p> <p><code>bool</code> EnableCoverVersion是否允许覆盖当前版本</p> <p><code>bool</code> LoginEnableIPFilter登录IP过滤</p> <p><code>string</code> LoginEnableIPRange登录IP范围</p> <p><code>string</code> LoginProhibitedIPRange禁止登录的IP范围</p> <p><code>bool</code> DelAttachmentIntoRecycle是否将附件删除到回收站</p> <p><code>int</code> SendMailServerType发邮件服务器类型</p> <p><code>bool</code> EnabledDownloadDesc是否启用下载填写备注功能</p> <p><code>int</code> VprinterFldId虚拟打印上传文件夹ID</p> <p><code>int</code> FaxReceiveFolderId传真接收文件夹</p> <p><code>string</code> FaxExts支持传真的文档类型</p> <p><code>string</code> FaxMetaAttrCustomerName存放传真客户的元数据属性</p> <p><code>bool</code> NewWindowForFullScreen在新窗口中打开全屏预览</p> <p><code>int</code> OcrType OCR类型</p> <p><code>bool</code> EnablePermSecLevel启用权限密级</p> <p><code>string</code> PortalVerion门户版面</p> <p><code>int</code> SearchType搜索类型</p> <p><code>string</code> SearchTrsHostName主机地址</p> <p><code>string</code> SearchTrsHostPort端口</p> <p><code>string</code> SearchTrsUserName用户名</p> <p><code>string</code> SearchTrsUserPassword密码</p> <p><code>bool</code> ProhibitEmailFileAttachment禁止以邮件附件方式发送文件</p> <p><code>int</code> MailSendWay邮件发送方式</p> <p><code>string</code> LoginLDAPServerType LDAP服务器类型</p>
返回值	Result=0 成功,
版本	4.8.1

1.11.2 系统基本设置(获取实例配置信息)4.8.1

接口形式	代理类
代理类	EDoc2.ApiClientManage.Document.DocumentApiProvider
方法	<code>public ResultEDocInstanceConfigInfo GetInstanceConfig(string token)</code>
参数说明	token: 用户凭证
返回值	<p>Result=0成功,</p> <p>configuration: 实例配置信息对象</p> <p>EDocInstanceConfigInfo configuration 对象属性说明</p> <p><code>string</code> DefaultPreviewMode系统默认的预览方式</p> <p><code>string</code> ArchiveDbServerName归档数据库服务器名</p> <p><code>string</code> ArchiveDbName归档数据库名称</p> <p><code>string</code> ArchiveDbUser归档数据库登录名</p> <p><code>string</code> ArchiveDbPwd归档数据库密码</p> <p><code>string</code> Language默认语言</p> <p><code>string</code> TimeZone默认时区</p>

string Theme默认主题
 int DateFormat默认日期格式
 int TimeFormat默认时间格式
 int PortalStartPageId Portal起始页ID
 int DocViewMode默认视图
 int DocListPageSize列表视图页大小
 int DocSummaryPageSize摘要视图页大小
 int DocIconPageSize图标视图页大小
 int SearchPageSize搜索页大小
 bool ShowFolderTree是否显示文件夹树
 int DefaultPublishDay默认发布天数
 string DefaultUserPassword默认密码
 string SmtServer SMTP服务器地址
 string SmtEmail邮件地址
 string SmtUserName邮件用户名
 string SmtPassword邮件密码
 int SmtPort邮件发送端口号
 bool SmtUseSSL邮件发送使用SSL加密
 bool EachFileRelate是否相互关联
 bool ListNonePermItem显示没有权限的项目
 bool SearchNonePermItem在搜索结果中显示没有权限的项目
 string LoginValidateType登录验证的类型
 string LoginADValidateAddress域验证地址
 string LoginADValidateAuto是否自动登录
 string LoginADDomainName登录域名称
 string LoginValidateImage登录验证图片
 bool EnableFolderMetaData是否启用文件夹元数据
 int SessionTimeOut会话过期时间（分钟）
 string PublishExternalAddress外部地址
 string MessageServerAddress消息推送地址
 string FolderSortOrder获取文件夹树排序
 bool FolderSortDesc是否文件夹倒序排列，默认false，升序
 string DepartmentSortOrder部门排序
 bool DepartmentSortDesc部门是否倒序排列，默认false，升序
 bool ReloginStrategy重复登录策略
 string VdverVersion vdriver版本
 string WokflowType流程类型
 string WfUltDomainName ultimus流程的域名
 string WfShowSameIncident是否显示相同流程实例的不同步骤
 string DefaultTab默认页面导航Tab
 bool EnableMultiTargetUpload是否支持多目标上传
 bool SecEncryptNewAutoCAD是否加密新建的AutoCAD文件
 bool SecEncryptAllLocalAutoCAD自动加密本地所有AutoCAD文件
 bool IsAllowAnonymityComment是否允许匿名评论
 string ExtJsFiles扩展Js文件

	<p><code>string EnabledPrinterModels</code> 启用打印机模型</p> <p><code>string EnabledPrintScreen</code> 屏幕截图</p> <p><code>bool GrayFileCanNotPreview</code> 将不能预览的文件置灰</p> <p><code>bool ProhibitUserPurgeFile</code> 禁止普通用户清除文件</p> <p><code>string ContentSafeEngine</code> 启用内容安全引擎</p> <p><code>bool EnableCoverVersion</code> 是否允许覆盖当前版本</p> <p><code>bool LoginEnableIPFilter</code> 登录IP过滤</p> <p><code>string LoginEnableIPRange</code> 登录IP范围</p> <p><code>string LoginProhibitedIPRange</code> 禁止登录的IP范围</p> <p><code>bool DelAttachmentIntoRecycle</code> 是否将附件删除到回收站</p> <p><code>int SendMailServerType</code> 发邮件服务器类型</p> <p><code>bool EnabledDownloadDesc</code> 是否启用下载填写备注功能</p> <p><code>int VprinterFldId</code> 虚拟打印上传文件夹ID</p> <p><code>int FaxReceiveFolderId</code> 传真接收文件夹</p> <p><code>string FaxExts</code> 支持传真的文档类型</p> <p><code>string FaxMetaAttrCustomerName</code> 存放传真客户的元数据属性</p> <p><code>bool NewWindowForFullScreen</code> 在新窗口中打开全屏预览</p> <p><code>int OcrType</code> OCR类型</p> <p><code>bool EnablePermSecLevel</code> 启用权限密级</p> <p><code>string PortalVerion</code> 门户版面</p> <p><code>int SearchType</code> 搜索类型</p> <p><code>string SearchTrsHostName</code> 主机地址</p> <p><code>string SearchTrsHostPort</code> 端口</p> <p><code>string SearchTrsUserName</code> 用户名</p> <p><code>string SearchTrsUserPassword</code> 密码</p> <p><code>bool ProhibitEmailFileAttachment</code> 禁止以邮件附件方式发送文件</p> <p><code>int MailSendWay</code> 邮件发送方式</p> <p><code>string LoginLDAPServerType</code> LDAP 服务器类型</p>
版本	4.8.1、LiteV2

2 edoc2 业务组件(仅适用 ECM)

2.1 组织选择

接口形式	页面接口
接口地址	EDoc2应用地址/ AppExt/Common/SelectOrganization.aspx
参数说明	<p><code>userTree</code>:成员树</p> <p><code>deptTree</code>:部门树</p> <p><code>userGroupTree</code>:用户组树</p> <p>以上三个参数的值传递的都是 json 格式字符串, 包含以下三个属性</p> <p><code>show</code>:是否显示 true/false</p> <p><code>multiSelect</code>:是否多选 true/false</p> <p><code>current</code>:是否默认选中 true/false (这个属性对 <code>deptTree</code> 无效)</p>

示例	显示成员树、部门树、用户组树，并且都是可多选，默认选中成员树 EDoc2 应用地址/edoc2v4/AppExt/Common/SelectOrganization.aspx?userTree={show:true,multiSelect:true,current:true}&deptTree={show:true,multiSelect:true}&userGroupTree={show:true,multiSelect:true}
返回值	例如用window.showModalDialog方式打开组织窗口，那么操作结果将通过window.returnValue返回 Js代码片段： <code>var res = window.showModalDialog(url, "", "dialogWidth:750px; dialogHeight:450px;");</code> 返回值解析： res.users:Json数组，选中的成员或组织 res.users [0]._data :{ birthday:"生日", deptId:"部门Id", deptName:"部门名称", email:"邮箱", expiredTime:"过期时间", faxNumber:"传真号码", gender:"", id:"主键Id, 唯一标识", loginName:"登录名称", mobile:"手机", sortOrder:"排序值", telephone:"电话", userCode:"用户编码", userRealName:"真实名称", userState:"用户状态"} res.depts:Json数组，选中的部门 res. depts [0]._data :{ code:"编号", id:"主键Id, 唯一标识", level:"", name:"名称", parentId:"父级部门Id", parentName:" 父级部门名称", positionId:" 对应职位Id", remark:" 备注", sortOrder:" 排序值"} res. userGroups:Json数组，选中的用户组 res. userGroups [0]._data :{ id:"主键Id, 唯一标识",

	<code>isSystem:</code> ”是否系统内置用户组”, <code>name:</code> ”名称”, <code>parentId:</code> ”父级用户组Id”, <code>remark:</code> ”备注”}
--	--

2.2 文档选择

接口形式	页面接口
接口地址	EDoc2应用地址/ AppExt/Common/SelectEDocFile.aspx
参数说明	目前发布的接口不需要跟任何参数，文档可多选
示例	
返回值	<p>例如用window.showModalDialog方式打开选择文档窗口，那么操作结果将通过window.returnValue返回</p> <p>Js代码片段：</p> <pre>var res = window.showModalDialog(url, "", "dialogWidth:750px; dialogHeight:450px;");</pre> <p>返回值解析：</p> <p>res:Json数组，选中的文档</p> <pre>res [0]._data :{ curVerId:"当前版本Id", curVerNumStr:"当前版本编号", id:"主键Id, 唯一标识", metaTypeId:"元数据类型Id", metaTypeName:"元数据类型名称", name:"文档名称", path:"文档完全路径", size:"文档大小" }</pre>

2.3 文件夹选择

接口形式	页面接口
接口地址	EDoc2应用地址/ AppExt/Common/SelectEDoc2Folder.aspx
参数说明	目前发布的接口不需要跟任何参数，文件夹只能单选
示例	
返回值	<p>例如用window.showModalDialog方式打开选择文件夹窗口，那么操作结果将通过window.returnValue返回</p> <p>Js代码片段：</p> <pre>var res = window.showModalDialog(url, "", "dialogWidth:750px; dialogHeight:450px;");</pre> <p>返回值解析：</p> <p>res:Json数组，选中的文件夹</p> <pre>res [0]._data :{ alertSize:""</pre>

	<p>childFileCount:”该文件夹下文档个数”,</p> <p>childFolderCount:”该文件夹下文件夹个数”,</p> <p>code:”编号”,</p> <p>createTime:”创建时间”,</p> <p>creatorId:”创建人Id”,</p> <p>creatorName:”创建人名称”,</p> <p>enabledFolderTypeSet:””,</p> <p>folderType:””,</p> <p>forbiddenFileExtensions:”禁止上传的文档后缀名”,</p> <p>id:” 主键Id, 唯一标识”,</p> <p>maxFileSize:”允许上传的最大文档大小”,</p> <p>maxFolderSize:””,</p> <p>modifyTime:”更新时间”,</p> <p>name:”名称”,</p> <p>parentFolderId:”父文件夹Id”,</p> <p>parentFolderName:”父文件夹名称”,</p> <p>path:”文件夹路径”,</p> <p>permission:””,</p> <p>remark:”备注”,</p> <p>securityLevel:””,</p> <p>size:”大小”,</p> <p>state:”状态”,</p> <p>type:”类型” }</p>
--	---

2.4 文件上传

接口形式	页面接口
接口地址	EDoc2应用地址/ Document/File UploadEx.aspx
参数说明	<p>folderId:目标文件夹 Id</p> <p>multifile:是否上传多个</p> <p>tkn:有该文件夹上传权限用户的 Token (用户登录且未跨域情况下非必传)</p> <p>key=upload: 该参数是为了避开 EDoc2 的权限管控加的参数, 用到时须在 EDoc2V4\Website\Web.config 中添加 uploadUserId 节点, 并赋予权限用户 Id, 如请求地址后加了此参数, 那么该上传接口页面会从 EDoc2V4\Website\Web.config 中读取 uploadUserId 配置的权限用户来模拟登录上传 (偶尔会造成当前登录用户被踢的现象, 不建议使用)</p>
示例	<p>上传到文件夹 Id 为 20 的文件夹, 允许多个上传</p> <p>EDoc2 应用地址/Document/File UploadEx.aspx?folderid=20&multifile=true</p> <p>需注意的是调用上传接口后的回调方法在 EDoc2V4.8.0 和 4.8.0 以下版本写法不一样</p> <p>EDoc2 V4.8.0 以下版本:</p>

	<pre><script type="text/javascript"> var edocUrl = "http://localhost/edoc2v4/"; function showPage() { var url = edocUrl + "Document/File_UploadEx.aspx?folderid=20&multifile=true"; var win = openUploadWnd(url, 480, 360); //指定上传完毕的回调函数 win.uploadCallback = function (filelist) { debugger; var dataRows; //遍历上传成功的附件信息 for (var i = 0; i < filelist.length; i++) { } win.close(); }; } function openUploadWnd(url, width, height) { return window.open(url, "_blank", "height=" + height + "px, width=" + width + "px"); } </script></pre> <p>EDoc2 V4.8.0 版本:</p> <pre><script type="text/javascript"> var edocUrl = "http://localhost/edoc2v4/"; var win; function showPage() { var url = edocUrl + "Document/File_UploadEx.aspx?folderid=20&multifile=true"; win = openUploadWnd(url, 480, 360); } //指定上传完毕的回调函数 win.uploadCallback = function (filelist) { debugger; var dataRows; //遍历上传成功的附件信息 for (var i = 0; i < filelist.length; i++) { } win.close(); }; function openUploadWnd(url, width, height) { return window.open(url, "_blank", "height=" + height + "px, width=" + width + "px"); } </script></pre>
返回值	<p>例如用window.open方式打开上传附件窗口，那么文件上传完成后将调用uploadCallback回调函数，并且将上传结果作为参数传入该回调方法</p> <p>Js代码片段：</p> <pre>win.uploadCallback = function (filelist)</pre> <p>返回值解析：</p> <p>filelist:Json数组，上传成功后的文件</p> <pre>filelist [0]:{ attachType:"", fileId:"文件Id", fileName:"名称", folderId:"所属文件夹id", wfResult:"流程状态"}</pre>

