微签电子签章平台

Web API 接口说明

FORWAVE

上海复园电子科技有限公司

目录

1. 简介

微签电子签章平台（WeiQianSeal）是上海复园电子科技有限公司根据多年的办公自动化、业务文档流转处理经验，开发的一套智能安全电子签章软件。它以版式文件为基础，实现对文档全文批注、电子印章、安全文档等功能。

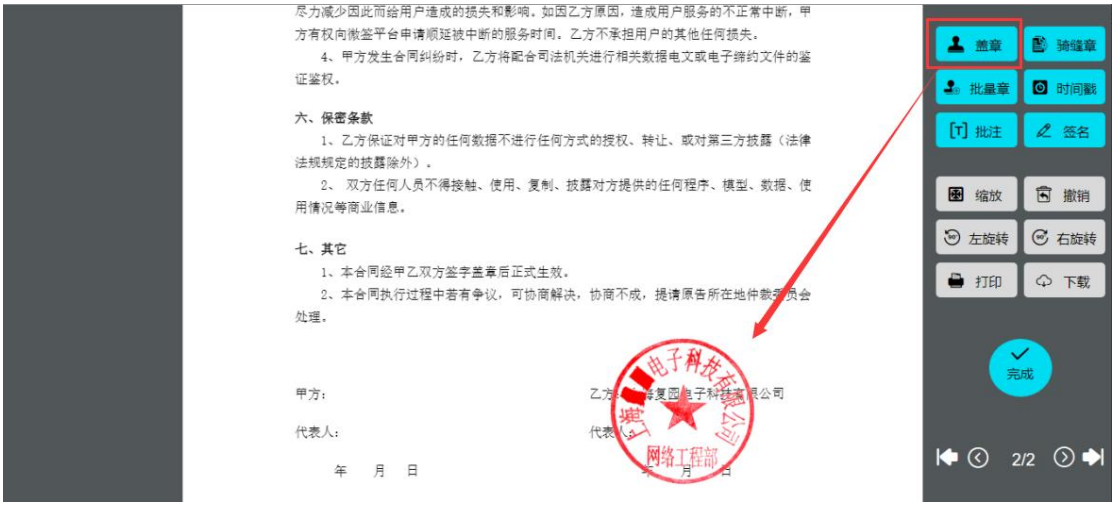
WeiQian Web API 接口，可以很方便地与各种 Web 环境相整合，与协同办公平台、电子政务平台、OA 办公自动化系统、ERP 系统集成，打通无纸化办公的最后一个环节。

