



---

# 微签电子签章平台

## Web API 开发接口说明

---

***FORWAVE***

上海复园电子科技有限公司

## 目录

1 简介.....	1
2 自动签章接口.....	1
2.1 实现原理.....	1
2.2 自动签章接口描述.....	1
2.3 请求参数说明（入参）.....	2
2.4 自动签章请求示例.....	3
2.5 返回参数说明（出参）.....	5
2.6 返回参数示例.....	5
2.7 签章成功文件下载.....	6
3 Ukey 自动签章接口.....	7
3.1 UKEY 自动签章接口准备.....	7
3.2 请求参数说明（入参）.....	7
3.3 UKEY 自动签章请求示例.....	8
3.4 返回参数说明（出参）.....	9
3.5 签章成功文件下载.....	9
4 手动签章接口.....	10
4.1 实现原理.....	10
4.2 手动签章接口描述.....	10
4.3 请求参数说明（入参）.....	10
4.4 手动签章请求示例.....	12
5 错误码参照.....	13
6 技术支持.....	16

## 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。

WeiQian Web API 接口，提供手动签章和自动签章两种接口方式，可以很方便地与各种 Web 业务系统集成，与协同办公平台、电子政务平台、OA 办公自动化系统、ERP、BPM 等系统集成，打通无纸化办公的最后一个环节。

## 2 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。

### 2.1 实现原理

第一步：需对接的系统向签章平台发送自动签章请求。

第二步：请求参数中包含签章平台已建立的自动签章规则名称。

第三步：签章平台响应请求，并调用指定的签章规则进行自动签章。

第四步：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

### 2.2 自动签章接口描述

接口地址：<http://WeiQianSealURL:9999/autoSeal/doSeal>

返回格式：JSON

请求方式：POST

## 2.3 请求参数说明（入参）

必选参数值为 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。

参数名	类型	说明	必选
signKey	String	接口签名 key	Yes
signSecret	String	接口签名秘钥	Yes
rulerName	String	规则名称	Yes
sealFileUrl	String	需签章的文件下载地址	Yes
sealUser	String	规则用章授权用户	Yes
password	String	规则用章授权用户密码	Yes

sealFileUrl 格式——http://host:port/\*/fileKey

如 http://192.168.1.29:8080/doSeal/provideSigFile/fe52a861b1eb4739bbb50cd1c73d72cd.pdf，file Key 就是 fe52a861b1eb4739bbb50cd1c73d72cd.pdf。

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。

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	类型	说明	必选
sealFileCharSetName	String	签章文件编码	No
marginLeft	float	签章文件转换 PDF 左边距	No
marginRight	float	签章文件转换 PDF 右边距	No
marginTop	float	签章文件转换 PDF 上边距	No

marginBottom	float	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 自动签章请求示例

后台请求方式:

```
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 signKey = Base64.base64Encode("ESFORWAVE1204");
String signSecret = Base64.base64Encode("9999999999999999");
String rulerName = Base64.base64Encode("t1");
String sealFileUrl = Base64.base64Encode("http://192.168.1.30:8080/doSeal/getHandledFile/h13.pdf");
params.put("sealUser", sealUser64);
params.put("password", password64);
params.put("signKey", signKey);
params.put("signSecret", signSecret);
params.put("rulerName", rulerName);
params.put("sealFileUrl", sealFileUrl);
autoSealReq(requestUrl,params);
```

```
//自动签章请求方法
private String autoSealReq(String requestUrl,Map<String,String> params) {
    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();
        String json = StringUtil.map2Json(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 StringUtil.is2String(inputStream);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    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 sealRequest() {
    var rulerName = btoa('t1');
    var sealFileUrl = btoa('http://192.168.1.29:8080/doSeal/getHandledFile/h13.pdf');
    var sealUser = btoa('sysuser');
    var password = btoa('66666666');
    var signKey = btoa('1234abc');
    var signSecret = btoa('99999999999999999999');
    $.ajax( {
        url: 'http://192.168.1.30:9999/autoSeal/doSeal',
        data: JSON.stringify({
            rulerName : rulerName,
            sealFileUrl : sealFileUrl,
            sealUser : sealUser,
            password : password,
            signKey : signKey,
            signSecret : signSecret
        }),
        type: 'post',
        dataType: 'json',
        contentType: "application/json;charset=UTF-8",
        success: function(result) {
            if(result.code == 0) {

```

## 2.5 返回参数说明（出参）

参数名	类型	说明
code	int	返回的状态码
msg	string	返回的说明信息
data	string	签章文件 FileKey
timestamp	Long	返回结果的时间戳

## 2.6 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（666666666666）不存在！",

"timestamp":1575460560586,

"data":null}
```

## 2.7 签章成功文件下载

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

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 httpurl = new URL(downloadUrl);
    URLConnection conn = httpurl.openConnection();
    InputStream inStream = conn.getInputStream();
    FileOutputStream fs = new FileOutputStream(desFile);
    byte[] buffer = new byte[1024];
    while ((byteread = inStream.read(buffer)) != -1) {
        fs.write(buffer, 0, byteread);
    }
    fs.flush();
    fs.close();
    return true;
} catch (MalformedURLException e) {
    e.printStackTrace();
}
```

前端下载文件并显示：



```
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{
    console.log(result.msg);
  }
},
```