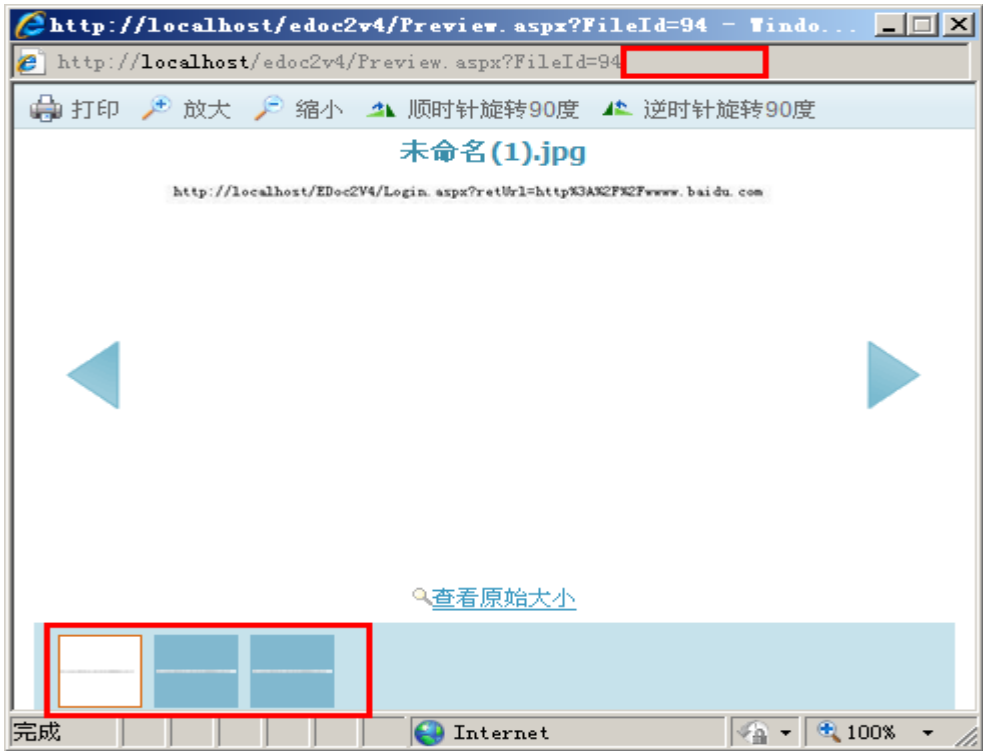
2.5 文件下载

接口形式	页面接口
接口地址	EDoc2应用地址/Document/File_Download.aspx
参数说明	file_Id:文件 Id ver_id:文件版本 Id

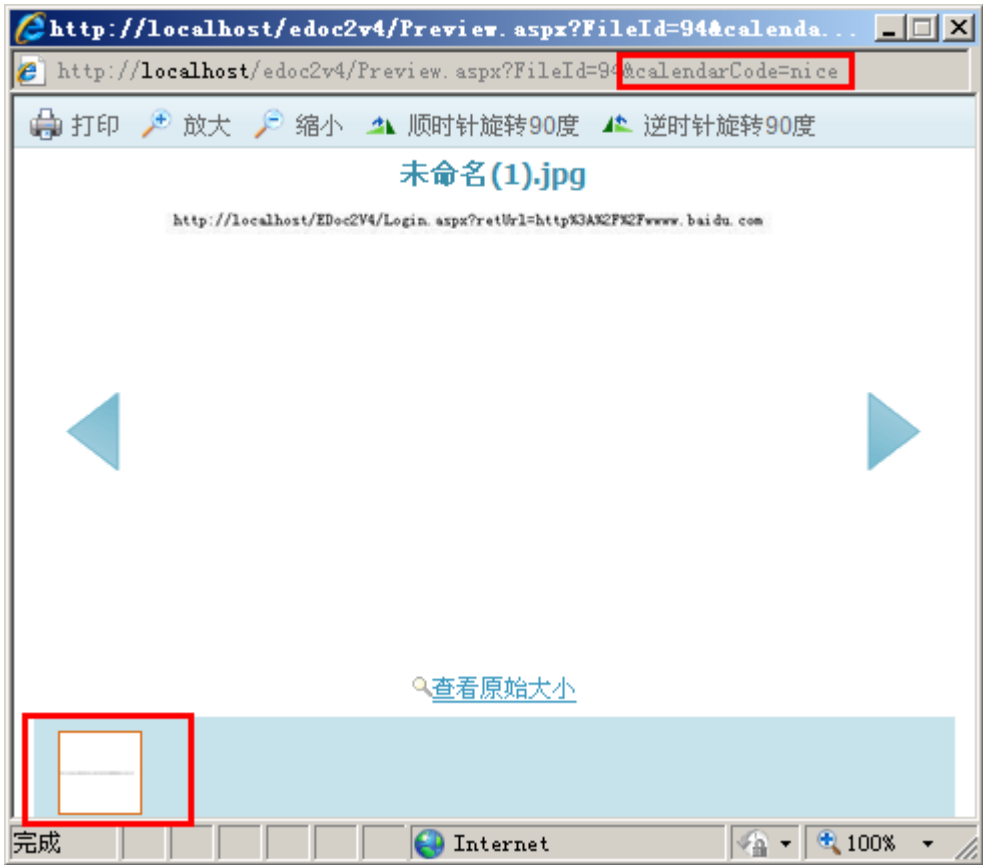
	<p>tkn:有该文件夹上传权限用户的 Token (用户登录且未跨域情况下非必传)</p> <p>key=upload: 该参数是为了避开 EDoc2 的权限管控加的参数, 用到时须在 EDoc2V4\Website\Web.config 中添加 uploadUserId 节点, 并赋予权限用户 Id, 如请求地址后加了此参数, 那么该上传接口页面会从 EDoc2V4\Website\Web.config 中读取 uploadUserId 配置的权限用户来模拟登录上传 (偶尔会造成当前登录用户被踢的现象, 不建议使用)</p> <p>file_id 和 ver_id 两个参数不同时出现, 要么下载指定文件, 要么下载文件的指定版本</p>
示例	<p>下载 Id 为 201 的文件的请求地址:</p> <p>EDoc2应用地址/ Document/File_Download.aspx?file_Id=201</p> <p>如 201 文件上一版本的版本 Id 为 198, 那么下载该版本文件请求地址为:</p> <p>EDoc2应用地址/ Document/File_Download.aspx?ver_id=198</p>
返回值	

2.6 文件预览

接口形式	页面接口
接口地址	EDoc2应用地址/Preview.aspx
参数说明	<p>FileId:文件 Id</p> <p>FileVerId:文件版本 Id</p> <p>calendarCode:当预览文件为图片时, 如果不跟此参数, 该接口默认会将同目录下的所有图片都能预览出来(如下图例 1), 当跟了这个参数并且赋予任意值则只能预览指定的文件(如下图例 2)</p> <p>tkn:有该文件夹上传权限用户的 Token (用户登录且未跨域情况下非必传)</p> <p>key=upload: 该参数是为了避开 EDoc2 的权限管控加的参数, 用到时须在 EDoc2V4\Website\Web.config 中添加 uploadUserId 节点, 并赋予权限用户 Id, 如请求地址后加了此参数, 那么该上传接口页面会从 EDoc2V4\Website\Web.config 中读取 uploadUserId 配置的权限用户来模拟登录上传 (偶尔会造成当前登录用户被踢的现象, 不建议使用)</p> <p>FileId 和 FileVerId 两个参数不同时出现, 要么预览指定文件, 要么预览文件的指定版本</p>
示例	<p>预览 Id 为 201 的文件的请求地址:</p> <p>EDoc2应用地址/ Preview.aspx? FileId =201&calendarCode=1</p> <p>如 201 文件上一版本的版本 Id 为 198, 那么预览该版本文件请求地址为:</p> <p>EDoc2应用地址/ Preview.aspx? FileVerId =198&calendarCode=1</p>
返回值	



图例 1



图例 2

2.7 搜索接口

接口形式	页面接口
------	------

接口地址	EDoc2应用地址/Preview.aspx
参数说明	<p>FileId: 文件 Id</p> <p>FileVerId: 文件版本 Id</p> <p>calendarCode: 当预览文件为图片时, 如果不跟此参数, 该接口默认会将同目录下的所有图片都能预览出来(如下图例 1), 当跟了这个参数并且赋予任意值则只能预览指定的文件(如下图例 2)</p> <p>tkn: 有该文件夹上传权限用户的 Token (用户登录且未跨域情况下非必传)</p> <p>key=upload: 该参数是为了避开 EDoc2 的权限管控加的参数, 用到时须在 EDoc2V4\Website\Web.config 中添加 uploadUserId 节点, 并赋予权限用户 Id, 如请求地址后加了此参数, 那么该上传接口页面会从 EDoc2V4\Website\Web.config 中读取 uploadUserId 配置的权限用户来模拟登录上传 (偶尔会造成当前登录用户被踢的现象, 不建议使用)</p> <p>FileId 和 FileVerId 两个参数不同时出现, 要么预览指定文件, 要么预览文件的指定版本</p>
示例	<p>预览 Id 为 201 的文件的请求地址:</p> <p>EDoc2应用地址/Preview.aspx?FileId=201&calendarCode=1</p> <p>如 201 文件上一版本的版本 Id 为 198, 那么预览该版本文件请求地址为:</p> <p>EDoc2应用地址/Preview.aspx?FileVerId=198&calendarCode=1</p>
返回值	

2.8 文件选择

接口形式	Js 组件
接口地址	
参数说明	<p>url: //webapi 服务地址</p> <p>token: "", //用户凭据</p> <p>topFolderIds: "", //指定显示文件夹项Id</p> <p>showSelectArea: true, //是否显示选择区</p> <p>multiple: true, //是否多选</p> <p>showPersonalFolder: false, //是否显示个人文件夹</p> <p>selectedFileIds: "", //默认选择文件项Id</p> <p>displayFolderIds: "", //指定显示文件夹项Id</p> <p>onlyShowFileType: "", //只显示文件类型</p> <p>noneShowFileType: "", //不显示文件类型</p> <p>onSelectedFileCallback: function (data) { } // dialog确定回调函数</p>
示例	<p>1、在使用组件的页面引用以下 js 和 css</p> <pre> <link rel="stylesheet" type="text/css" href="js/bootstrap/css/bootstrap.css"> <link rel="stylesheet" type="text/css" href="js/easyui/themes/bootstrap/easyui.css"> <link rel="stylesheet" type="text/css" href="js/select2/select2.css"> <link rel="stylesheet" type="text/css" href="js/easyui/themes/icon.css"> <link rel="stylesheet" type="text/css" href="css/fileSelect.css" /> <script type="text/javascript" src="js/jquery-1.8.3.js"></script> <script type="text/javascript" src="js/jquery.json-2.4.js"></script> </pre>

	<pre><script type="text/javascript" src="js/easyui/jquery.easyui.all.js"></script> <script type="text/javascript" src="js/easyui/locale/easyui-lang-zh_CN.js"></script> <script type="text/javascript" src="js/bootstrap/js/bootstrap.js"></script> <script type="text/javascript" src="js/select2/select2.js"></script> <script type="text/javascript" src="js/json2.js"></script> <script type="text/javascript" src="js/select2/select2_locale_zh-CN.js"></script> <script type="text/javascript" src="js/fileSelect.js"></script> <script type="text/javascript" src="js/local/EDoc2FileSelect-lang-zh_CN.js"></script> 2、具体使用示例代码： <input type="button" id="aa" value="aa"/> <script type="text/javascript"> \$("#aa").EDoc2FileSelect({ url: 'http://localhost/EDoc2WebApi', //服务地址 token: '1149d05d-68e4-4098-91d2-256c95b582f1', //用户凭据 showSelectArea: true, //是否显示选择区 multiple: true, //是否多选 onSelectedFileCallback: function (data) { alert(data.length); } }); </script></pre>
返回值	<p>1、 设置确定按钮的回调函数，选择的文件夹信息返回给回调函数，如下所示：</p> <pre>onSelectedFileCallback: function (data) { alert(data.length); }</pre> <p>2、 返回值类型：json数组，单个数据元素返回的字段信息如下：</p> <p>FileId文件编号 FileName文件名称 ParentFolderId所在文件夹编号 FileExtName文件扩展名 FileCurVerNumStr文件当前版本 FileCurSize文件当前大小 FileCurCode文件当前编码 FileCurRemark文件当前备注 FileCurrentOperatorId文件当前操作人编号 FileCreateTime文件创建时间 FileCreateOperator文件创建人编号 FileCreateType文件创建类型 FileArchiveTime文件归档时间 FileArchiveOperator文件归档人编号 FileLastCode文件最后编码 FileLastRemark文件最后备注 FileLastSize文件最后大小 FileLastVerExtName文件最后版本扩展名</p>

FileLastVerId	文件最后版本
FileLastVerName	文件最后版本名称
FileLastVerNumStr	文件最后版本号字符串
FileModifyOperator	文件最后修改操作人编号
FileModifyTime	文件修改时间
FileState	文件状态
FileTotalSize	文件统计大小
FileType	文件类型
CreateUserName	新建人
ModifyUserName	修改人
NowOrNewVersions	当前/最新版本
ModifyDateTime	修改时间
ShowFileSize	处理后的文件大小