2. 电脑端——手动签章接口

2.1. 手动签章效果

- **手动盖章**

手动盖章，是手动将电子印章拖放到文件的任意位置加盖的操作。单页盖章样式：



- **签名**

微签既可以对电脑端的文件在线签字，也可在移动端 APP 或小程序中签字。

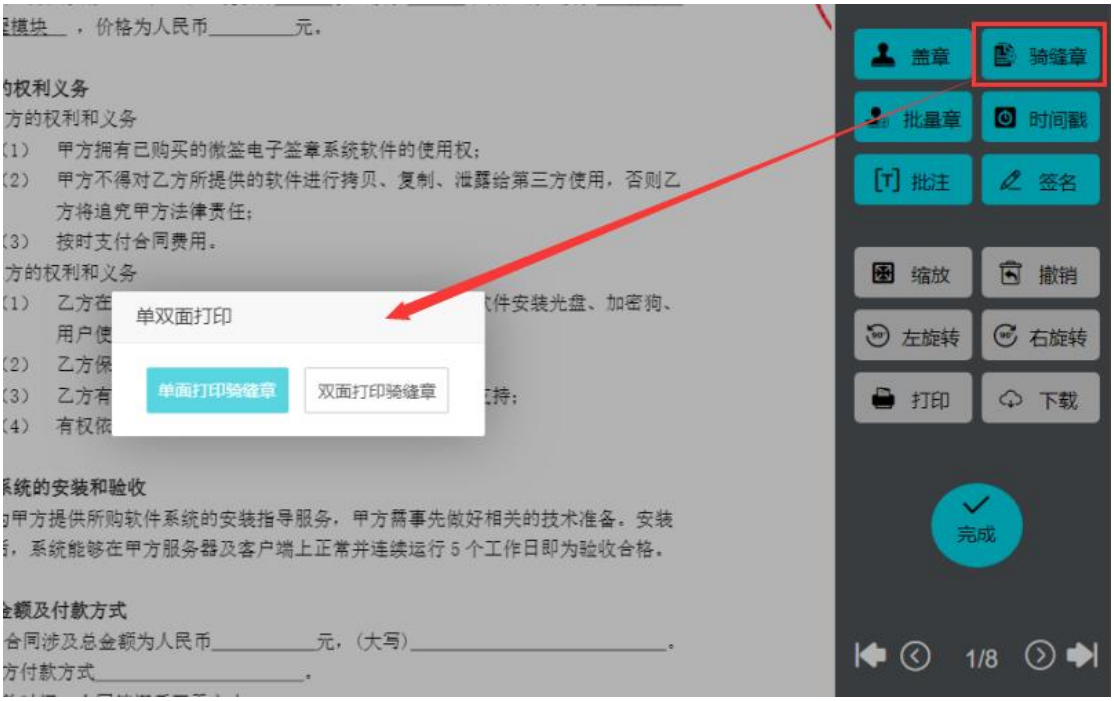
若在电脑端操作，则打开文件签章页面，选择“签名”，会生成二维码：

用手机微信扫描，在手机手写签名：

签好的字，会在文件上生成，可以随意拖动，调整大小：

- **骑缝章**

盖骑缝章是为了防范风险，当签署的合同页数比较多，担心合同内容被更换，就需要骑缝章的加盖来保证合同文件的完整性。一个印章会切分成若干均块分在每一页的侧面。



印章均分在每一页文件的上面，骑缝章样式：

- **批量章**

如果大量的文件需要在**相同位置盖相同章**，可以使用“批量章”功能，节省大量的时间，提高工作效率。

批量章样式：

- **时间戳**

在文件上添加当前签署的时间，就是“时间戳”功能的作用。时间戳可由管理员预先设置样式，使用者选择样式。

时间戳支持拖动，样式如下：

- **批注**

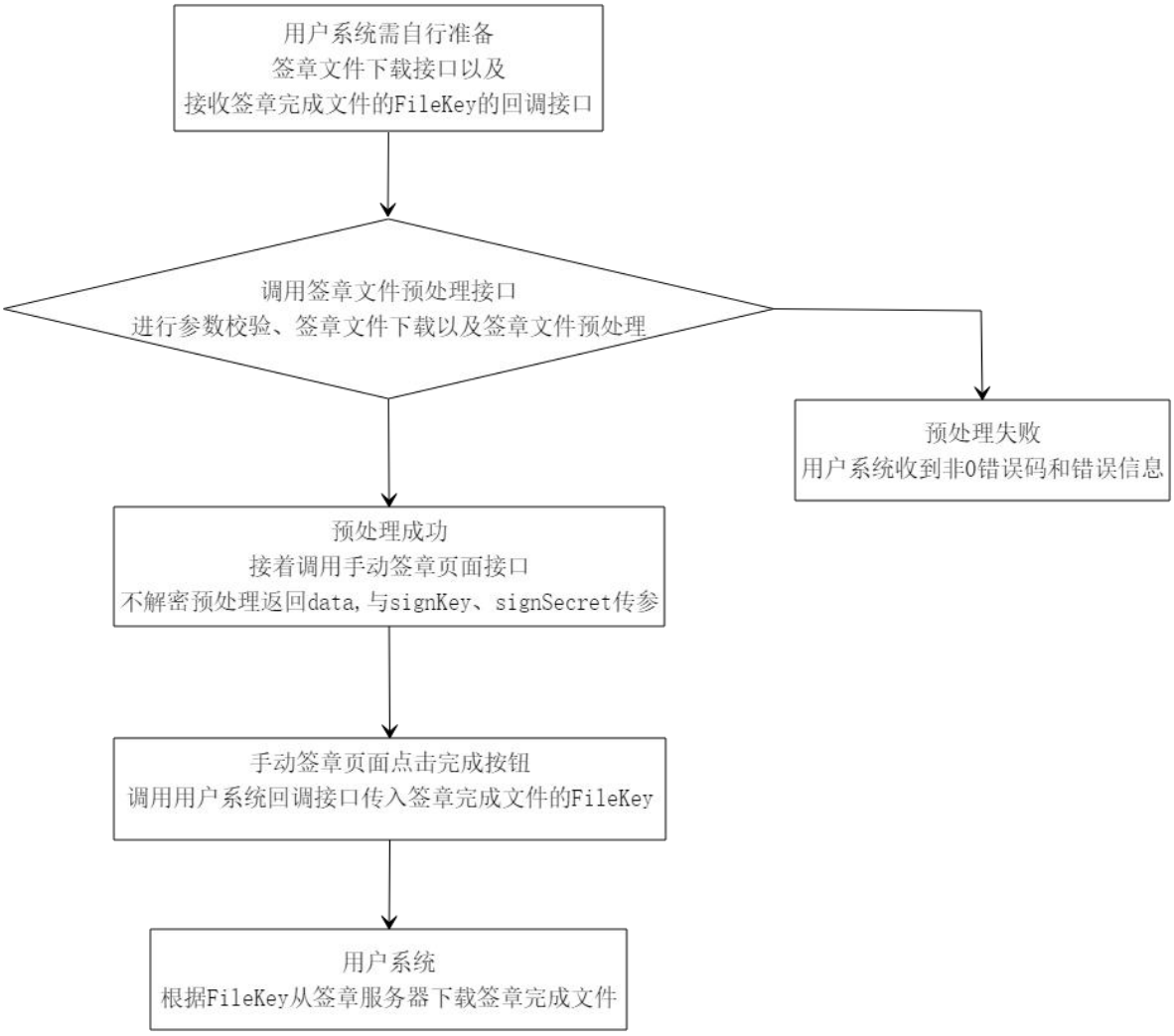
如果要在签署文件的正文中添加内容，则可使用“批注”功能，可在文件上添加任意文本内容，并可以设置字体、大小、颜色。

批注同样支持拖动，可以添加在任意位置。

2.2. 实现原理

- 用户系统需自行提供签章文件下载接口和签章完成结果回调接口。
- 先调用签章文件预处理接口，进行签章文件预处理。
- 预处理成功完成返回后，使用返回的 data（不解密直接传参）和相应参数传参再调用手动签章接口，跳转到签章服务器手动签章页面。
- 手动签章页面点击完成按钮，签章服务器将回调用户系统提供的签章完成接口，向客户端回传签章完成文件 filekey。
- 用户系统使用 fileKey，调用签章服务器的下载接口，即可下载签章完成文件。

2.3. 流程说明



2. 4. 手动签章接口描述

2.4.1. 签章文件预处理接口：

http://WeiQianSealURL:9999/manualSig/preprocess
请求方式：POST
参数格式：JSON
返回格式：JSON

2.4.1.1. 预处理接口请求参数说明（入参）

参数名	说明	必选
signKey	接口签名 key	Yes
signSecret	接口签名秘钥	Yes
sealUser	规则创建用户	Yes
password	用户密码	Yes
provideSigFile	签章文件下载地址	Yes
getSigFile	手动签章完成后的回调接口	Yes
fileKey	签章文件 key 或对应的任务 ID	No

- 所有参数需为 Base64 编码，中文编码为 UTF-8，签章平台编码解码类为 sun.misc.BASE64Decoder 和 sun.misc.BASE64Encoder。
- signKey 和 signSecret 由我司商务部提供。
- 用户系统提供下载的签章文件，要指定签章文件类型，以下三种指定方式必须具备其一。

  指定方式一，使用 fileKey 参数，并且参数中指定文件类型，如：FWBPDtask000000001.pdf。

  指定方式二，在提供的下载地址中指定。如：

  http://192.168.1.31:8080/download?id=FWBPDtask000000001.pdf

  指定方式三，下载接口代码中指定，使用 response.addHeader("Content-Disposition", "attachment;filename=FWBPDtask000000001.pdf")方式。

- 非必须参数 fileKey，是为用户系统把签章结果和自己的系统任务对应起来而设计的。如果请求参数中提供了 fileKey，那么此请求的签章结果也将返回 fileKey；如果请求参数中没有提供，那么签章结果将返回一串随机序号。fileKey 建议只使用英文和数字，最好包含用户系统的编号或名称，以及当前任务的 ID。如：FWBPDtask000000001202107121412
- 签章文件不是 pdf 类型时，有下列非必须参数（传参时不用 Base64 编码）。当参数为空时，将取系统参数设置值。

参数名	说明	必选
sealFileCharSetName	签章文件编码	No
marginLeft	签章文件转换 PDF 左边距	No
marginRight	签章文件转换 PDF 右边距	No
marginTop	签章文件转换 PDF 上边距	No
marginBottom	签章文件转换 PDF 下边距	No

参数名称	参数值	更新人
用户初始密码	1234	system
签章初始密码	1234	system
数字证书签章原因	微签用章	system
数字证书实体	复园电子科技	system
数字证书实体网址	www.forwave.com	system
签章文件编码	GBK	system
签章文件转换PDF左边距	15	system
签章文件转换PDF右边距	15	system
签章文件转换PDF上边距	30	system
签章文件转换PDF下边距	30	system