注意：fileKey 只支持一次下载，下载完成后 fileKey 将失效！

## 3 Ukey 自动签章接口

### 3.1 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

### 3.2 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）。

参数名	类型	说明	必选
signKey	String	接口签名 key	Yes
signSecret	String	接口签名密钥	Yes
rulerName	String	规则名称	Yes
sealFileUrl	String	需签章的文件下载地址	Yes
sealUser	String	规则用章授权用户	Yes
password	String	规则用章授权用户密码	Yes
ukPin	String	Ukey PIN 码	Yes

### 3.3 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。

```
//UKey自动签章
function doUKAutoSig() {
    //参数赋值
    let _rulerName = $("#rulerName").val();
    let _sealUser = $("#sealUser").val();
    let _signKey = $("#signKey").val();
    let _password = $("#password").val();
    let _signSecret = $("#signSecret").val();
    let _provideSigFile = $("#provideSigFile").val();
    rulerName = btoa(unescape(encodeURIComponent(_rulerName)));
    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);
}
```

- Ukey 自动签章页面必须提供 `getSealResult()` 方法，供 Ukey 自动签章 JS 包调用，传入 Ukey 自动签章结果。

```
//获取UKey签章返回结果回调方法
//responseCode 结果状态码, 0成功, 非0失败
//responseData 签章成功后返回的FileKey
//responseMsg 签章结果信息
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
        console.log(responseMsg);
    }
}
```

### 3.4 返回参数说明（出参）

参数名	类型	说明
responseCode	int	返回的状态码
responseMsg	string	返回的说明信息
responseData	string	签章文件 FileKey

注：Ukey 自动签章返回结果，必须在页面提供 `getSealResult` 回调方法，供 Ukey 自动签章 JS 包调用，传入 Ukey 自动签章结果。

### 3.5 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 2.7 章节。

## 4 手动签章接口

### 4.1 实现原理

- 客户端访问签章服务器手动签章文件预处理接口。
- 根据预处理接口返回结果，再访问手动签章接口，将跳转到签章服务器手动签章页面。
- 在手动签章页面可进行手动签章操作。
- 手动签章页面点击完成按钮，签章服务器将回调签章完成接口，向客户端回传签章完成文件 Filekey。

### 4.2 手动签章接口描述

签章文件预处理接口：<http://WeiQianSealURL:9999/manualSig/preprocess>

返回格式：JSON

请求方式：POST

手动签章页面接口：[http://WeiQianSealURL:9999/manualSig/manualSigPage?sigFile=FileKey&signKey=\\_signKey&signSecret=\\_signSecret](http://WeiQianSealURL:9999/manualSig/manualSigPage?sigFile=FileKey&signKey=_signKey&signSecret=_signSecret)

请求方式：GET

### 4.3 请求参数说明（入参）

- 签章文件预处理接口参数说明。

所有参数进行 Base64 编码。

getSigFile 参数必须提供一个回调接口，手动签章页面点击完成按钮，签章服务器将回调此接口，向客户端回传签章完成文件 Filekey。

参数名	类型	说明	必选
signKey	String	接口签名 key	Yes
signSecret	String	接口签名密钥	Yes
sealFileUrl	String	需签章的文件下载地址	Yes
getSigFile	String	手动签章完成后的回调接口	Yes

sealUser	String	手动签章请求用户	Yes
password	String	手动签章用户密码	Yes

- 手动签章页面请求接口参数说明。

signKey 和 signSecret 需 Base64 编码。

sigFile 无需 Base64 编码。

参数名	类型	说明	必选
sigFile	String	手动签章预处理完成文件 FileKey	Yes
signKey	String	接口签名 key	Yes
signSecret	String	接口签名密钥	Yes

## 4.4 手动签章请求示例

- 签章文件预处理接口

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

- 手动签章页面请求接口

```
success: function(result){
    if(result.code == 0){
        let sigUrl = _wqSigServer + '/manualSig/manualSigPage?sigFile=' +
            result.sealedFile + '&signKey=' + _signKey + '&signSecret=' + _signSecret;
        getManualSigPage(sigUrl);
    }else{
        console.log(result.msg);
    }
},
```

- 回调接口（<http://192.168.1.29:8080/doSeal/getSigFile>）

```

/**
 * 获取签章完成文件
 *
 * @param
 * @return
 */
@RequestMapping(value = "/getSigFile")
public ModelAndView getSigFile(@RequestParam("sigFile") String sigFile) {
    String sigFilePrefix = wqSigServer + "/no/getSealFile/";
    return new ModelAndView( viewName: "redirect:/showSigFile.html?sigFilePrefix="
        + sigFilePrefix + "&sigFile=" + sigFile);
}

```

## 5 错误码参照

错误码	说明
101	验签参数有空值
102	请求参数中，签章文件 Url 为空
103	请求参数中，签章规则为空
104	签章用户不存在
105	签章用户密码错误
106	SignSecret 错误
107	签章规则不存在
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	自动签章文件 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
114	签章文件预处理失败，可能是加密文件
115	自动签章用户已被禁用

错误码	说明
116	自动签章规则已被禁用
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
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
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



错误码	说明
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
4013	手动签章 License 获取失败
4014	手动签章用户已被禁用

## 6 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

传真：021-55666857

E-mail: [support@forwave.com](mailto:support@forwave.com)

网址: [www.forwave.com](http://www.forwave.com)