2.9 文件夹选择

接口形式	Js 组件
接口地址	
参数说明	<p>url: //webapi 服务地址</p> <p>token: //用户凭据</p> <p>showModal: 展现形式 dialog: 弹出窗体; comboxTree: 下拉树</p> <p>topFolderIds: '1', //顶级文件夹编号</p> <p>selectedFolderIds: ", //默认选中的文件夹编号</p> <p>showPersonalFolder: false, //是否显示个人文件夹</p> <p>multiple: false, //是否允许多选</p> <p>editable: true, //是否启用模糊查询</p> <p>minChars: 2, //最小输入字符数</p> <p>permValue: 1, //权限值 1:显示顶级文件夹;2:不显示顶级文件夹</p> <p>enableCreate: false, //是否开启新建菜单</p> <p>okCallback: function () { }, //dialog确定回调函数</p> <p>onSelectChanged: function () { } //文件夹选择更改事件</p>
示例	<p>1、在使用组件的页面引用以下 js 和 css</p> <pre> <link rel="stylesheet" type="text/css" href="js/bootstrap/css/bootstrap.css"> <link rel="stylesheet" type="text/css" href="js/easyui/themes/bootstrap/easyui.css"> <link rel="stylesheet" type="text/css" href="js/easyui/themes/icon.css"> <link rel="stylesheet" type="text/css" href="js/bootstrap/css/bootstrap-responsive.css"> <script src="js/jquery-1.8.3.js"></script> <script src="js/jquery.json-2.4.js"></script> <script src="js/easyui/jquery.easyui.all.js"></script> <script src="js/easyui/locale/easyui-lang-zh_CN.js"></script> <script src="js/bootstrap/js/bootstrap.js"></script> <script src="js/select2/select2_locale_zh-CN.js"></script> <script src="js/folderSelect.js"></script> </pre>

	<p>2、具体使用示例代码：</p> <pre><div>下拉树形式：<input id="comtree" /> <input type="button" id="dd" class="btn btn-mini" value="获取值" /></div> <div>dialog形式： <input type="button" id="btnopen" class="btn btn-mini" value="打开" /></div> <div id="cc"></div> <script type="text/javascript"> \$(document).ready(function () { \$("#comtree").EDoc2FolderSelect({ url: 'http://localhost/EDoc2WebApi', token: '1149d05d-68e4-4098-91d2-256c95b582f1', showPersonalFolder: false, editable: true, multiple: false }); \$("#dd").on("click", function () { var val = \$("#comtree").EDoc2FolderSelect('getValue'); alert(val); }); \$("#cc").EDoc2FolderSelect({ url: 'http://localhost/EDoc2WebApi', //服务地址 token: '1149d05d-68e4-4098-91d2-256c95b582f1', //用户凭据 showModal: 'dialog', //以弹出框呈现 multiple: false, //是否允许多选 editable: true, //是否允许模糊查询 okCallback: function (data) { alert(data); } }); \$("#btnopen").on("click", function () { \$("#cc").EDoc2FolderSelect('open'); }); }); </script></pre>
返回值	<p>1、以下拉树形式展现时，获取选择文件夹信息的方法为：</p> <pre>\$("#comtreeId").EDoc2FolderSelect('getValue');</pre> <p>2、以dialog打开，需设置确定按钮的回调函数，选择的文件夹信息返回给回调函数，如下所示：</p> <pre>okCallback: function (data) { alert(data); }</pre>

	<p>3、返回值类型: json数组, 单个数据元素返回的字段信息如下:</p> <p>FolderId: 文件夹编号</p> <p>AreaId: 区域编号</p> <p>ParentFolderId: 父文件夹编号</p> <p>FolderCode: 文件夹编号</p> <p>FolderSortOrder: 文件排序值</p> <p>FolderName: 文件夹名称</p> <p>FolderPath: 文件夹路径</p> <p>FolderNamePath: 文件名路径</p> <p>FolderSize: 文件夹大小</p> <p>FolderMaxFolderSize: 文件夹最大尺寸</p> <p>FolderAlertSize: 文件夹提示大小</p> <p>FolderMaxFileSize: 文件夹中文件最大尺寸</p> <p>FolderForbiddenFileExtensions: 禁止上传文件扩展名列表</p> <p>FolderChildFoldersCount: 文件夹子文件夹数量</p> <p>FolderChildFilesCount: 文件夹子文件数量</p> <p>SecurityLevelId: 安全等级编号</p> <p>FolderState: 文件夹状态</p> <p>FolderLockCount: 文件夹锁定数量{ <code>get</code>; <code>set</code>; }</p> <p>FolderCreateTime: 文件夹创建时间</p> <p>FolderCreateOperator: 文件夹创建人编号</p> <p>FolderModifyTime: 文件夹修改时间</p> <p>FolderModifyOperator: 文件夹修改人编号</p> <p>FolderArchiveTime: 文件夹归档时间</p> <p>FolderArchiveOperator: 文件夹归档人编号</p> <p>FolderCurVerId: 文件夹版本号</p> <p>FolderNewestVerId</p> <p>FolderType: 文件夹类型</p> <p>FolderGuid</p> <p>FolderOwnerId: 文件夹拥有人编号</p> <p>IsDeleted: 是否删除</p> <p>FolderDeleteTime: 文件夹删除时间</p> <p>FolderDeleteOperator: 文件夹删除人编号</p> <p>FolderIsCascadeDelete</p> <p>RelativePath</p> <p>IsArcFolder</p> <p>HasBoundStorageArea</p>
--	---

2.10 edoc2 主界面操作说明

2.10.1 默认选中指定页签

相关文件: EDoc2V4\Website\Default.aspx

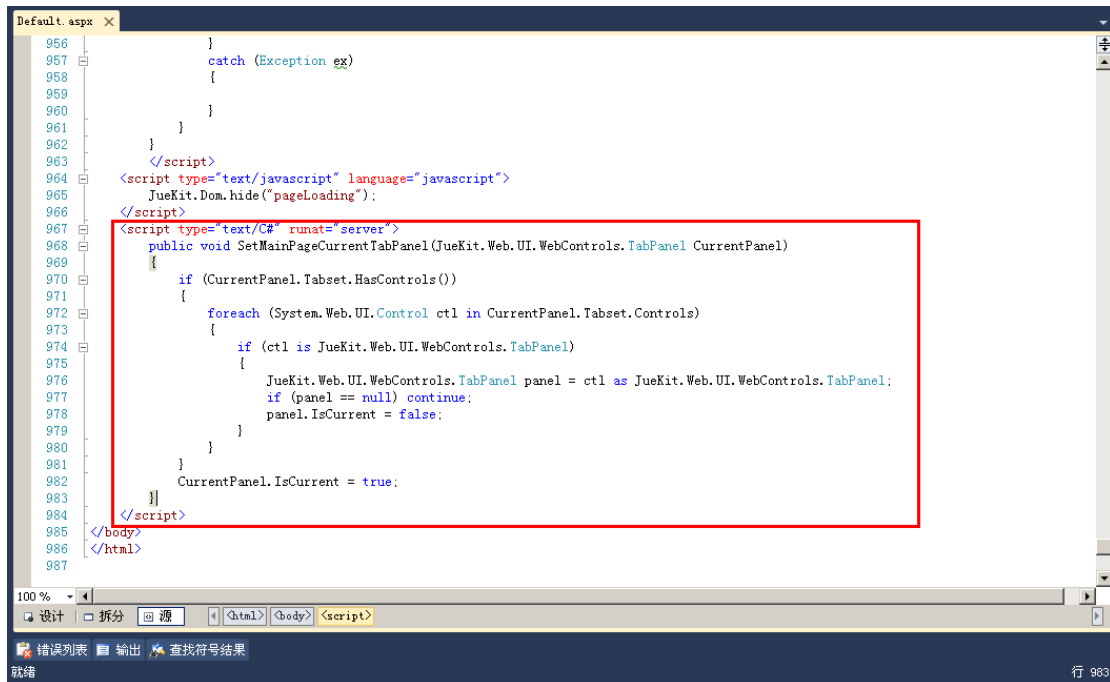
示例: edoc2 主界面默认选中“资源数据库”页签

① Edoc 主页加载完后选中指定页签, 假设要默认选中“资源数据库”页签



② 修改 Edoc2V4 Default.aspx，在</body>节点前加入如下代码

```
<script type="text/C#" runat="server">
    public void SetMainPageCurrentTabPanel(JueKit.Web.UI.WebControls.TabPanel CurrentPanel)
    {
        if (CurrentPanel.Tabset.HasControls())
        {
            foreach (System.Web.UI.Control ctl in CurrentPanel.Tabset.Controls)
            {
                if (ctl is JueKit.Web.UI.WebControls.TabPanel)
                {
                    JueKit.Web.UI.WebControls.TabPanel panel = ctl as
JueKit.Web.UI.WebControls.TabPanel;
                    if (panel == null) continue;
                    panel.IsCurrent = false;
                }
            }
            CurrentPanel.IsCurrent = true;
        }
    }
</script>
```



```
956         }
957         catch (Exception ex)
958         {
959         }
960     }
961 }
962 </script>
963 <script type="text/javascript" language="javascript">
964     JueKit.Dom.hide("pageLoading");
965 </script>
966 <script type="text/C#" runat="server">
967     public void SetMainPageCurrentTabPanel(JueKit.Web.UI.WebControls.TabPanel CurrentPanel)
968     {
969         if (CurrentPanel.Tabset.HasControls())
970         {
971             foreach (System.Web.UI.Control ctl in CurrentPanel.Tabset.Controls)
972             {
973                 if (ctl is JueKit.Web.UI.WebControls.TabPanel)
974                 {
975                     JueKit.Web.UI.WebControls.TabPanel panel = ctl as JueKit.Web.UI.WebControls.TabPanel;
976                     if (panel == null) continue;
977                     panel.IsCurrent = false;
978                 }
979             }
980             CurrentPanel.IsCurrent = true;
981         }
982     }
983 </script>
984 </body>
985 </html>
986
987
```

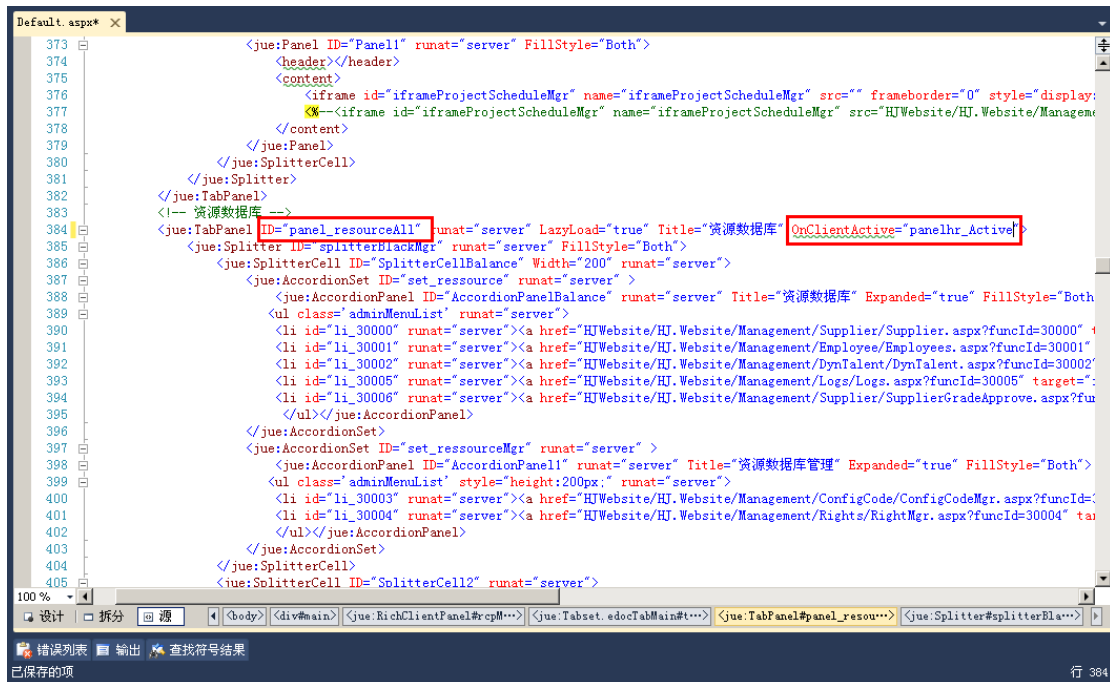
③ 还需要加上如下代码（具体位置没有特殊要求，只要不要打乱页面结构就行。）



```
826 if (location.search.indexOf("client_id=vdriver") != -1)
827 {
828     document.write('<OBJECT ID="object_SearchListMsgo" CLASSID="CLSID:85F13298-B58E-4223-B6A2-7869F4CA4E2C" width="0" height="0">');
829     document.write('<script type="text/javascript">object_SearchListMsgo_success = false;</script>');
830     document.write('<script></OBJECT>');
831 }
832 </script>
833 </div>
834 <uc2:Footer ID="Footer1" runat="server" ShowLayoutSwitch="true" ShowLanguageSwitch="true" />
835 <script language="javascript" type="text/javascript">
836     function fileDeleteRemark_btnOK_Click(sender, args) ...
837 </script>
838 <script type="text/javascript" language="javascript">
839     function panelhr_Active(sender, args) {
840         if (args.firstActive) {
841             JueKit.Dom.hide("pageLoading");
842         }
843     }
844 </script>
845 <script type="text/C#" runat="server">
846     public void Page_LoadComplete(object sender, EventArgs e)
847     {
848     }
849 </script>
850
```

```
<script type="text/javascript" language="javascript">
    function panelhr_Active(sender, args) {
        if (args.firstActive) {
            JueKit.Dom.hide("pageLoading");
        }
    }
</script>
```