2.4.1.2. 预处理接口返回参数说明（出参）

参数名	说明
code	返回的状态码，0 是成功，非 0 是错误码
msg	返回的说明信息
data	手动签章页面 Key
timestamp	返回结果的时间戳

- 当预处理成功时，返回 data 是加密的手动签章页面 key。无需解密，直接使用此数据传参调用手动签章页面接口。

2.4.2. 手动签章页面接口：

http://WeiQianSealURL:9999/manualSig/manualSigPage
请求方式：GET
返回格式：JSON

2.4.2.1. 签章页面请求参数说明（入参）

signKey、signSecret 要求 Base64 编码。

sigFile 为预处理接口成功完成返回 data，不需要 Base64 编码。

参数名	说明	必选
signKey	接口签名 key	Yes
signSecret	接口签名秘钥	Yes
sigFile	预处理接口成功返回 data	Yes

2.4.3. 签章完成文件下载接口

http://WeiQianSealURL:9999/no/getSealFile/FileKey
请求方式：GET
返回：签章完成文件

注意：使用 fileKey 下载签章完成文件时，当下载完成后，此 fileKey 将失效。

2.5. 返回参数示例

{"code":"0",

"msg":"手动签章文件预处理完成！",

"timestamp":1575461457169,

"data":"fe52a861b1eb4739bbb50cd1c73d72cd=="}

{"code":"4008",

"msg":"手动签章验签参数有空值！",

"timestamp":1575460560586,

"data":null}

2.6. 手动签章请求示例

```
function doManualSig() {
        //手动签章请求入参
        let _sealUser = btoa($("#sealUser").val());
        let _signKey = btoa($("#signKey").val());
        let _password = btoa($("#password").val());
        let _signSecret = btoa($("#signSecret").val());
        let _fileKey = btoa($("#fileKey").val());
        let _provideSigFile = btoa(($("#provideSigFile").val()));
        let _getSigFile = btoa($("#getSigFile").val());
        $.ajax( {
            url: _wqSigServer + '/manualSig/preprocess',
            data: JSON.stringify({
                sealUser : _sealUser,
                password : _password,
                signKey : _signKey,
                signSecret : _signSecret,
                fileKey : _fileKey,
                provideSigFile : _provideSigFile,
                getSigFile : _getSigFile
            }),
            type:'post',
            dataType:'json',
            contentType: "application/json;charset=UTF-8",
            success:function(result) {
                if(result.code == 0){
```

```
                let sigUrl = _wqSigServer +'/manualSig/manualSigPage?sigFile=' +
result.data + '&signKey=' + _signKey + '&signSecret=' + _signSecret;
                if(app=="app"){
                    let _pageType = btoa('1');
                    sigUrl = _wqSigServer +'/manualSig/manualSigPage?sigFile='+
result.data + '&pageType=' + _pageType + '&signKey=' +_signKey + '&signSecret=' + _signSecret;
                }
                getManualSigPage(sigUrl);
            }else{
                console.log(result.msg);
            }
        },
    });
}
```

2.7. 签章页面完成按钮回调接口示例

- 示例中完成按钮回调接口传参：http://192.168.1.29:8080/doSeal/getSigFile
- 签章服务器通过回调接口出参（sigFile），返回已签章文件 fileKey。
- 出参（sigCount）返回用户本次操作的签章数目。如值为 0 则表示用户没有签章就点击完成按钮返回。

```
@RequestMapping(value = "/getSigFile")
    public ModelAndView getSigFile(@RequestParam("sigFile") String sigFile,
                                   @RequestParam("sigCount") Integer sigCount) {
        String sigFilePrefix = wqSigServer + "/no/getSealFile/";
        return new ModelAndView("redirect:/showSigFile.html?sigFilePrefix="
                + sigFilePrefix + "&sigFile=" + sigFile + "&sigCount=" + sigCount);
    }
```

2.8. 已签章文件下载接口

接口地址：http://WeiQianSealURL:9999/no/getSealFile/fileKey
返回：签章完成文件
请求方式：GET

注意：使用 fileKey 下载签章完成文件时，当下载完成后，此 fileKey 将失效。

后台请求下载示例：

```
String downloadUrl = "http://192.168.1.30:9999/no/"
                + "getSealFile/fe52a861b1eb4739bbb50cd1c73d72cd.pdf";
        String desFile = "D:/temp/seal/对账单_20200711.pdf";
        int byteRead = 0;
    try {
        URL url = new URL(downloadUrl);
        URLConnection connection    = url.openConnection();
            InputStream input = connection.getInputStream();
            FileOutputStream output = new FileOutputStream(desFile);
            byte[] buffer = new byte[1024];
            while((byteRead = input.read(buffer)) != -1){
            output.write(buffer, 0, byteRead);
            }
            output.flush();
            output.close();
    } catch (MalformedURLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
```

前端下载文件并显示：

```
success:function(result){
                    if(result.code == 0){
                        window.location.href = 'http://192.168.1.30:9999/no/getSealFile' + result.data;
                    }else{
                        alert(result.msg);
                    }
                },
```

注意：fileKey 只支持一次下载，下载完成后 fileKey 将失效！

3. 移动端——手动签章接口

3.1. 移动端签章效果

业务系统调用微签移动端接口，在手机上完成盖章、签字。
移动端签章工具菜单：

签章效果：

3.2. 实现原理

- 用户系统需自行提供签章文件下载接口和签章完成结果回调接口。
- 先调用签章文件预处理接口，进行签章文件预处理。
- 预处理成功完成返回后，使用返回的 data（不解密直接传参）和相应参数传参再调用移动端手动签章接口，跳转到签章服务器移动端手动签章页面。
- 手动签章页面点击完成按钮，签章服务器将回调用户系统提供的签章完成接口，向客户端回传签章完成文件 filekey。
- 用户系统使用 fileKey，调用签章服务器的下载接口，即可下载签章完成文件。

注：移动端手动签章接口与 PC 端手动签章接口的区别在于，调用手动签章页面时的参数不同。

3. 3. 移动端手动签章接口描述

3.3.1. 签章文件预处理接口：

http://WeiQianSealURL:9999/manualSig/preprocess
请求方式：POST
参数格式：JSON
返回格式：JSON

3.3.1.1. 预处理接口请求参数说明（入参）

参数名	说明	必选
signKey	接口签名 key	Yes
signSecret	接口签名秘钥	Yes
sealUser	规则创建用户	Yes
password	用户密码	Yes
provideSigFile	签章文件下载地址	Yes
getSigFile	手动签章完成后的回调接口	Yes
fileKey	签章文件 key 或对应的任务 ID	No

- 所有参数需为 Base64 编码，中文编码为 UTF-8，签章平台编码解码类为 sun.misc.BASE64Decoder 和 sun.misc.BASE64Encoder。
- signKey 和 signSecret 由我司商务部提供。
- 用户系统提供下载的签章文件，要指定签章文件类型，以下三种指定方式必须具备其一。

  指定方式一，使用 fileKey 参数，并且参数中指定文件类型，如：FWBPDtask000000001.pdf。

  指定方式二，在提供的下载地址中指定。如：

  http://192.168.1.31:8080/download?id=FWBPDtask000000001.pdf

  指定方式三，下载接口代码中指定，使用 response.addHeader("Content-Disposition", "attachment;filename=FWBPDtask000000001.pdf")方式。
- 非必须参数 fileKey，是为用户系统把签章结果和自己的系统任务对应起来而设计的。如果请求参数中提供了 fileKey，那么此请求的签章结果也将返回 fileKey；如果请求参数中没有提供，那么签章结果将返回一串随机序号。fileKey 建议只使用英文和数字，最好包含用户系统的编

号或名称，以及当前任务的 ID。如：FWBPDtask000000001202107121412

- 签章文件不是 pdf 类型时，有下列非必须参数（传参时不用 Base64 编码）。当参数为空时，将取系统参数设置值。

参数名	说明	必选
sealFileCharSetName	签章文件编码	No
marginLeft	签章文件转换 PDF 左边距	No
marginRight	签章文件转换 PDF 右边距	No
marginTop	签章文件转换 PDF 上边距	No
marginBottom	签章文件转换 PDF 下边距	No

参数名称	参数值	更新人
用户初始密码	1234	system
签章初始密码	1234	system
数字证书签章原因	微签用章	system
数字证书实体	复园电子科技	system
数字证书实体网址	www.forwave.com	system
签章文件编码	GBK	system
签章文件转换PDF左边距	15	system
签章文件转换PDF右边距	15	system
签章文件转换PDF上边距	30	system
签章文件转换PDF下边距	30	system