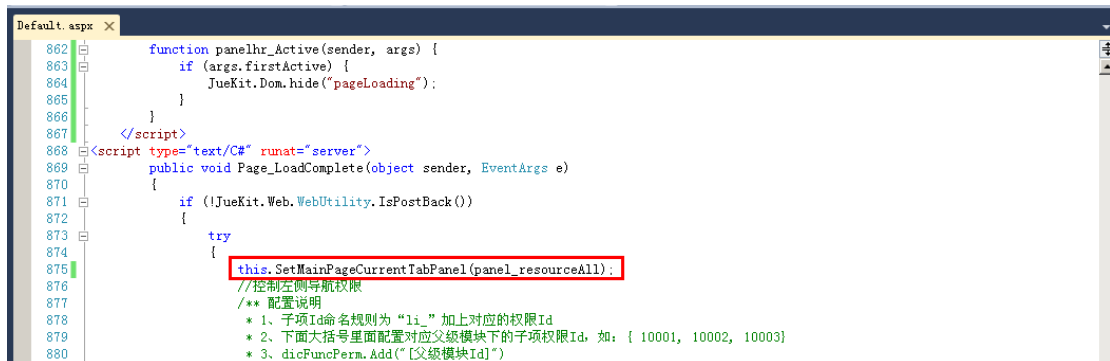
④ 在 Default.aspx 页面找到“资源数据库”页签的位置



⑤ 在相应位置加上 `OnClientActive="panelhr_Active"`，`panelhr_Active` 是上面插入的 js 方法的方法名称，并且记录下该页签的 `Id ID="panel_resourceAll"`；

```
861 </script>
862 <script type="text/javascript" language="javascript">
863     function panelhr_Active(sender, args) {
864         if (args.firstActive) {
865             JueKit.Dom.hide("pageLoading");
866         }
867     }
868 </script>
```

⑥ 在 `Page_LoadComplete` 方法中加入如下代码



`this.SetMainPageCurrentTabPanel(panel_resourceAll);`

`panel_resourceAll` 就是记录下来的“资源数据库”页签的 `Id`

2.10.2 添加页签和导航并控制权限

相关文件：EDoc2V4\Website\Default.aspx

相关数据库表：EDoc2V4.oc_function

示例：在“系统管理”页签前面新增“新增页签”，并控制显示权限

2.10.2.1 主界面添加页签

修改EDoc2V4\Website\Default.aspx文件，在 `<jue:tabpanel id="panelAdmin" runat="server" lazyload="false">` 前面插入如下代码，注意ID不要和Default.aspx上的其他元素ID重名，记录下ID。

```
<jue:TabPanel ID="panel_NewPanel" runat="server" LazyLoad="false" Title="新增页签">
    <iframe frameborder="0" style="display: block; width: 100%; height: 100%; scrolling="auto"></iframe>
</jue:TabPanel>
```



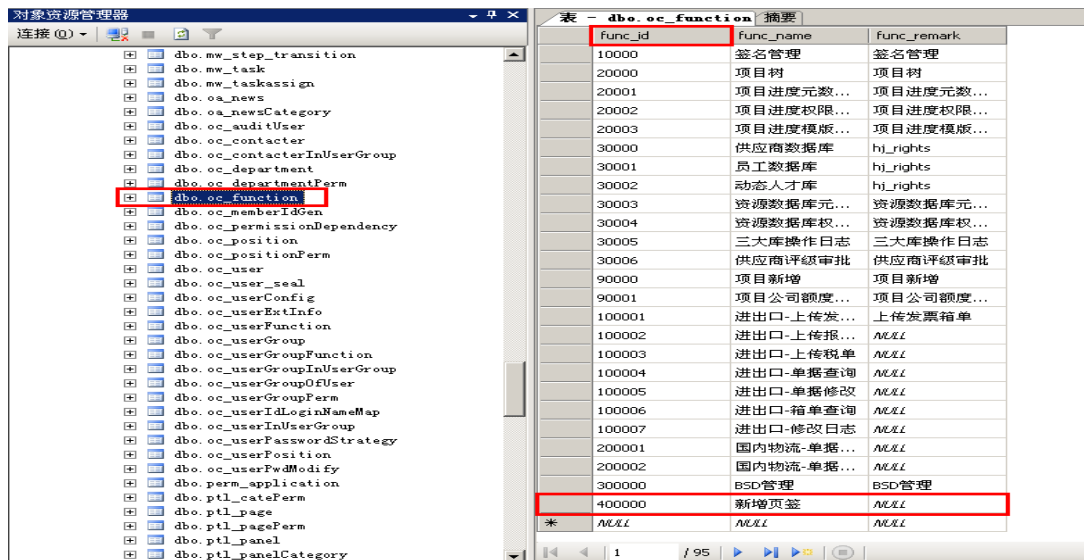
其中iframe可以为其src属性赋值，为该页签下显示的默认页面，jue:TabPanel的LazyLoad属性表示是否在主界面加载的时候加载iframe的内容，为ture时加载，false时则在用户点击该页签时才加载。

添加代码保存后刷新主界面效果图如下：



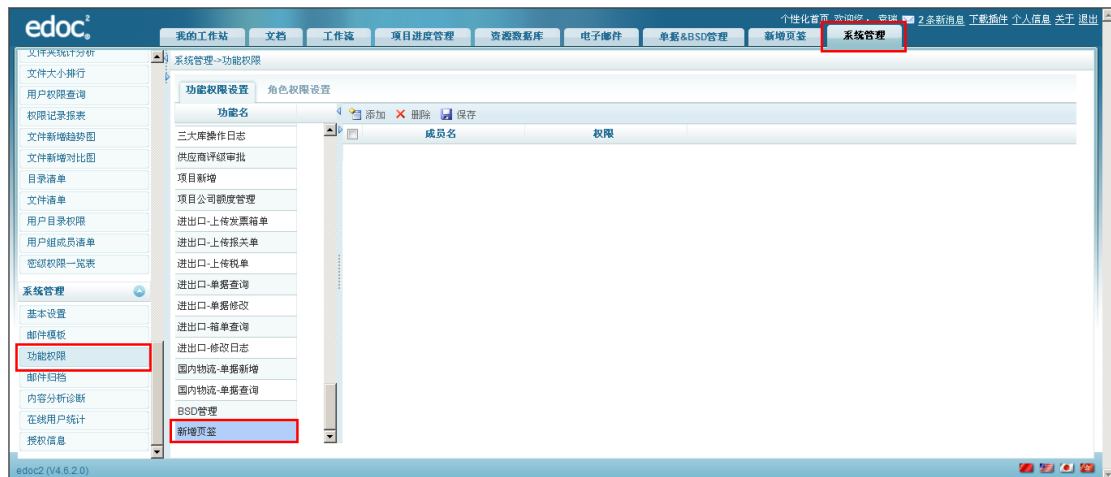
2.10.2.2 控制权限

① 在EDoc2V4数据库的oc_function表中添加数据,注意func_id不要重复,并记录下func_id配置的值,后面会用到



func_id	func_name	func_remark
10000	签名管理	签名管理
20000	项目树	项目树
20001	项目进度元数...	项目进度元数...
20002	项目进度权限...	项目进度权限...
20003	项目进度模版...	项目进度模版...
30000	供应商数据库	hj_rights
30001	员工数据库	hj_rights
30002	动态人才库	hj_rights
30003	资源数据库元...	资源数据库元...
30004	资源数据库权...	资源数据库权...
30005	三大库操作日志	三大库操作日志
30006	供应商评级审批	供应商评级审批
90000	项目新增	项目新增
90001	项目公司额度...	项目公司额度...
100001	进出口-上传发...	上传发票箱单
100002	进出口-上传报...	NULL
100003	进出口-上传税单	NULL
100004	进出口-单据查询	NULL
100005	进出口-单据修改	NULL
100006	进出口-箱单查询	NULL
100007	进出口-修改日志	NULL
200001	国内物流-单据...	NULL
200002	国内物流-单据...	NULL
300000	BSD管理	BSD管理
400000	新增页签	NULL
*	NULL	NULL

添加数据后 EDoc2“系统管理” “功能权限” 中效果图：



② 以上配置 Ok 后，我们开始控制该新增页签的显示权限，仍然是在 Default.aspx 添加代码在 Default.aspx 的</body>节点前面加入如下代码：

```
<script type="text/C#" runat="server">
    public void Page_LoadComplete(object sender, EventArgs e)
    {
        if (!JueKit.Web.WebUtility.IsPostBack())
        {
            try
            {
                //判断对新增页签是否有限 新增页签对应权限Id 400000
                if(GetFuncPerm(400000)!=EDoc2.EDoc2ApiConst.FUNC_PERM_NOPERMISSION)
                {
                    //有权限，显示新增页签
                    panel_NewPanel.Visible = true;
                }
            }
            else
            {
            }
        }
    }
}
```



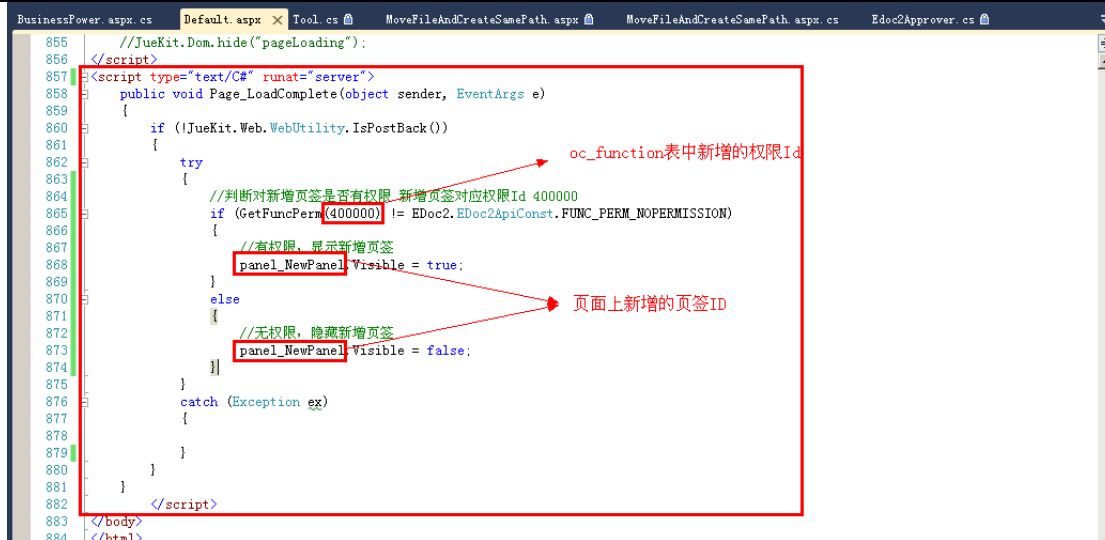
```

    {
        //无权限，隐藏新增页签
        panel_NewPanel.Visible = false;
    }
}
catch (Exception ex)
{
    //异常处理
}
}
}
</script>

```

其中GetFuncPerm方法是关键，它是获取当前用户是否具有某个功能权限的方法，传入权限Id（即oc_function里面的func_id），返回值如果不等于

EDoc2.EDoc2ApiConst.FUNC_PERM_NOPERMISSION，则说明是有权限的。

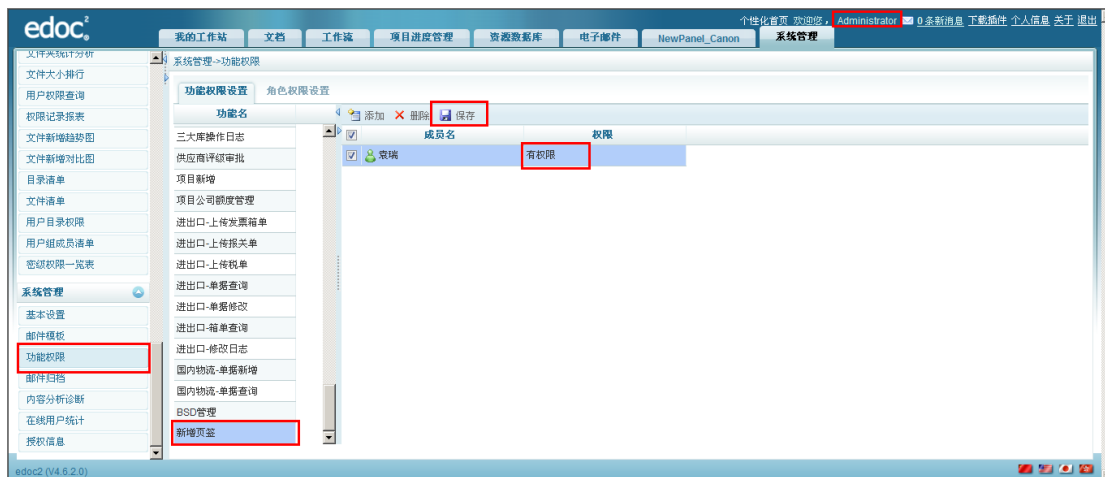


2.10.2.3 验证

① 刷新页面后，这个用户是看不到这个页签了



② 用 admin 登录设置权限

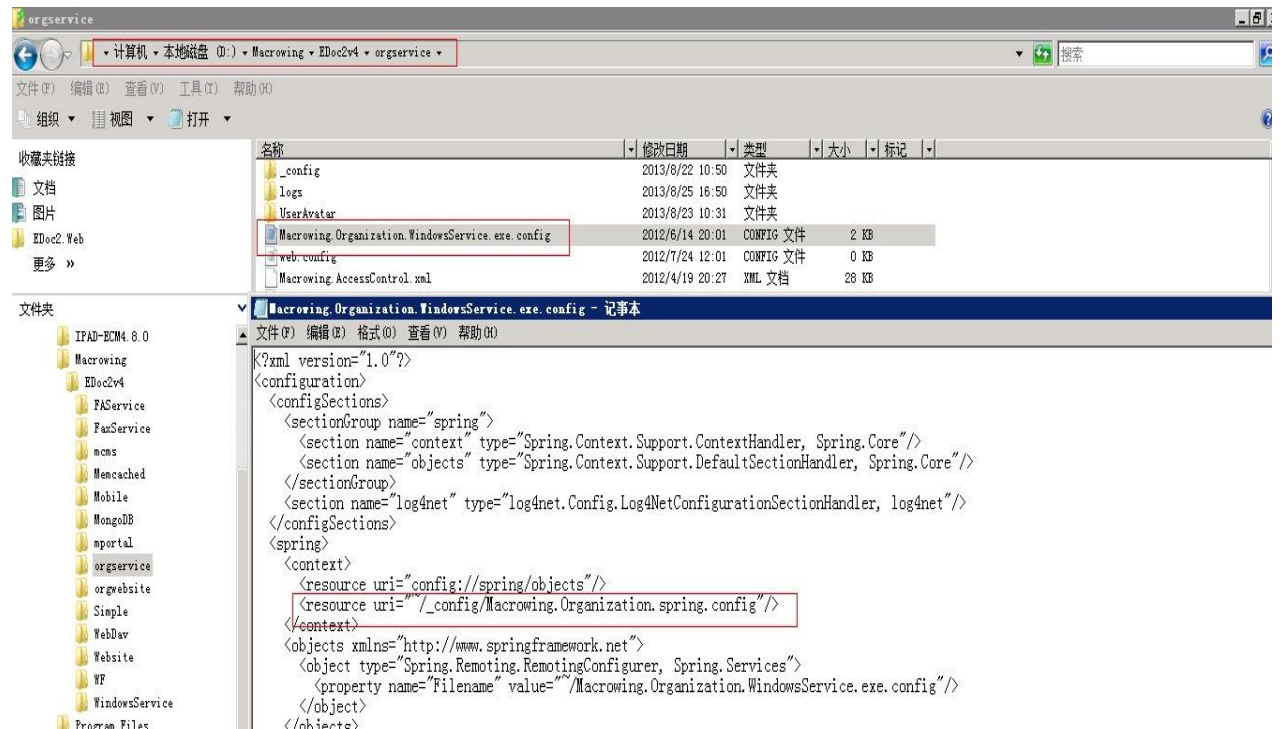


③ 登录后发现显示“新增页签”了



3 edoc2 组织集成(仅适用 ECM)

3.1 组织集成配置文件修改



1> 打开 **orgservice\Macrowing.Organization.WindowsService.exe.config** 配置文件

2> 修改成配置文件路径

```
<resource uri="_config/Macrowing.Organization.thirdpart.spring.config"/>
```

在 appSettings 增加

```
<appSettings>
```

```
<!--是否使用OCM的职位-->
```

```
<add key="EnablePosition" value="false"></add>
```

```
<!--是否使用OCM的用户组-->
```

```
<add key="EnableGroup" value="true"/>
```

```
<!--是否缓存第三方数据-->
```

```
<add key="EnableCache" value="true"/>
```

```
<!--缓存第三方数据时间(单位: 秒)-->
```

```
<add key="CacheExpiration" value="1800"/>
```

```
<!--是否初始化edoc2扩展信息-->
```

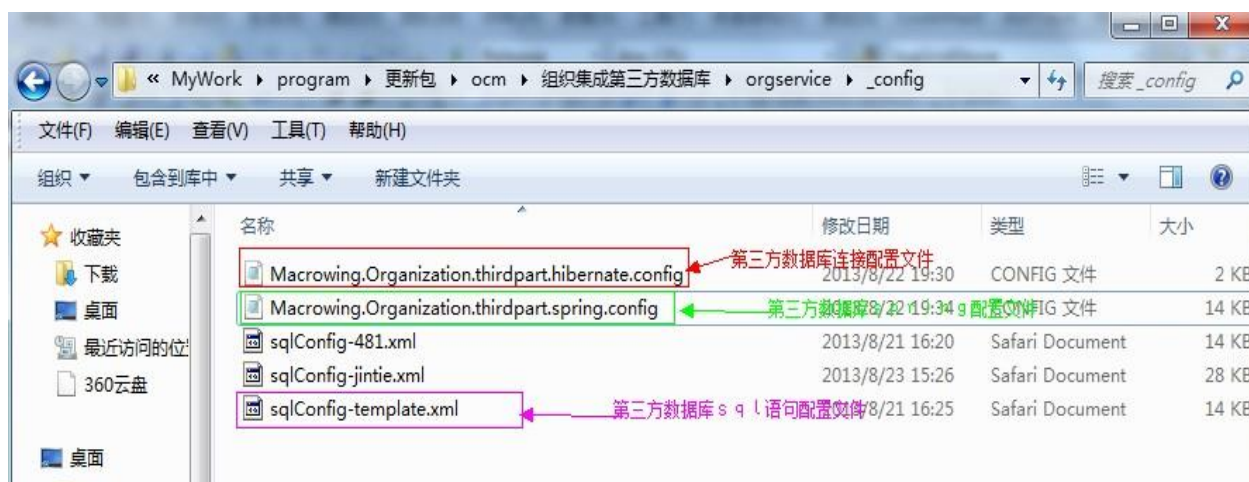
```
<add key="InitEDoc2Extension" value="true"></add>
```

```
<!--数据库部门主职位前缀标识-->
```

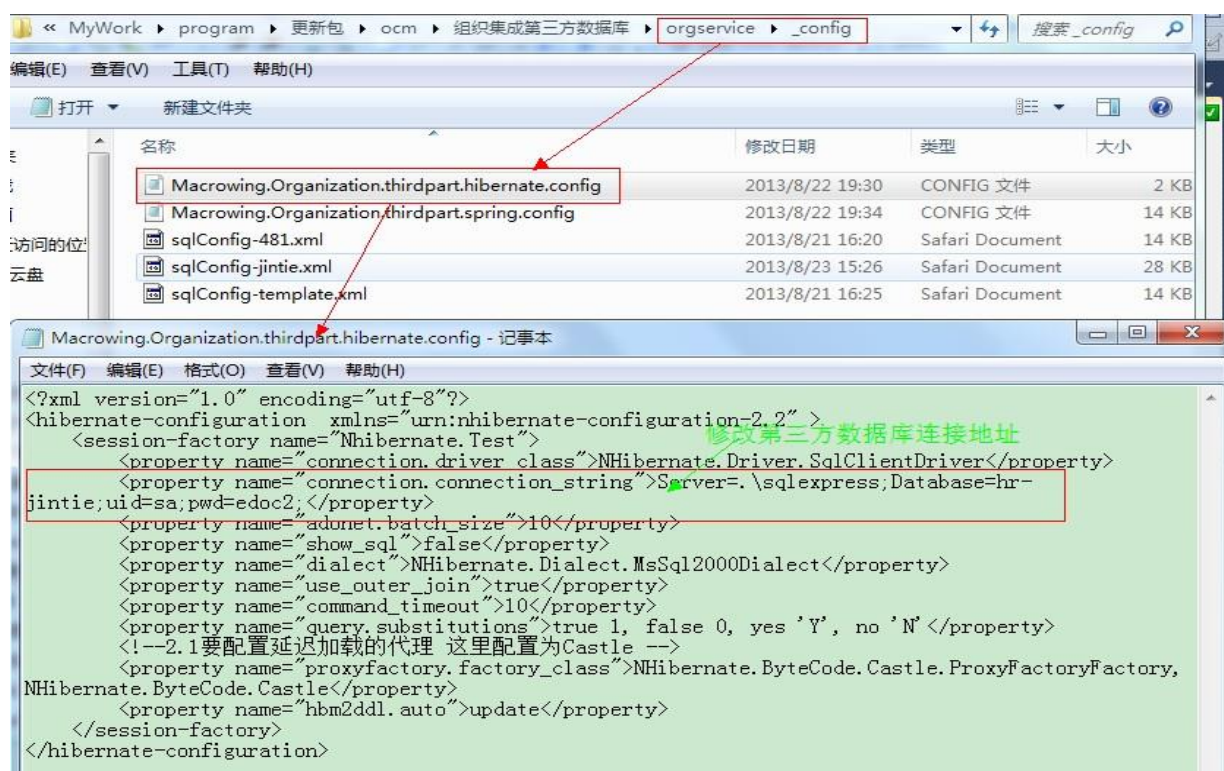
```
<add key="DeptMasterPositionIdPrefix" value="deptMasterPositionId-"/>
```

```
</appSettings>
```

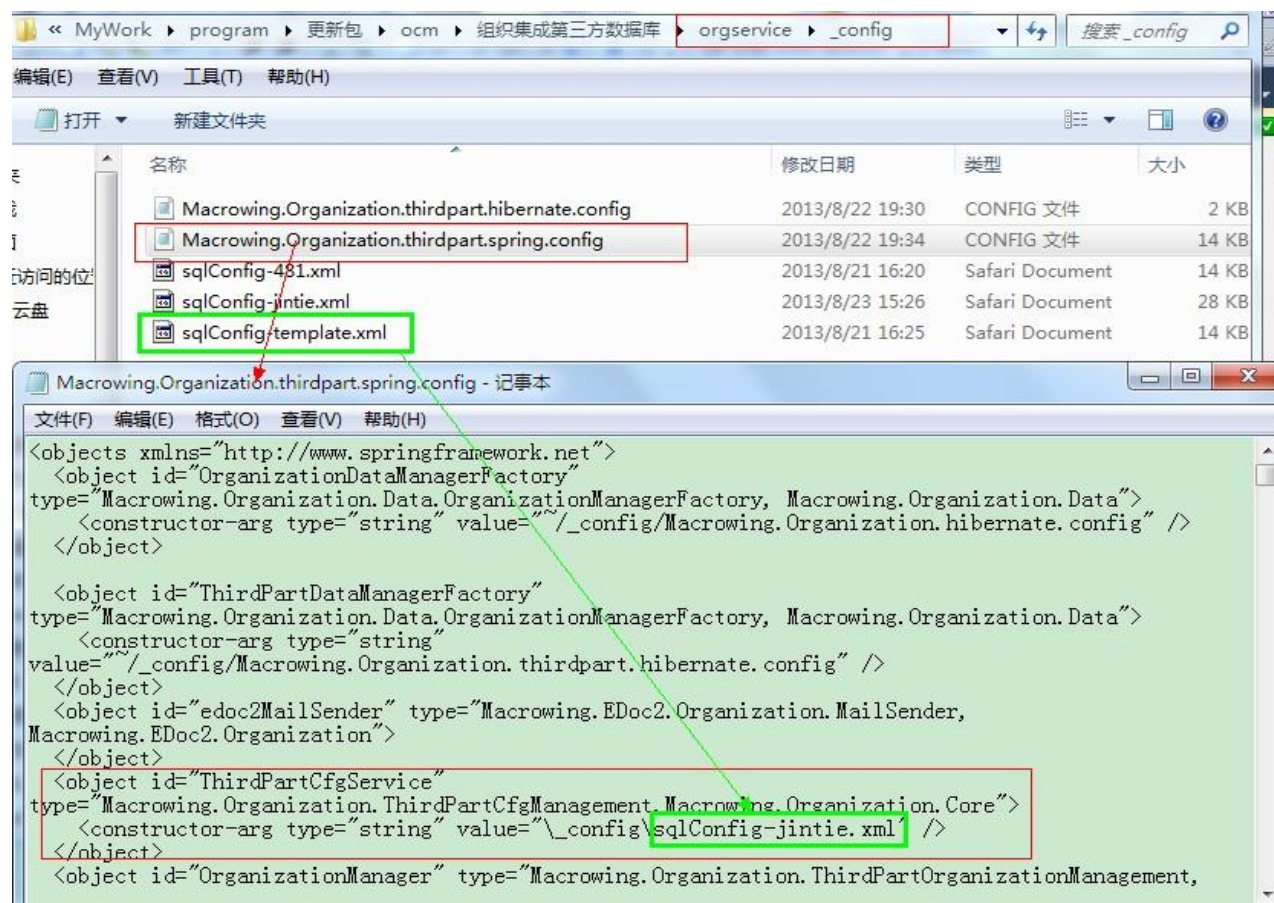
3> 修改文件具体内容



4> **Macrowing.Organization.thirdpart.hibernate.config** 中修改数据库配置文件



5> **Macrowing.Organization.thirdpart.spring.config** 中的配置与文件对应



在此文件的“ThirdPartCfgService”的配置节中的 Value 要与文件对应

3.2 SqlConfig-template.xml 文件 sql 配置项说明

3.2.1 查询组织顶级部门

名称	TopDepartment	代码	1001	是否必填项:	是
描述	查询组织顶级部门				
返回字段	dept_id: 部门Id not null dept_identityID: 部门自增才Id, 没有返回-1 dept_code: 部门代码.如果有则必须配置GetDepartmentByCode, 否则返回部门Id dept_thirdPartId: 返回部门Id dept_masterPositionId: 主职位Id, 无则返回部门Id, 获取职位时使用sql语句根据部门Id模拟出职位信息 not null dept_name: 部门名称 not null dept_sort: 排序 dept_remark: 备注				
实例:	<pre>select a.DepID as dept_id , -1 as dept_identityid, '' as dept_code, '' as dept_thirdpartid, b.JobID as dept_masterpositionid,</pre>				

	<pre>a.DepCName as dept_name, 0 as dept_sort, '' as dept_remark from eip_oVA_Department a inner join eip_evw_emp b on a.director=b.badge inner join eip_oVA_Job c on b.jobid=c.jobid where a.adminid is null;</pre>
--	---

3.2.2 根据部门 Id 查询上级部门

名称	parentDepartment	代码	1002	是否必配项:	是
描述	根据部门Id查询上级部门				
返回字段	同代码 1001				
实例:					

3.2.3 根据部门 Id 查询部门信息

名称	GetDepartmentById	代码	1003	是否必配项:	是
描述	根据部门Id查询部门信息				
返回字段	同代码 1001				
实例:					

3.2.4 根据部门代码查询部门信息

名称	GetDepartmentByCode	代码	1004	是否必配项:	否
描述	根据部门代码查询部门信息，无空则根据部门Id查询，此时部门id等于部门code				
返回字段	同代码 1001				
实例:					

3.2.5 根据部门自增长 Id 获取部门信息

名称	GetDepartmentByIdentityId	代码	1005	是否必配项:	否
描述	根据部门自增Id获取部门信息(如果为空则根据部门Id查询)				
返回字段	同代码 1001				
实例:					