3.3.1.2. 预处理接口返回参数说明（出参）

参数名	说明
code	返回的状态码，0 是成功，非 0 是错误码
msg	返回的说明信息

参数名	说明
data	移动端手动签章页面 key
timestamp	返回结果的时间戳

3.3.2. 移动端手动签章页面接口：

http://WeiQianSealURL:9999/manualSig/manualSigPage
请求方式：GET
返回格式：JSON

3.3.2.1. 签章页面请求参数说明（入参）

signKey、signSecret、pageType 要求 Base64 编码。

sigFile 为预处理接口成功完成返回 data，不需要 Base64 编码。

参数名	说明	必选
signKey	接口签名 key	Yes
signSecret	接口签名秘钥	Yes
pageType	移动端签章页面值为 1	Yes
sigFile	预处理接口成功返回 data	Yes

- 所有参数需为 Base64 编码，中文编码为 UTF-8，签章平台编码解码类为 sun.misc.BASE64Decoder 和 sun.misc.BASE64Encoder。
- signKey 和 signSecret 由我司商务部提供。

3. 4. 移动端手动签章请求示例

```
function doManualSig() {
        //手动签章请求入参
        let _sealUser = btoa($("#sealUser").val());
        let _signKey = btoa($("#signKey").val());
        let _password = btoa($("#password").val());
        let _signSecret = btoa($("#signSecret").val());
        let _fileKey = btoa($("#fileKey").val());
        let _provideSigFile = btoa(($("#provideSigFile").val()));
```

```
            let _getSigFile = btoa($("#getSigFile").val());
            $.ajax( {
                url: _wqSigServer + '/manualSig/preprocess',
                data: JSON.stringify({
                    sealUser : _sealUser,
                    password : _password,
                    signKey : _signKey,
                    signSecret : _signSecret,
                    fileKey : _fileKey,
                    provideSigFile : _provideSigFile,
                    getSigFile : _getSigFile
                }),
                type:'post',
                dataType:'json',
                contentType: "application/json;charset=UTF-8",
                success:function(result) {
                    if(result.code == 0){
                        let sigUrl = _wqSigServer +'/manualSig/manualSigPage?sigFile=' +
result.data + '&signKey=' + _signKey + '&signSecret=' + _signSecret;
                        if(app=="app"){
                            let _pageType = btoa('1');
                            sigUrl = _wqSigServer +'/manualSig/manualSigPage?sigFile='+
result.data + '&pageType=' + _pageType + '&signKey=' +_signKey + '&signSecret=' + _signSecret;
                        }
                        getManualSigPage(sigUrl);
                    }else{
                        console.log(result.msg);
                    }
                },
            });
        }
```

3.5. demo 中的回调接口示例

使用 http://192.168.1.29:8080/doSeal/getSigFile 来接收手动签章完成后返回的 FileKey。

```
    @RequestMapping(value = "/getSigFile")
        public ModelAndView getSigFile(@RequestParam("sigFile") String sigFile,
                                       @RequestParam("sigCount") Integer sigCount) {
            String sigFilePrefix = wqSigServer + "/no/getSealFile/";
            return new ModelAndView("redirect:/showSigFile.html?sigFilePrefix="
                    + sigFilePrefix + "&sigFile=" + sigFile + "&sigCount=" + sigCount);
        }
```

4. 电脑端——自动签章接口

自动签章接口是签章平台与各系统对接，实现无人工干预自动盖章的接口。

4.1. 实现原理

第一步：用户系统自行准备签章文件下载接口。

第二步：用户系统向签章平台发送自动签章请求。

第三步：请求参数中包含签章平台已建立的自动签章规则名称。

第四步：签章平台响应请求，下载签章文件并调用指定的签章规则进行自动签章。

第五步：签章平台返回签章完成文件 FileKey。

第六步：用户系统根据 FileKey，调用签章服务器下载接口，下载签章完成文件。

备注：签章规则的类型有如下 7 种：

- **指定坐标单页盖章**
- **指定坐标多页盖章**
- **首位关键字盖章**
- **全部关键字盖章**
- **骑缝章**
- **指定坐标单页时间戳**
- **指定坐标多页时间戳**