3.2.6 根据职位 Id 查询此职位所在的部门

名称	GetDepartmentByPositionId	代码	1006	是否必配项:	否
描述	根据职位Id查询此职位所在的部门（如果为空则根据部门Id查询，此时职位Id=部门Id）				
返回字段	同代码 1001				
实例:					

3.2.7 根据第三方 Id 获取部门信息

名称	GetDepartmentByThirdPartId	代码	1007	是否必配项:	否
----	----------------------------	----	------	--------	---

描述	根据第三方Id获取部门信息（Value如果为空则根据部门Id查询）
返回字段	
实例	

3.2.8 根据部门名称获取顶级部门

名称	GetTopDepartmentByDeptName	代码	1008	是否必配项:	是
描述	根据部门名称获取顶级部门，value不能为空				
返回字段	同代码 1001				
实例:	<pre>select a.depid as dept_id, -1 as dept_identityid, '' as dept_code, '' as dept_thirdpartid, isnull(b.jobid, a.depid) as dept_masterpositionid, a.depcname as dept_name, 0 as dept_sort, a.description as dept_remark from eip_oVA_Department a left join eip_evw_emp b on a.director=b.badge left join eip_oVA_Job c on b.jobid=c.jobid where a.AdminID is null and a.DepCName = '{0}'</pre>				

3.2.9 根据部门名称模糊查询符合条件的部门

名称	SearchDepartmentByName	代码	1009	是否必配项:	是
描述	根据部门名称模糊查询符合条件的部门				
返回字段	同代码 1001				
实例:	<pre>select a.depid as dept_id, -1 as dept_identityid, '' as dept_code, '' as dept_thirdpartid, isnull(b.jobid,a.depid) as dept_masterpositionid, a.depcname as dept_name, 0 as dept_sort, a.description as dept_remark from eip_oVA_Department a left join eip_evw_emp b on a.director=b.badge left join eip_oVA_Job c on b.jobid=c.jobid where a.DepCName like N'%{0}%'</pre>				

3.2.10 根据部门名称查询符合条件的部门

名称	GetDepartmentByDeptName	代码	1010	是否必配项:	是
描述	根据部门名称查询符合条件的部门				
返回字段	同代码 1001				
实例:	<pre>select</pre>				

	<pre> a.depid as dept_id, -1 as dept_identityid, '' as dept_code, '' as dept_thirdpartid, isnull(b.jobid,a.depid) as dept_masterpositionid, a.depcname as dept_name, 0 as dept_sort, a.description as dept_remark from eip_oVA_Department a left join eip_evw_emp b on a.director=b.badge left join eip_oVA_Job c on b.jobid=c.jobid where a.DepCName like N'%{0}%' </pre>
--	--

3.2.11 根据当前部门Id 查询子部门

名称	GetChildrenDepartment	代码	1011	是否必配项:	是
描述	根据当前部门Id查询子部门				
返回字段	同代码 1001				
实例:	<pre> select a.depid as dept_id, -1 as dept_identityid, '' as dept_code, '' as dept_thirdpartid, isnull(b.jobid ,a.depid) as dept_masterpositionid, a.depcname as dept_name, 0 as dept_sort, a.description as dept_remark from eip_oVA_Department a left join eip_evw_emp b on a.director=b.badge left join eip_oVA_Job c on b.jobid=c.jobid where A.AdminID ='{0}' </pre>				

3.2.12 根据部门Id 查询部门下的用户

名称	GetDepartmentUserById	代码	1012	是否必配项:	是
描述	根据部门Id查询部门下的用户				
返回字段	<pre> user_id: 用户Id not null user_identityID:用户自增才Id, 无则返回-1 user_code:用户编号。如果有则必须配置GetUserByCode, 否则返回用户Id user_thirdPartId:返回用户Id user_account: 用户账号 not null user_password: 用户密码 user_digesthash user_email user_secuQuestion user_secuAnswer user_name : 用户姓名 not null </pre>				

	<code>user_gender: 性别</code> <code>user_role: 角色 0:用户; : 管理员 not null</code> <code>user_birthday: 生日</code> <code>user_telephone: 手机</code> <code>user_mobile</code> <code>user_avatar: 头像</code> <code>user_status: 用户状态 0: 正常; : 注销; : 锁定 not null</code> <code>user_fax</code> <code>user_expirationTime</code> <code>user_creatorId: 创建人</code> <code>user_createTime: 创建时间</code> <code>user_changeUserId</code> <code>user_changeTime</code> <code>user_passwordChangeTime</code> <code>user_lastLoginTime</code> <code>user_lastLoginIp</code> <code>user_signed</code> <code>user_passwordWrongCount</code> <code>user_remark</code>
实例:	

3.2.13 根据部门 Id 查询部门下的用户

名称	<code>GetLeaderByDepartmentId</code>	代码	1013	是否必配项:	是
描述	根据部门Id查询部门下的用户				
返回字段	同代码 1012				
实例:					

3.2.14 查询组织顶级职位信息

名称	<code>GetTopPosition</code>	代码	2001	是否必配项:	是
描述	查询组织顶级职位信息（如果没有则需根据部门信息模拟提供职位）				
返回字段	<code>position_id: 职位Id not null</code> <code>position_identityID: 职位自增才Id, 无则返回-1</code> <code>position_code: 职位代码。如果有则必须配置GetPositionByCode, 否则返回职位Id</code> <code>position_thirdPartId: 直接返回职位Id</code> <code>position_parentId: 上级职位Id not null;</code> <code>position_name: 职位名称 not null;</code> <code>position_levelId</code> <code>position_type: 职位类型 not null</code> <code>position_sort</code> <code>position_remark</code> <code>dept_id: 职位所在部门Id not null</code>				
实例:	<code>select</code>				

```

a.JobID as position_id,
-1 as position_identityid,
a.JobID as position_code,
a.JobID as position_thirdpartid,
b.JobID as position_parentid,
a.JobCName as position_name,
0 as position_levelId,
0 as position_sort, '' as position_remark,
case
  when ISNULL ( c.childrenCount,0)>0
  then 16
  else 4
end as position_type,
b.DepID as dept_id
from eip_oVA_Job a
inner join eip_evw_emp b on a.JobID=b.JobID
inner join eip_oVA_Department d on d.DepID=b.DepID
left join (select Reporttto ,COUNT (Reporttto) as childrenCount
           from eip_evw_emp
           group by Reporttto) c on b.Badge =c.Reporttto
where b.Badge ='sha10001'

```

3.2.15 根据职位 ID 查询上级职位信息

名称	GetParentPosition	代码	2002	是否必配项:	是
描述	查询上级职位信息1: 根据职位ID				
返回字段	同 2001				
实例:	<pre> select distinct d.JobID as position_id, -1 as position_identityID, d.JobID as position_code, d.JobID as position_thirdPartId, d.JobCName as position_name, e.AdminID as dept_id , e.AdminID as position_parentid, 16 as position_type, 0 as position_levelId, 0 as position_sort, '' as position_remark from eip_oVA_Job a left join eip_evw_emp b on a.JobID =b.JobID left join eip_evw_emp c on b.Reporttto=c.Badge left join eip_oVA_Job d on c.JobID =d.JobID left join eip_oVA_Department e on c.DepID =e.DepID where a.JobID ='{0}' </pre>				

3.2.16 根据职位 ID 查询上级职位信息

名称	GetParentPositionByDeptId	代码	2002	是否必配项:	是
描述	查询上级职位信息1: 根据职位ID				
返回字段	同 2001				
实例:					

3.2.17 根据职位对应的部门 Id 查询职位信息

名称	GetPositionByDeptId	代码	2003	是否必配项:	是
描述	查询职位信息: 根据职位对应的部门Id				
返回字段	同 2001				
实例:	<pre> select position_id, -1 as position_identityID, position_id as position_code, position_id as position_thirdPartId, position_name,position_type, 0 as position_levelId, 0 as position_sort, '' as position_remark, '{0}' as position_parentId,dept_id from (select a.DepID as dept_id, case when ISNULL (c.JobID, '')='' then a.DepID else c.JobID end as position_id, case when ISNULL (c.JobCName, '')='' then a.DepCName+N'主职位' else c.JobCName end as position_name,16 as position_type from eip_oVA_Department a left join eip_evw_emp b on a.Director =b.Badge left join eip_oVA_Job c on b.JobID =c.JobID where a.DepID ='{0}')</pre>				

3.2.18 根据职位对应的部门 Id 查询职位信息

名称	GetPositionById	代码	2003	是否必配项:	是
描述	查询职位信息: 根据职位对应的部门Id				
返回字段	同 2001				
实例:					

3.2.19 根据职位编号查询职位信息

名称	GetPositionByCode	代码	2004	是否必配项:	否
描述	根据职位编号查询职位信息: 未配置则根据职位Id查询				
返回字段	同 2001				
实例:					

3.2.20 根据职位自增长 Id 查询职位信息

名称	GetPositionByIdentityId	代码	2005	是否必配项:	否
描述	根据职位自增长Id查询职位信息				
返回字段	同 2001				
实例:					

3.2.21 根据职位名称查询职位信息

名称	GetPositionByName	代码	2006	是否必配项:	是
描述	根据职位名称查询职位信息				
返回字段	同 2001				
实例:					

3.2.22 根据职位第三方 Id 查询职位信息

名称	GetPositionByThirdPartId	代码	2007	是否必配项:	否
描述	根据职位第三方Id查询职位信息				
返回字段	同 2001				
实例:					

3.2.23 查询用户的职位

名称	GetUserPositions	代码	2008	是否必配项:	是
描述	查询用户的职位				
返回字段	同 2001				
实例:					

3.2.24 根据父职位 Id 查询子职位

名称	GetPositionsByParentId	代码	2009	是否必配项:	是
描述	查询子职位1: 根据父职位Id				
返回字段	同 2001				
实例:	<pre>select d.JobID as position_id, -1 as position_identityID, d.JobID as position_code, d.JobID as position_thirdPartId,</pre>				

	<pre> a.JobID as position_parentId, d.JobCName as position_name, 0 as position_levelId, 0 as position_sort, '' as position_remark, b.DepID as dept_id, 4 as position_type from eip_oVA_Job a left join eip_evw_emp b on a.JobID=b.JobID inner join eip_evw_emp c on b.Badge =c.Reportto inner join eip_oVA_Job d on c.JobID =d.JobID where a.JobID ='{0}' </pre>
--	--

3.2.25 根据部门 Id 查询子职位

名称	GetPositionsByDepartmentId	代码	2009	是否必配项:	是
描述	查询子职位2: 根据部门Id				
返回字段	同 2001				
实例:	<pre> select position_id, -1 as position_identityID, position_id as position_code, position_id as position_thirdPartId, position_name,position_type, 0 as position_levelId, 0 as position_sort, '' as position_remark, '{1}' as position_parentId,dept_id from (select a.DepID as dept_id, case when ISNULL (c.JobID, '')='' then a.DepID else c.JobID end as position_id, case when ISNULL (c.JobCName, '')='' then a.DepCName+N'主职位' else c.JobCName end as position_name,16 as position_type from eip_oVA_Department a left join eip_evw_emp b on a.Director =b.Badge left join eip_oVA_Job c on b.JobID =c.JobID where a.AdminID ='{0}')) tab </pre>				

3.2.26 根据职位 Id 查询此职位下的部门

名称	GetChildrenDepartmentByPositionId	代码	2010	是否必配项:	否
描述	根据职位Id查询此职位下的部门(Value为空则使用GetChildrenDepartment查询, 此时职位Id=部门Id)				

返回字段	同 2001
实例:	

3.2.27 根据职位 Id 查询此职位下已注销的用户信息

名称	GetLogOffedUserByPositionId	代码	2011	是否必配项:	是
描述	根据职位Id查询此职位下已注销的用户信息 (value为空则根据部门Id查询, 此时职位Id=部门Id)				
返回字段	同 2001				
实例:					

3.2.28 根据职位 Id 查询职位下的用户

名称	GetUserByPositionId	代码	2012	是否必配项:	否
描述	根据职位Id查询职位下的用户 (value为空则根据部门Id查询, 此时职位Id=部门Id)				
返回字段	同 1012				
实例:					

3.2.29 查询所有的职位信息

名称	GetAllPosition	代码	2013	是否必配项:	是
描述	查询所有的职位信息				
返回字段	同 2001				
实例:					

3.2.30 查询所有的用户信息

名称	GetAllUser	代码	3001	是否必配项:	是
描述	查询所有的用户信息				
返回字段	同 1012				
实例:					

3.2.31 查询正常状态用户的个数

名称	GetNormalUserCount	代码	3002	是否必配项:	是
描述	查询正常状态用户的个数				
返回字段	select count(*) from evw_changeemp				
实例:					

3.2.32 查询用户所在的部门

名称	GetUserMainDepartment	代码	3003	是否必配项:	是
描述	查询用户所在的部门				
返回字段	同 1012				

实例:	
-----	--

3.2.33 根据用户 Id 查询用户所有的职位

名称	GetUserPositionsByUserId	代码	3004	是否必配项:	是
描述	根据用户Id查询用户所有的职位				
返回字段	同 2001				
实例:					

3.2.34 根据帐号查询用户信息

名称	GetUserByAccount	代码	3005	是否必配项:	是
描述	根据帐号查询用户信息				
返回字段	同 1012				
实例:					

3.2.35 根据编号查询用户信息

名称	GetUserByCode	代码	3006	是否必配项:	否
描述	根据编号查询用户信息				
返回字段	同 1012				
实例:					

3.2.36 根据用户 Id 查询用户信息

名称	GetUserById	代码	3007	是否必配项:	是
描述	根据用户Id查询用户信息				
返回字段	同 1012				
实例:					

3.2.37 根据用户自增长 Id 查询用户信息

名称	GetUserByIdentityId	代码	3008	是否必配项:	否
描述	根据用户自增长Id查询用户信息				
返回字段	同 1012				
实例:					