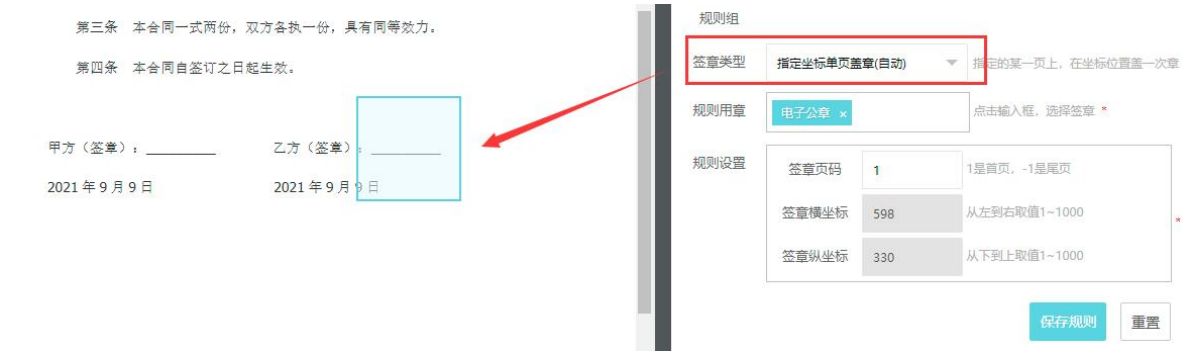
4.2. 签章规则设置

在签章服务器后台，可设置各种签章规则，签章规则可任意组合使用：

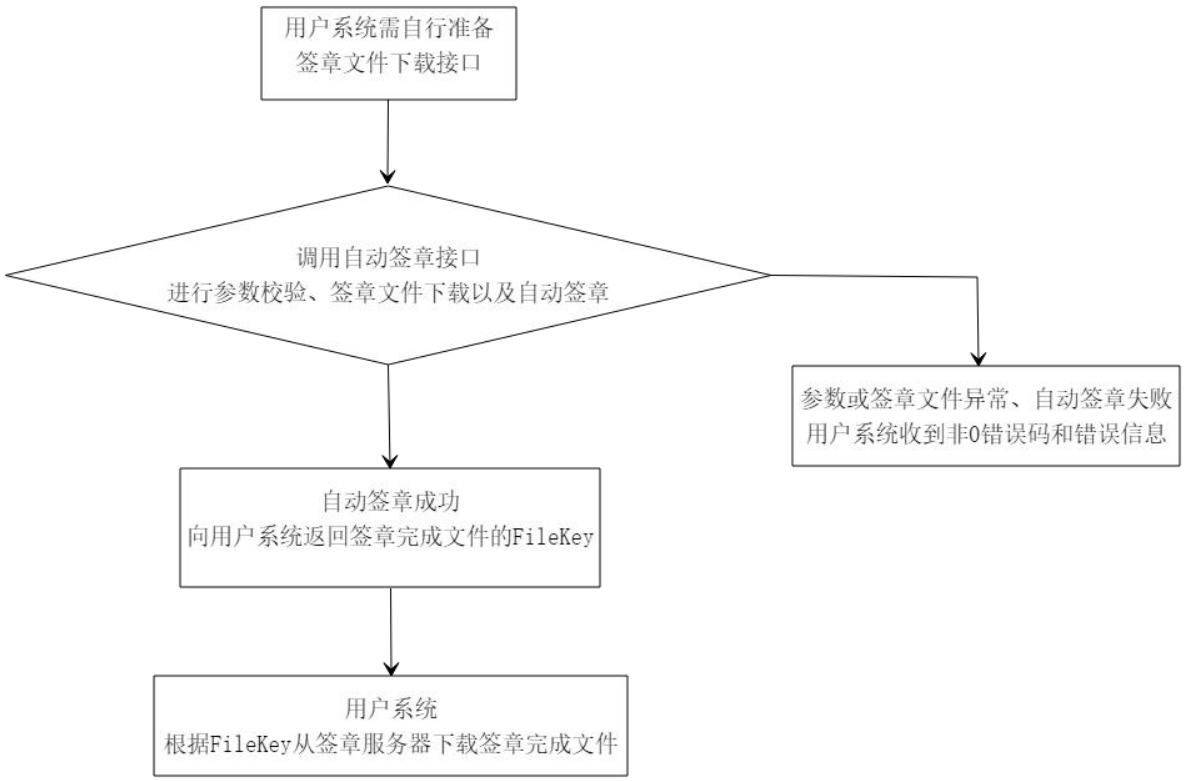
在下方图片的红色方框内设置签章的签章方式：

如，可全文搜索关键词的位置自动盖章：

如，在指定坐标位置自动盖章：



4.3. 流程说明



4.4. 自动签章接口描述

接口地址：http://WeiQianSealURL:9999/autoSeal/doSeal
请求方式：POST
参数格式：JSON
返回格式：JSON

4.5. 请求参数说明（入参）

参数名	说明	必选
signKey	接口签名 key	Yes
signSecret	接口签名秘钥	Yes
sealUser	规则用章授权用户	Yes
password	规则用章授权用户密码	Yes

ruleName	规则名称	Yes
provideSigFile	需签章的文件下载地址	Yes
fileKey	签章文件 key 或对应的任务 ID	No

- 所有参数需为 Base64 编码，中文编码为 UTF-8，签章平台编码解码类为 sun.misc.BASE64Decoder 和 sun.misc.BASE64Encoder。
- signKey 和 signSecret 由我司商务部提供。
- 用户系统提供下载的签章文件，要指定签章文件类型，以下三种指定方式必须具备其一。

  指定方式一，使用 fileKey 参数，并且参数中指定文件类型，如：FWBPDtask000000001.pdf。

  指定方式二，在提供的下载地址中指定。如：

  http://192.168.1.31:8080/download?id=FWBPDtask000000001.pdf

  指定方式三，下载接口代码中指定，使用 response.addHeader("Content-Disposition", "attachment;filename=FWBPDtask000000001.pdf")方式。
- 非必须参数 fileKey，是为用户系统把签章结果和自己的系统任务对应起来而设计的。如果请求参数中提供了 fileKey，那么此请求的签章结果也将返回 fileKey；如果请求参数中没有提供，那么签章结果将返回一串随机序号。fileKey 建议只使用英文和数字，最好包含用户系统的编号或名称，以及当前任务的 ID。如：FWBPDtask0000000012021071214 12
- 签章文件不是 pdf 类型时，有下列非必须参数（传参时不用 Base64 编码）。当参数为空时，将取系统参数设置值。

参数名	说明	必选
sealFileCharSetName	签章文件编码	No
marginLeft	签章文件转换 PDF 左边距	No
marginRight	签章文件转换 PDF 右边距	No
marginTop	签章文件转换 PDF 上边距	No
marginBottom	签章文件转换 PDF 下边距	No

参数名称	参数值	更新人
用户初始密码	1234	system
签章初始密码	1234	system
数字证书签章原因	微签用章	system
数字证书实体	复园电子科技	system
数字证书实体网址	www.forwave.com	system
签章文件编码	GBK	system
签章文件转换PDF左边距	15	system
签章文件转换PDF右边距	15	system
签章文件转换PDF上边距	30	system
签章文件转换PDF下边距	30	system

4.6. 自动签章请求示例

后台请求方式：

```
public SealResponse autoSealReqParam() {
        Map<String,String> params = new HashMap<String,String>();
        String requestUrl = "http://192.168.1.29:9999/autoSeal/doSeal";
        String sealUser64 = Base64.base64Encode("admin");
        String password64 = Base64.base64Encode("999999");
        String signKey64 = Base64.base64Encode("WQS20214205b6b");
        String signSecret64 = Base64.base64Encode("9999999999999999");
        String ruleName64 = Base64.base64Encode("rule1");
        String sealFileUrl64 =
Base64.base64Encode("http://192.168.1.30:8080/doSeal/getHandledFile/h1.pdf");
        params.put("sealUser", sealUser64);
        params.put("password", password64);
        params.put("signKey", signKey64);
        params.put("signSecret", signSecret64);
        params.put("ruleName", ruleName64);
        params.put("sealFileUrl", sealFileUrl64);

        //调用 httpRequest 方法，这个方法主要用于请求地址，并加上请求参数
        try {
            URL url = new URL(requestUrl);
            HttpURLConnection connection = (HttpURLConnection) url.openConnection();
            connection.setDoOutput(true);
```

```
            connection.setDoInput(true);
            connection.setRequestMethod("POST");
            connection.setUseCaches(false);
            connection.setInstanceFollowRedirects(true);
            connection.setRequestProperty("Content-Type", "application/json");
            connection.connect();
            ObjectMapper mapper = new ObjectMapper();
            String json = mapper.writeValueAsString(params);
            BufferedWriter writer = new BufferedWriter(new
OutputStreamWriter(connection.getOutputStream(), "UTF-8"));
            writer.write(json);
            writer.close();
            int responseCode = connection.getResponseCode();
            if(responseCode == HttpURLConnection.HTTP_OK){
                InputStream inputStream = connection.getInputStream();
                return mapper.readValue(inputStream, SealResponse.class);
            }
        } catch (Exception e) {
            e.printStackTrace();
            logger.error("自动签章请求异常！ ");
        }
        return null;
    }

 public static String base64Encode(String plainTxt) {
        try {
            return new BASE64Encoder().encode(plainTxt.getBytes("UTF-8"));
        } catch (UnsupportedEncodingException e) {
            e.printStackTrace();
            return null;
        }
    }

    public static String base64Decode(String secureTxt) {
        byte[] bytes;
        try {
            bytes = new BASE64Decoder().decodeBuffer(secureTxt);
            return new String(bytes, "UTF-8");
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }
```

前端请求方式：

```
function doAutoSig() {
        let _ruleName = btoa('rule1');
        let _sealUser = btoa('sysuser');
        let _password = btoa('666666666');
        let _signKey = btoa('WQS202189898776');
        let _signSecret = btoa('999999999999999999');
        let _fileKey = btoa('1234abc');
        let _provideSigFile = btoa('http://192.168.1.30:8080/doSeal/getHandledFile/h1.pdf"');

        $.ajax( {
            url: _wqSigServer + '/autoSeal/doSeal',
            data: JSON.stringify({
                ruleName : _ruleName,
                sealUser : _sealUser,
                password : _password,
                signKey : _signKey,
                signSecret : _signSecret,
                fileKey : _fileKey,
                provideSigFile : _provideSigFile
            }),
            type:'post',
            dataType:'json',
            contentType: "application/json;charset=UTF-8",
            success:function(result){
                if(result.code == 0){
                    window.location.href = 'http://192.168.1.30:9999/no/getSealFile' +
result.data;
                }else{
                    alert(result.msg);
                }
            },
        });
    }
```

4.7. 返回参数说明（出参）

参数名	说明
code	返回的状态码，0 是成功，非 0 是错误码
msg	返回的说明信息

参数名	说明
data	签章文件 FileKey
timestamp	返回结果的时间戳

4.8. 返回参数示例

{"code":"0",

"msg":"关键字自动盖章完成！",

"timestamp":1575461457169,

"data":"fe52a861b1eb4739bbb50cd1c73d72cd.pdf"}

{"code":"107",

"msg":"签章规则（6666666666666）不存在！",

"timestamp":1575460560586,

"data":null}

4.9. 签章成功文件下载

接口地址：http://WeiQianSealURL:9999/no/getSealFile/fileKey

请求方式：GET

返回：签章完成文件

注意：使用 fileKey 下载签章完成文件时，当下载完成后，此 fileKey 将失效。

后台请求下载示例：

```
String downloadUrl = "http://192.168.1.30:9999/no/"
                    + "getSealFile/fe52a861b1eb4739bbb50cd1c73d72cd.pdf";
            String desFile = "D:/temp/seal/对账单_20200711.pdf";
            int byteRead = 0;
        try {
            URL url = new URL(downloadUrl);
```

```
            URLConnection connection    = url.openConnection();
                InputStream input = connection.getInputStream();
                FileOutputStream output = new FileOutputStream(desFile);
                byte[] buffer = new byte[1024];
                while((byteRead = input.read(buffer)) != -1){
                output.write(buffer, 0, byteRead);
                }
                output.flush();
                output.close();
        } catch (MalformedURLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
```

前端下载文件并显示：

```
success:function(result){
                        if(result.code == 0){
                            window.location.href = 'http://192.168.1.30:9999/no/getSealFile' +
result.data;
                        }else{
                            alert(result.msg);
                        }
                    },
```

注意：fileKey 只支持一次下载，下载完成后 fileKey 将失效！

5. Ukey 自动签章接口

5.1. Ukey 自动签章实现效果原理

Ukey 签章是从 Ukey 中拿去印章盖章的，有着更高的安全性。

5.2. Ukey 自动签章接口准备

- 客户端电脑安装 Ukey 驱动——WeiQian_UKey_Driver_4.1.19.1216.exe
- 引入 wqToken.js 和 mToken.js

5.3. UKey 自动签章接口概述

- 准备入参参数，因 ukPin 只用作前端校验 Ukey，所以不需要 Base64 编码。
- 调用 WQ_UkeySignature 方法进行 UKey 校验，只需传入 ukPin。
- UKey 校验成功后，会自动调用准备的入参参数提交签章请求。
- 通过 responseCode、responseData、responseMsg 可获得签章返回结果。

5.4. 请求参数说明（入参）

与 2.3 自动签章接口相比，Ukey 自动签章多了一个必须参数——ukPin（此参数无需 Base64 编码）。

provideSigFile 格式——http://host:port/*/fileKey（例如：http://192.168.1.29:8080/doSeal/provideSigFile/fe52a861b1eb4739bbb50cd1c73d72cd.pdf）。

fileKey 是 fe52a861b1eb4739bbb50cd1c73d72cd.pdf，注意不能包含中文字符。

签章完成后，签章服务器的返回结果，也同样包含此 fileKey，通过它调用下载接口，就可以下载签章完成的文件。

参数名	说明	必选
signKey	接口签名 key	Yes
signSecret	接口签名秘钥	Yes
ruleName	规则名称	Yes

provideSigFile	需签章的文件下载地址	Yes
sealUser	规则用章授权用户	Yes
password	规则用章授权用户密码	Yes
ukPin	Ukey PIN 码	Yes
fileKey	签章文件 key 或对应的任务 ID	No

5.5. UKey 自动签章请求示例

- 引入 Ukey 自动签章 JS

```
<title>微签电子签章服务器 API 演示平台</title>
    <link rel="icon" href="images/FY.ico" type="images/x-ico"/>
    <link rel="stylesheet" href="css/main.css">
    <script type="text/javascript" src="js/jquery-1.8.3.min.js"></script>
    <script type="text/javascript" src="js/base64.js"></script>
    <script type="text/javascript" src="js/mToken.js"></script>
    <script type="text/javascript" src="js/wqToken.js"></script>
    <script type="text/javascript">
```

- Ukey 自动签章方法对参数赋值，注意 ukPin 无需 Base64 编码，调用 WQ_UkeySignature 方法进行 Ukey 自动签章时，传入 ukPin 即可。

```
function doUKAutoSig() {
            //参数赋值
            let _ruleName = $("#ruleName").val();
            let _sealUser = $("#sealUser").val();
            let _signKey = $("#signKey").val();
            let _password = $("#password").val();
            let _signSecret = $("#signSecret").val();
            let _provideSigFile = $("#provideSigFile").val();
            wqSigServer = _wqSigServer;
            ruleName = btoa(unescape(encodeURIComponent(_ruleName)));
            sealUser = btoa(_sealUser);
            password = btoa(_password);
            signKey = btoa(_signKey);
            signSecret = btoa(_signSecret);
            provideSigFile = btoa((_provideSigFile));
            let ukPin = $("#certPw").val();
```

```
            //调用 WQ_UkeySignature 进行 UKey 签章，只需传入 UKey PIN 码
            WQ_UkeySignature(ukPin);
        }
```

- 通过 responseCode、responseData、responseMsg 可获得签章返回结果。

```
function getSealResult() {
            if(responseCode == 0){
                //UKey 自动签章成功
                let sigFilePrefix = wqSigServer + '/no/getSealFile/';
                window.location.href = 'showSigFile.html?sigFilePrefix=' + sigFilePrefix + '&sigFile=' + responseData;
            }else{
                //UKey 自动签章失败
                alert(responseMsg);
            }
        }
```

5.6. 返回参数说明（出参）

参数名	说明
responseCode	返回的状态码，0 是成功，非 0 是错误码
responseMsg	返回的说明信息
responseData	签章文件 FileKey

5.7. 签章成功文件下载

接口地址与自动签章接口相同，仍为：http://WeiQianSealURL:9999/no/getSealFile/fileKey。具体下载示例，请参考 3.8 章节。

6. 自动签章规则接口

6.1. 自动签章规则接口概述

通过调用接口，程序将会跳转到签章服务器的规则编辑页面，进行签章规则新建和编辑。

6.2. 新建签章规则接口

6.2.1. 新建签章规则接口说明

- 新建规则时必须提供规则模板文件的下载地址。
- 首先调用规则模板预处理接口，进行模板文件预处理。
- 模板文件预处理成功后，再调用手动签章接口，传入 pageType 为 2。

6.2.2. 模板文件预处理接口

接口地址：http://WeiQianSealURL:9999/manualSig/ruleFilePreprocess

请求方式：POST

参数格式：JSON

返回格式：JSON

请求参数说明（入参）

参数名	说明	必选
signKey	接口签名 key	Yes
signSecret	接口签名秘钥	Yes
provideSigFile	规则模板文件下载地址	Yes
sealUser	规则创建用户	Yes
password	用户密码	Yes
fileKey	签章文件 key 或对应的任务 ID	No