3.2.38 根据第三方 Id 查询用户信息

名称	GetUserByThirdPartId	代码	3009	是否必配项:	否
描述	根据第三方Id查询用户信息, 如果为空则根据部门Id查询。				
返回字段	同 1012				
实例:					

3.2.39 根据用户帐号模糊查询用户信息

名称	GetUsersByLikeAccount	代码	3010	是否必配项:	是
----	---------------------------------------	----	------	--------	---

描述	根据用户帐号模糊查询用户信息
返回字段	同 1012
实例:	

3.2.40 根据用户姓名模糊查询用户信息

名称	GetUsersByLikeName	代码	3011	是否必配项:	是
描述	根据用户姓名模糊查询用户信息				
返回字段	同 1012				
实例:					

3.2.41 根据用户姓名和帐号模糊查询用户信息

名称	GetUsersByLikeAccountAndName	代码	3012	是否必配项:	是
描述	根据用户姓名和帐号模糊查询用户信息				
返回字段	同 1012				
实例:					

3.2.42 根据用户 Id 查询此用户所在的所有部门信息

名称	GetUserAllDepartments	代码	3013	是否必配项:	是
描述	根据用户Id查询此用户所在的所有部门信息				
返回字段	同 1012				
实例:					

3.2.43 根据用户 Id 查询用户所在的主职位

名称	GetUserMainPositionByUserId	代码	3014	是否必配项:	是
描述	根据用户Id查询用户所在的主职位				
返回字段	同 2001				
实例:					

3.2.44 根据用户组 Id 查询用户组

名称	GetGroupById	代码	4001	是否必配项:	是
描述	根据用户组Id查询用户组信息				
返回字段	group_id: 用户组 Id not null group_identityID: 用户组自增才 Id, 没有则返回-1 group_code: 用户组代码, 有则必须配置 GetGroupByCode , 否则返回用户组 Id group_thirdPartId: 返回用户组 Id group_name: 用户组名称 not null group_type: 用户组类型。0: 一般用户组; 1: Everyone 用户组; 2: 系统用户组; 3: 个人用户组; 4: 虚拟用户组 group_createTime: 用户组创建时间 group_creatorId: 用户组创建人 group_sort: 用户组排序编号 group_remark: 描述				
实例:					

3.2.45 根据用户组编号查询用户组

名称	GetGroupByCode	代码	4002	是否必配项:	否
描述	根据用户组代码查询用户组信息，如果第三方数据库没有此字段，则无须配置，此时用户组编号返回用户组id				
返回字段	group_id: 用户组 Id not null group_identityID: 用户组自增长 Id，没有则返回-1 group_code: 用户组代码，有则必须配置 GetGroupByCode ，否则返回用户组 Id group_thirdPartId: 返回用户组 Id group_name: 用户组名称 not null group_type: 用户组类型。0: 一般用户组；1: Everyone 用户组；2: 系统用户组；3: 个人用户组；4: 虚拟用户组 group_createTime: 用户组创建时间 group_creatorId: 用户组创建人 group_sort: 用户组排序编号 group_remark: 描述				
实例:					

3.2.46 根据用户组自增长 Id 查询用户组

名称	GetGroupByIdentityId	代码	4003	是否必配项:	否
描述	根据用户组自增长Id查询用户组信息，如group_identityId返回-1，则无须配置此项				
返回字段	group_id: 用户组 Id not null group_identityID: 用户组自增长 Id，没有则返回-1 group_code: 用户组代码，有则必须配置 GetGroupByCode ，否则返回用户组 Id group_thirdPartId: 返回用户组 Id group_name: 用户组名称 not null group_type: 用户组类型。0: 一般用户组；1: Everyone 用户组；2: 系统用户组；3: 个人用户组；4: 虚拟用户组 group_createTime: 用户组创建时间 group_creatorId: 用户组创建人 group_sort: 用户组排序编号 group_remark: 描述				
实例:					

3.2.47 查询所有的用户组

名称	GetAllGroup	代码	4005	是否必配项:	是
描述	查询所有的用户组信息				
返回字段	group_id: 用户组 Id not null group_identityID: 用户组自增长 Id，没有则返回-1 group_code: 用户组代码，有则必须配置 GetGroupByCode ，否则返回用户组 Id group_thirdPartId: 返回用户组 Id group_name: 用户组名称 not null				

	group_type: 用户组类型。0: 一般用户组; 1: Everyone 用户组; 2: 系统用户组; 3: 个人用户组; 4: 虚拟用户组 group_createTime: 用户组创建时间 group_creatorId: 用户组创建人 group_sort: 用户组排序编号 group_remark: 描述
实例:	

3.2.48 根据用户组 Id 查询用户

名称	GetUserByGroupId	代码	4006	是否必配项:	是
描述	根据用户组Id查询此用户组成员中的用户				
返回字段	同 1012				
实例:					

3.2.49 根据用户组 Id 查询部门

名称	GetDepartmentByGroupId	代码	4007	是否必配项:	否
描述	根据用户组Id查询此用户组成员中的部门				
返回字段	同代码 1001				
实例:					

3.2.50 根据用户组 Id 查询职位

名称	GetPositionByGroupId	代码	4008	是否必配项:	否
描述	根据用户组Id查询此用户组成员中的职位				
返回字段	同 2001				
实例:					

3.2.51 根据用户 Id 查询用户组

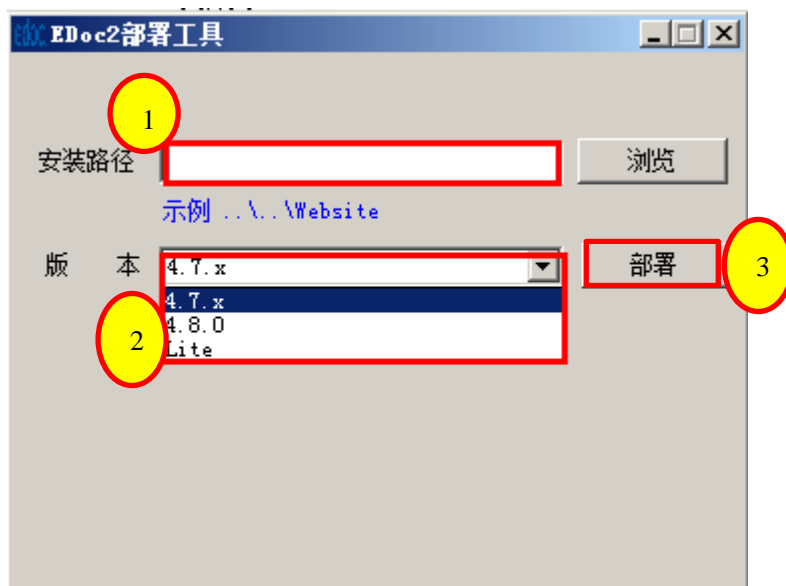
名称	GetUserGroups	代码	4009	是否必配项:	是
描述	根据用户Id查询此用户所在的用户组				
返回字段	group_id: 用户组 Id not null group_identityID: 用户组自增才 Id, 没有则返回-1 group_code: 用户组代码, 有则必须配置 GetGroupByCode , 否则返回用户组 Id group_thirdPartId: 返回用户组 Id group_name: 用户组名称 not null group_type: 用户组类型。0: 一般用户组; 1: Everyone 用户组; 2: 系统用户组; 3: 个人用户组; 4: 虚拟用户组 group_createTime: 用户组创建时间 group_creatorId: 用户组创建人 group_sort: 用户组排序编号				

	group_remark: 描述
实例:	

4 edoc2 SDK 集成工具介绍

4.1 edoc2SdkDeployTool 配置文件布置工具

4.1.1 工具窗口



4.1.2 使用方法

- ① EDoc2 安装路径的选择
- ② EDoc2 版本的选择
- ③ 点击部署 按钮

4.1.3 注意事项

EDoc2_Versions_4_7_x 文件夹

EDoc2_Versions_4_8_0 文件夹

EDoc2_Versions_Lite 文件夹 必须和

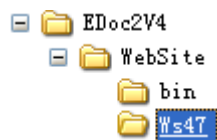
EDoc2SdkDeployTool.exe 在同一目录中

名称	修改日期	类型	大小
EDoc2_Versions_4_7_x	2013/8/15 8:50	文件夹	
EDoc2_Versions_4_8_0	2013/9/4 17:24	文件夹	
EDoc2_Versions_Lite	2013/10/23 18:06	文件夹	
EDoc2SdkDeployTool.exe	2013/10/23 18:06	应用程序	18 KB

5 附录

5.1 API 配置

5.1.1 edoc2 服务端部署



Bin\EDoc2.ApiProvider.dll

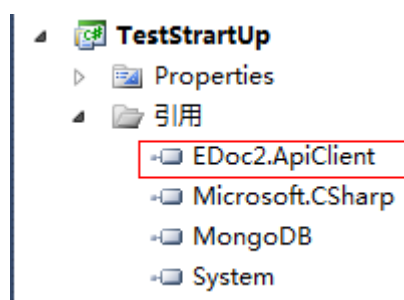
Ws47 目录下文件



分别复制到相应的 EDoc2 文件夹中

5.1.2 调用 API 项目端部署

5.1.2.1 引用类库



引用 EDoc2.ApiClientManage.dll

5.1.2.2 项目配置文件修改

5.1.2.2.1 添加 AppSetting 方式

```
0 <appSettings>
1   <add key="ApiServiceUrl" value="http://192.168.253.231/EDoc2v4"/>
2 </appSettings>
```

5.1.2.2.2 原生配置方式(红色为添加部分)

```

2 <configuration>
3   <configSections>
4     <sectionGroup name="applicationSettings" type="System.Configuration.ApplicationSettingsGroup, System, Version=2.0.0.0,
5       Culture=neutral, PublicKeyToken=b77a5c561934e089">
6       <section name="TestStartUp.Properties.Settings" type="System.Configuration.ClientSettingsSection, System, Version=4.0.0.0,
7         Culture=neutral, PublicKeyToken=b77a5c561934e089" requirePermission="false" />
8       <section name="EDoc2.ApiClient.Properties.Settings" type="System.Configuration.ClientSettingsSection, System,
9         Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" requirePermission="false"/>
10    </sectionGroup>
11  </configSections>
12  <applicationSettings>
13    <TestStartUp.Properties.Settings>
14      <setting name="TestStartUp_FileUpload_FileUploadService" serializeAs="String">
15        <value>http://localhost/edoc2v4/Ws404/FileUploadService.asmx</value>
16      </setting>
17    </TestStartUp.Properties.Settings>
18    <EDoc2.ApiClient.Properties.Settings>
19      <setting name="EDoc2_ApiClient_EDoc2UploadFileWebProvider_EDoc2UploadFileWebService" serializeAs="String">
20        <value>http://192.168.253.49/EDoc2v4/Ws47/EDoc2UploadFileWebService.asmx</value>
21      </setting>
22      <setting name="EDoc2_ApiClient_EDoc2OrganizationWebProvider_EDoc2OrganizationWebService" serializeAs="String">
23        <value>http://192.168.253.49/EDoc2v4/Ws47/EDoc2OrganizationWebService.asmx</value>
24      </setting>
25      <setting name="EDoc2_ApiClient_EDoc2DocumentWebProvider_EDoc2DocumentWebService" serializeAs="String">
26        <value>http://192.168.253.49/EDoc2v4/Ws47/EDoc2DocumentWebService.asmx</value>
27      </setting>
28      <setting name="EDoc2_ApiClient_EDoc2MetaWebProvider_EDoc2MetaWebService" serializeAs="String">
29        <value>http://192.168.253.49/EDoc2v4/Ws47/EDoc2MetaWebService.asmx</value>
30      </setting>
31    </EDoc2.ApiClient.Properties.Settings>
32  </applicationSettings>

```

```

<configuration>
  <configSections>
    <sectionGroup name="applicationSettings"
type="System.Configuration.ApplicationSettingsGroup, System,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" >
      <section name="EDoc2.ApiClientManage.Properties.Settings"
type="System.Configuration.ClientSettingsSection, System, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
requirePermission="false" />
      <section name="EDoc2.ApiClient.Properties.Settings"
type="System.Configuration.ClientSettingsSection, System, Version=2.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089"
requirePermission="false" />
    </sectionGroup>
  </configSections>
  <applicationSettings>
    <EDoc2.ApiClientManage.Properties.Settings>
      <setting
name="EDoc2_ApiClient_EDoc2UploadFileWebProvider_EDoc2UploadFileWebServi
ce"
      serializeAs="String">
<value>http://localhost/EDoc24/Ws47/EDoc2UploadFileWebService.asmx</valu
e>
      </setting>
      <setting
name="EDoc2_ApiClient_EDoc2OrganizationWebProvider_EDoc2OrganizationWebS
ervice"
      serializeAs="String">
<value>http://localhost/EDoc24/Ws47/EDoc2OrganizationWebService.asmx</va
lue>

```

```
</setting>
<setting
name="EDoc2_ApiClient_EDoc2DocumentWebProvider_EDoc2DocumentWebService"
    serializeAs="String">

<value>http://localhost/EDoc24/Ws47/EDoc2DocumentWebService.asmx</value>
</setting>
<setting
name="EDoc2_ApiClient_EDoc2MetaWebProvider_EDoc2MetaWebService"
    serializeAs="String">

<value>http://localhost/EDoc24/Ws47/EDoc2MetaWebService.asmx</value>
</setting>
</EDoc2.ApiClientManage.Properties.Settings>
<EDoc2.ApiClient.Properties.Settings>
    <setting
name="EDoc2_ApiClient_EDoc2UploadFileWebProvider_EDoc2UploadFileWebServ
ice"
    serializeAs="String">

<value>http://localhost/EDoc24/Ws47/EDoc2UploadFileWebService.asmx</valu
e>

</setting>
<setting
name="EDoc2_ApiClient_EDoc2OrganizationWebProvider_EDoc2OrganizationWebS
ervice"
    serializeAs="String">
    <value>EDoc2应用地址
/Ws47/EDoc2OrganizationWebService.asmx</value>
</setting>
<setting
name="EDoc2_ApiClient_EDoc2DocumentWebProvider_EDoc2DocumentWebService"
    serializeAs="String">
    <value>EDoc2应用地址
/Ws47/EDoc2DocumentWebService.asmx</value>
</setting>
<setting
name="EDoc2_ApiClient_EDoc2MetaWebProvider_EDoc2MetaWebService"
    serializeAs="String">
    <value>EDoc2应用地址/Ws47/EDoc2MetaWebService.asmx</value>
</setting>
</EDoc2.ApiClient.Properties.Settings>
</applicationSettings>
</configuration>
```