- 所有参数需为 Base64 编码，中文编码为 UTF-8，签章平台编码解码类为 sun.misc.BASE64Decoder 和 sun.misc.BASE64Encoder。
- signKey 和 signSecret 由我司商务部提供。
- 用户系统提供下载的签章文件，要指定签章文件类型，以下三种指定方式必须具备其一。

  指定方式一，使用 fileKey 参数，并且参数中指定文件类型，如：FWBPDtask000000001.pdf。

  指定方式二，在提供的下载地址中指定。如：

  http://192.168.1.31:8080/download?id=FWBPDtask000000001.pdf

指定方式三，下载接口代码中指定，使用 response.addHeader("Content-Disposition", "attachment;filename=FWBPDtask000000001.pdf")方式。

- sealUser 必须是具有创建规则权限用户。

6.2.3. 新建签章规则接口

接口地址：http://WeiQianSealURL:9999/manualSig/manualSigPage

请求方式：GET

返回格式：JSON

请求参数说明（入参）

signKey、signSecret、pageType、callbackUrl 要求 Base64 编码。

sigFile 为预处理接口成功完成，返回的 data，不需要 Base64 编码。

参数名	说明	必选
signKey	接口签名 key	Yes
signSecret	接口签名秘钥	Yes
pageType	新建规则值为 2	Yes
sigFile	预处理接口成功返回 data	Yes
callbackUrl	规则页面返回按钮的返回地址	Yes

6.2.4. 新建签章规则示例

```
function makeAutoSealRule() {
        let _sealUser = btoa($("#sealUser").val());
        let _signKey = btoa($("#signKey").val());
        let _password = btoa($("#password").val());
        let _signSecret = btoa($("#signSecret").val());
        let _provideSigFile = btoa(($("#provideSigFile").val()));
        let _callbackUrl = btoa(($("#callbackUrl").val()));
```

```
        let _pageType = btoa('2');
        $.ajax( {
            url: _wqSigServer + '/manualSig/ruleFilePreprocess',
            data: JSON.stringify({
                password : _password,
                sealUser : _sealUser,
                signKey : _signKey,
                signSecret : _signSecret,
                provideSigFile : _provideSigFile
            }),
            type:'post',
            dataType:'json',
            contentType: "application/json;charset=UTF-8",
            success:function(result) {
                if(result.code == 0){
                    let sigUrl = _wqSigServer +'/manualSig/manualSigPage?sigFile=' + result.data + '&signKey=' + _signKey + '&signSecret=' + _signSecret + '&callbackUrl=' + _callbackUrl + '&pageType=' + _pageType;
                    getManualSigPage(sigUrl);
                }else{
                    alert(result.msg);
                }
            },
        });
    }
```

6.2.5. 返回参数说明（出参）

参数名	说明
code	返回的状态码，0 是成功，非 0 是错误码
msg	返回的说明信息
data	签章文件 FileKey
timestamp	返回结果的时间戳

6.3. 编辑签章规则接口

6.3.1. 编辑签章规则接口说明

- 提供具有编辑签章规则权限用户和规则名称。
- 调用规则编辑接口，就会跳转到签章服务器编辑签章规则页面。

6.3.2. 规则编辑接口描述

接口地址：http://WeiQianSealURL:9999/manualSig/ruleUpdatePage

请求方式：GET

返回格式：JSON

请求参数说明（入参）

参数名	说明	必选
signKey	接口签名 key	Yes
signSecret	接口签名秘钥	Yes
ruleName	规则名称	Yes
callbackUrl	规则页面返回按钮的返回地址	Yes
sealUser	编辑规则用户	Yes
password	用户密码	Yes

- 所有参数需为 Base64 编码，中文编码为 UTF-8，签章平台编码解码类为 sun.misc.BASE64Decoder 和 sun.misc.BASE64Encoder。
- signKey 和 signSecret 由我司商务部提供。
- sealUser 必须是具有创建规则权限用户。

6.3.3. 编辑签章规则示例

```
function editAutoSealRule() {
```

```
let _signKey = btoa($("#signKey").val());
let _signSecret = btoa($("#signSecret").val());
let _ruleName = btoa(unescape(encodeURIComponent($("#ruleName").val())));
let _sealUser = btoa($("#sealUser").val());
let _password = btoa($("#password").val());
$.ajax( {
    url: _wqSigServer +'/manualSig/preprocessRule',
    data: JSON.stringify({
        signKey : _signKey,
        signSecret : _signSecret,
        ruleName : _ruleName,
        sealUser : _sealUser,
        password : _password
    }),
    type:'post',
    dataType:'json',
    contentType: "application/json;charset=UTF-8",
    success:function(result){
        if(result.code == 0){
            let _callbackUrl = btoa(($("#callbackUrl").val()));
            window.location.href = _wqSigServer +'/manualSig/ruleUpdatePage?signKey='
                + _signKey + '&signSecret=' + _signSecret + '&callbackUrl=' + _callbackUrl
                + '&ruleFile=' + result.data;
        }else{
            alert(result.msg);
        }
    }
});
}
```

7. 签章分页查询接口

7.1. 签章分页查询接口概述

通过调用接口，可以分页查询指定用户个人授权签章信息或绑定的所有用户组授权的签章信息。

7.2. 签章分页接口描述

接口地址：http://WeiQianSealURL:9999/no/getPagedSealList

请求方式：POST
参数格式：JSON
返回格式：JSON

7.3. 请求参数说明（入参）

参数名	说明	必选
signKey	接口签名 key	Yes
signSecret	接口签名秘钥	Yes
sealUser	签章授权用户	Yes
password	授权用户密码	Yes
sealType	签章授权类型	No
sealName	签章名称模糊搜索	No
pageNum	分页查询页码	No
pageSeals	分页查询每页签章数	No

- 所有参数需为 Base64 编码，中文编码为 UTF-8，签章平台编码解码类为 sun.misc.BASE64Decoder 和 sun.misc.BASE64Encoder。
- signKey 和 signSecret 由我司商务部提供。
- 非必须参数 sealType，默认值为 0（查询个人授权章），值为 1（查询用户组签章）。注意：用户组授权签章不能修改密码和启用禁用密码，需管理员登录后台统一设置整个用户组签章密码。
- 非必须参数 pageNum 和 pageSeals 默认值分别为 1 和 10。

7.4. 签章分页查询前端示例

```
function getPagedSealList() {
        let _sealUser = btoa($("#sealUser").val());
        let _password = btoa($("#password").val());
        let _pageNum = btoa($("#pageNum").val());
        let _pageSeals = btoa($("#pageSeals").val());
        let _signKey = btoa($("#signKey").val());
        let _signSecret = btoa($("#signSecret").val());
        let _sealName = btoa(unescape(encodeURIComponent($("#sealName").val())));
        let _sealType = btoa(($("#sealType").val()));
        $.ajax( {
```

```
            url: _wqSigServer + '/no/getPagedSealList',
            data: JSON.stringify({
                sealUser : _sealUser,
                password : _password,
                signKey : _signKey,
                signSecret : _signSecret,
                pageNum : _pageNum,
                pageSeals : _pageSeals,
                sealType : _sealType,
                sealName : _sealName
            }),
            type:'post',
            dataType:'json',
            contentType: "application/json;charset=UTF-8",
            success:function(result){
                if(result.code == 0){
                    let data = result.data;
                    console.log(data);
                    if(data.length == 0){
                        alert('用户没有任务签章授权！')
                    }else{
                        sessionStorage.setItem('pageSeals',JSON.stringify(data));
                        window.location.href = 'showSealTable.html';
                    }
                }else{
                    alert(result.msg);
                }
            },
        });
    }
```

7.5. 返回参数说明（出参）

参数名	说明
code	返回的状态码，0 是成功，非 0 是错误码
msg	返回的说明信息
data	签章分页数据
timestamp	返回结果的时间戳

- 查询成功返回的分页数据 data 为数组

```
▼Array(10) i
  ▶0: {sort: 'createTime', ascFlag: false, enabledFlag: null, signKey: null, signSecret: null, …}
  ▶1: {sort: 'createTime', ascFlag: false, enabledFlag: null, signKey: null, signSecret: null, …}
  ▶2: {sort: 'createTime', ascFlag: false, enabledFlag: null, signKey: null, signSecret: null, …}
  ▶3: {sort: 'createTime', ascFlag: false, enabledFlag: null, signKey: null, signSecret: null, …}
  ▶4: {sort: 'createTime', ascFlag: false, enabledFlag: null, signKey: null, signSecret: null, …}
  ▶5: {sort: 'createTime', ascFlag: false, enabledFlag: null, signKey: null, signSecret: null, …}
  ▶6: {sort: 'createTime', ascFlag: false, enabledFlag: null, signKey: null, signSecret: null, …}
  ▶7: {sort: 'createTime', ascFlag: false, enabledFlag: null, signKey: null, signSecret: null, …}
  ▶8: {sort: 'createTime', ascFlag: false, enabledFlag: null, signKey: null, signSecret: null, …}
  ▶9: {sort: 'createTime', ascFlag: false, enabledFlag: null, signKey: null, signSecret: null, …}
    length: 10
  ▶[[Prototype]]: Array(0)
```


- 每条数据中签章信息有 isealId、sealName、enabledPw（启用密码 0，禁用密码 1）、keepPw（不记住密码 0，记住密码 1）、sealFile（Base64 码签章图片）。

签章ID	签章名称	启用密码	记住密码	签章印模
13	s12	0	0	
12	s11	0	0	
11	s10	0	0	

8. 签章密码功能接口

8.1. 签章密码功能接口概述

调用此接口可以修改个人授权签章密码功能，但是对于用户组授权签章无效。

8.2. 签章密码功能接口描述

接口地址：http://WeiQianSealURL:9999/no/editSealPassword
请求方式：POST
参数格式：JSON
返回格式：JSON

8.3. 请求参数说明（入参）

参数名	说明	必选
signKey	接口签名 key	Yes
signSecret	接口签名秘钥	Yes

sealUser	签章授权用户	Yes
sealId	签章 ID	Yes
password	签章密码（不是用户账号密码）	Yes
newPassword	签章新密码（不修改密码可不提供）	No
enabledPw	启用/禁用签章密码	No
keepPw	是否记住密码	No

- 所有参数需为 Base64 编码，中文编码为 UTF-8，签章平台编码解码类为 sun.misc.BASE64Decoder 和 sun.misc.BASE64Encoder。
- signKey 和 signSecret 由我司商务部提供。
- 非必须参数 newPassword，不需要修改签章密码，只想启用或禁用密码功能，或只想记住密码时，就不用提供。如果提供 newPassword 就认为用户是想要修改签章密码。
- 非必须参数 enabledPw（禁用密码 0，启用密码 1）默认禁用密码。
- 非必须参数 keepPw（不记住密码 0，记住密码 1）默认不记住密码。当用户修改值为记住密码时，此时就算启用了密码功能，在签章页面也不会要求输入密码。

8.4. 签章密码功能接口前端示例

```
function editSealPassword() {
        let _sealId = btoa($("#sealId").val());
        let _sealUser = btoa($("#sealUser").val());
        let _password = btoa($("#password").val());
        let _signKey = btoa($("#signKey").val());
        let _signSecret = btoa($("#signSecret").val());
        let _newPassword = btoa(($("#newPassword").val()));
        let _enabledPw = btoa(($("#enabledPw").val()));
        let _keepPw = btoa(($("#keepPw").val()));
        $.ajax( {
            url: _wqSigServer + '/no/editSealPassword',
            data: JSON.stringify({
                sealId : _sealId,
                sealUser : _sealUser,
                password : _password,
                signKey : _signKey,
                signSecret : _signSecret,
                newPassword : _newPassword,
                enabledPw : _enabledPw,
```

```
                keepPw : _keepPw
            }),
            type:'post',
            dataType:'json',
            contentType: "application/json;charset=UTF-8",
            success:function(result){
                alert(result.msg);
            },
        });
    }
```

8.5. 返回参数说明（出参）

参数名	说明
code	返回的状态码，0 是成功，非 0 是错误码
msg	返回的说明信息
timestamp	返回结果的时间戳

9. 错误码参照

错误码	说明
101	请求的必须参数有空值
104	请求用户不存在或被禁用
105	请求用户密码错误
106	SignSecret 错误
107	签章规则不存在或被禁用
108	签章用户无权调用规则用章
109	签章文件是不支持类型
110	签章文件 url 异常
111	提交签章的文件下载失败
112	规则指定用章不存在
113	用户证书与接口类型不符

错误码	说明
114	签章文件预处理失败，可能是加密文件
231	关键字盖章未发现关键字
232	关键字盖章异常
233	关键字盖章失败
251	骑缝章文件页码异常
252	骑缝章签章异常
261	用户无权调用规则时间戳
262	时间戳盖章异常
1001	未导入证书或规则证书异常
1002	初始化数字证书异常
1101	指定坐标单页数字证书盖章异常
1102	签章时间服务中心连接失败！
1201	指定坐标多页数字证书盖章异常
1301	关键字盖章未发现关键字
1302	首次/全部关键字盖章异常
2001	请求下载的签章文件不存在
2002	html 签章文件转换失败

错误码	说明
2003	下载或转换的签章文件读取异常
2004	签章文件没有 license 权限
3001	Ukey 签章接口调用错误
3002	Ukey 自动签章不支持多规则同时调用
3003	Ukey 签章文件签名数据获取失败
3004	Ukey 自动签章失败
3005	当前 Ukey 与规则指定 Ukey 不符
3006	Ukey 盖章未发现关键字
3007	Ukey 关键字盖章异常
4001	手动签章文件是不支持类型
4002	手动签章文件 url 异常
4003	手动签章没有 license 权限
4004	手动签章文件下载失败
4005	复制到临时签章目录失败
4006	复制到前端显示目录失败
4007	手动盖章文件初始化失败
4008	手动签章验签参数有空值
4009	手动签章用户不存在
4010	手动签章文件 Url 为空
4011	手动签章用户密码错误
4012	手动签章 signSecret 错误

错误码	说明
4013	手动签章 License 获取失败
4014	手动签章用户已被禁用
5001	请求用户没有创建规则权限
5002	规则预览签章失败
5003	签章预览关键字盖章异常
5004	签章预览未发现关键字
6001	用户组签章，不能编辑密码
6002	签章密码错误
6003	数据库更新签章密码失败
6004	用户还没有绑定签章用户组

10. 技术支持

如果您在使用中遇到问题，请仔细阅读此用户手册。如果仍不能解决问题，请与上海复园电子科技有限公司技术部取得联系，我们将为您提供及时、周到的服务。

上海复园电子科技有限公司

地址：上海市国定路 335 号 2 号楼 20 层
电话：021-65654240 转技术支持
E-mail：support@forwave.com
网址：www.forwave.com