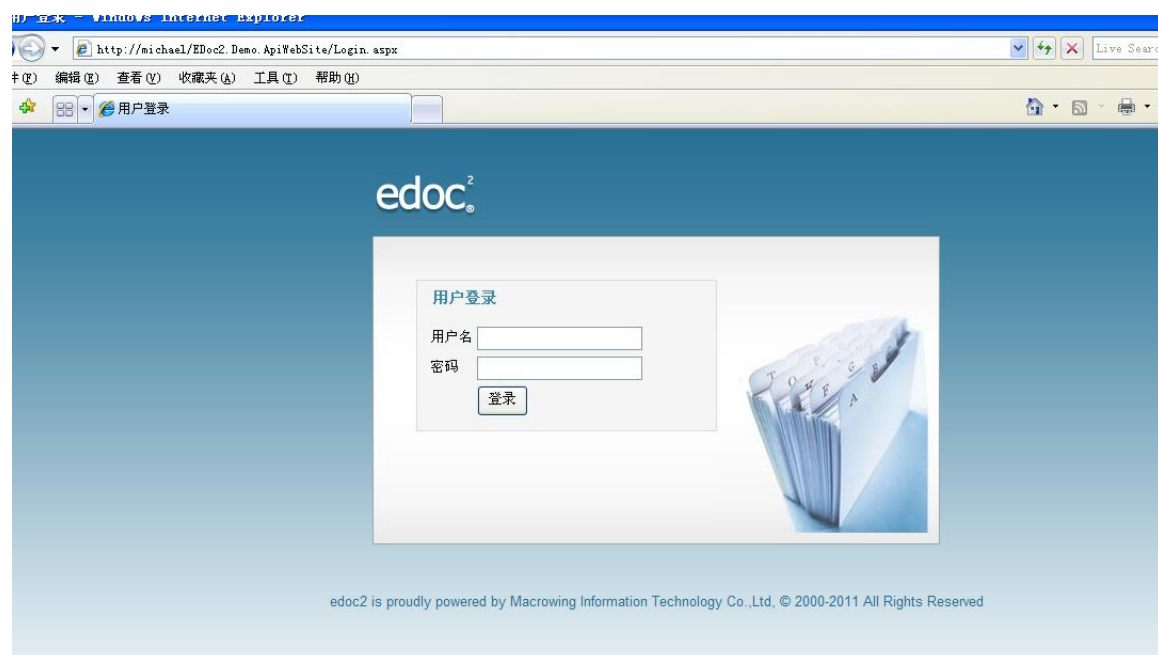
5.2 API Demo(仅适用 ECM)

5.2.1 配置

修改配置文件中的地址

5.2.2 API 预览

5.2.2.1 用户登录



输入用户名和密码进入

5.2.2.2 主界面

edoc² | SDK 开发包示例 v1.0

用户操作

文件操作

文件夹操作

组织操作

权限操作

用户信息

属性	内容
用户编号	3
用户登录名	test01
用户注册时间	Mon Oct 24 17:53:43 UTC+0800 2011
用户过期时间	Tue Jan 1 08:00:00 UTC+0800 1901
用户最后登录时间	Sun May 13 21:10:48 UTC+0800 2012
用户最后登录IP	192.168.253.118
用户邮箱	lxwin02@126.com
用户状态	1
部门编号	50
部门名称	Test
用户文件夹	0
用户编码	test01
用户排序	0

5.2.2.3 上传文件

控件上传文件

选择上传目录

hello

选择目录

选择上传文件

E:\No.8 [NTFS MyData]\根目录\娱乐

浏览文件

开始上传

7.00%

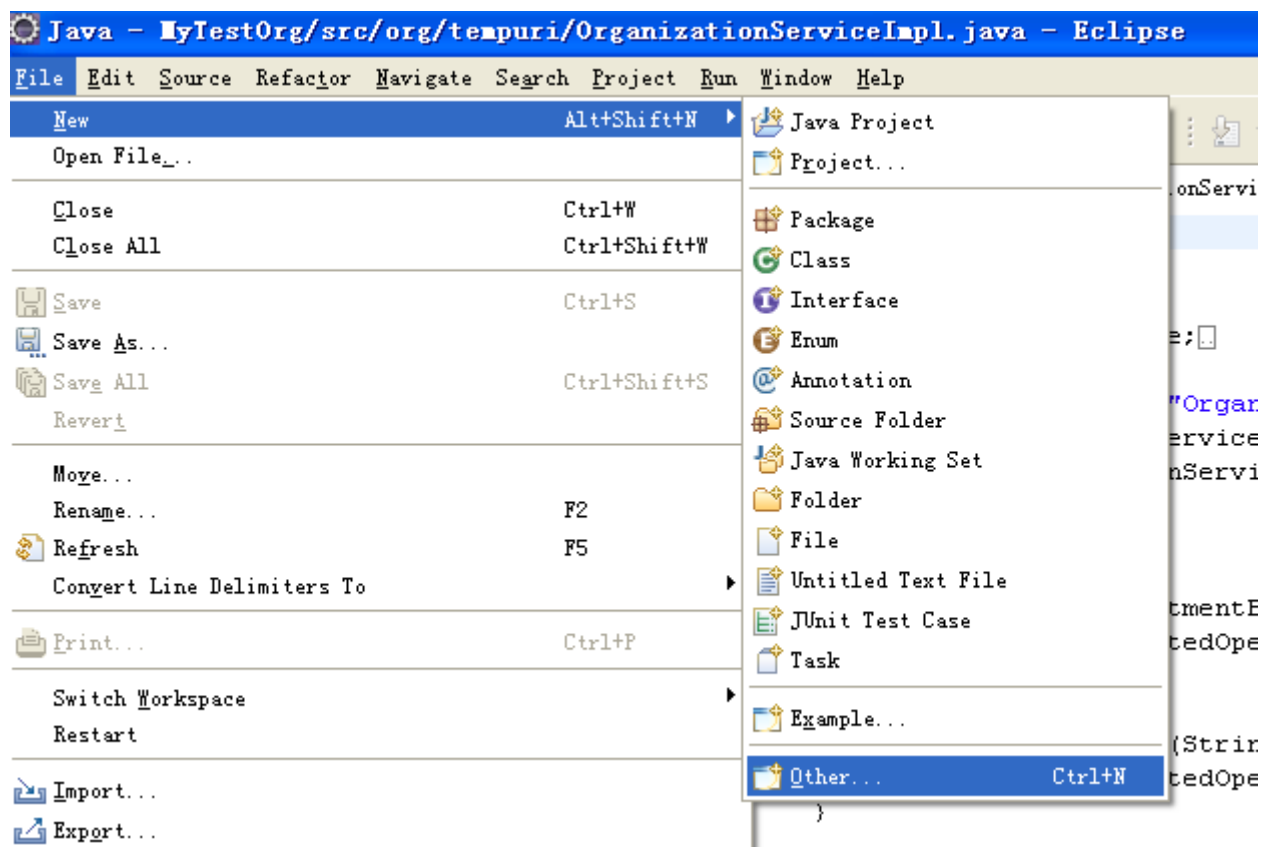
5.3 Java 调用

5.3.1 在 Eclipse 中安装 XFire 插件

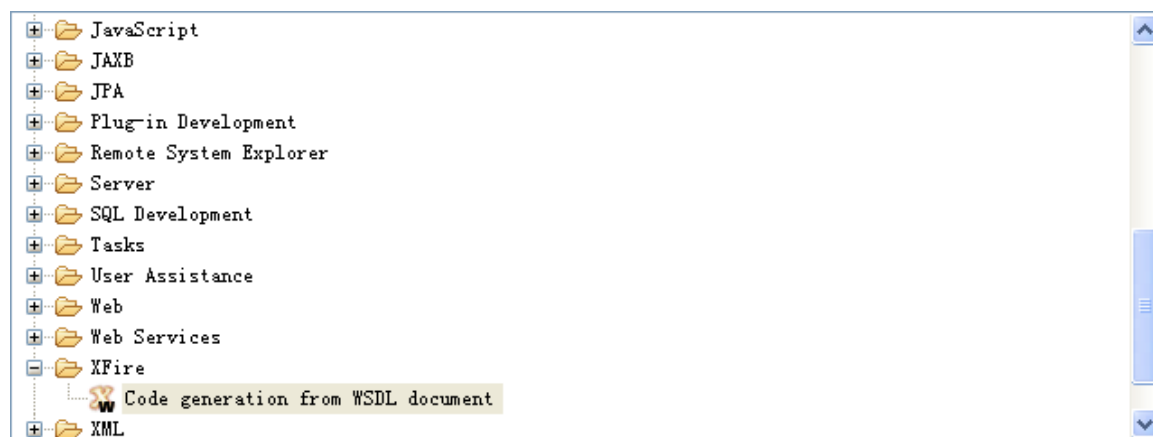
在 Help 中 Install new Software 查找 xfire 进行安装

参考:<http://xfire.codehaus.org/Eclipse+Plugin>

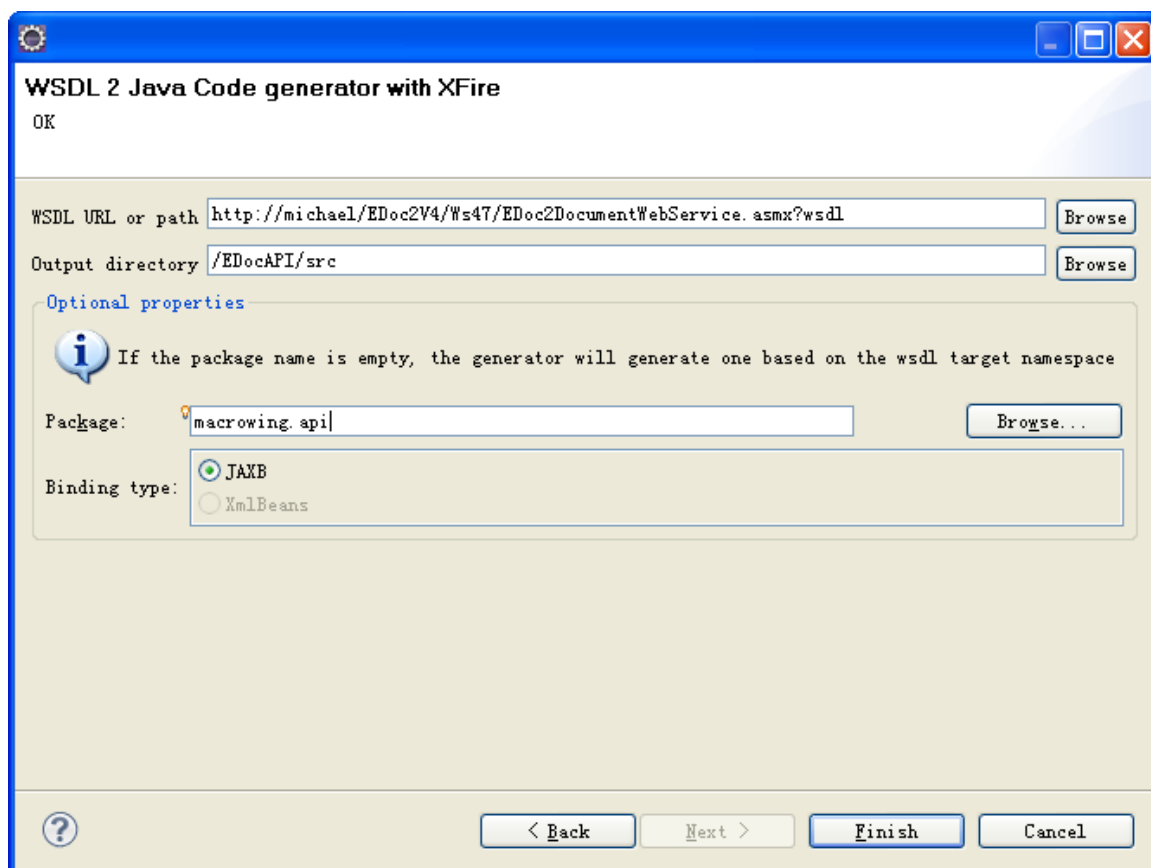
5.3.2 创建项目



选择“Other...”

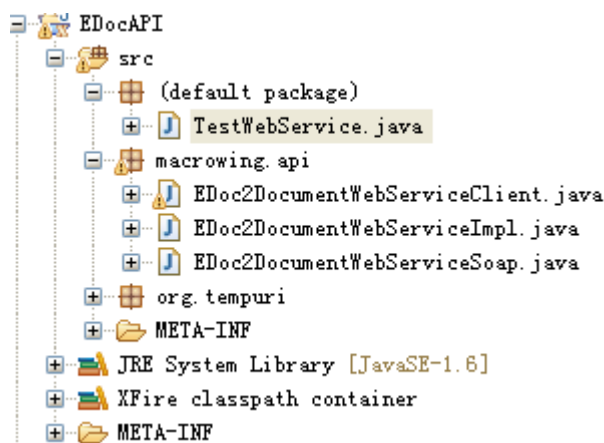


选择“XFire”项目



假设引用 EDoc2 文档操作的 API 写好地址
选择创建的项目 “Finish” 完成

自动创建调用类



New Class 用 Java 开始调用

```
public class TestWebService {  
  
    /**  
     * @param args  
     */  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        macrowing.api.EDoc2DocumentWebServiceClient client=new macrowing.api.EDoc  
        macrowing.api.EDoc2DocumentWebServiceSoap soap=client.getEDoc2DocumentWeb  
        soap.  
    }  
}
```

