下载js

```
npm i parser_x_x.js
或者
<script type="text/javascript" src="https://cdn.jsdelivr.net/npm/parserhelper.js"></
<script type="text/javascript" src="https://cdn.jsdelivr.net/npm/parser_helper_x.js"
<script type="text/javascript" src="https://cdn.jsdelivr.net/npm/parser_x_x.js"></sc
```

授权码

参数	值
appKey	vNY5W9ghj5JguuuD
appSecret	5zSssFRm3r24nvTjaytvEYLvhFubnYEV

校验码使用方式

```
后续接口secret生成方式： `${appKey}${13位的时间戳}${随机数}`
后续接口digest生成方式： `md5(${appKey}${13位的时间戳}${随机数}${appSecret})`
算这一组的时候，时间戳和随机数是一样的，13位时间戳用于控制digest的有效期，
建议secret和digest服务器端生成，避免appSecret存放在js端有泄露风险
```

解析OFD文件

参数	说明	是否必填
ofdFile	OFD文件。支持File、ArrayBuffer及url	是
secret	看使用方式	是
digest	看使用方式	是
haaders	如果url需要headers传入此参数，如{key1:value1, key2:value2}	否
waterText	水印内容	否

```
parseOfdDocument({
        ofd: ofdFile,
        secret: '',
        digest: '',
        success() {

        },
        fail(error){
            console.log(error)
        }
    })
```

获取OFD文档页数

此方法需要在parseOfdDocument success回调后使用

参数	说明	是否必填
documentIndex	ofd文档中document的索引，默认从0开始	是

```
getOFDPageCount(documentIndex)
```

获取OFD文档对应页的页宽

此方法需要在parseOfdDocument success回调后使用

参数	说明	是否必填
documentIndex	ofd文档中document的索引，默认从0开始	是
pageIndex	ofd文档中页码，默认从0开始	是

```
pageWidth(documentIndex, pageIndex)
```

获取OFD文档对应页的页高

此方法需要在parseOfdDocument success回调后使用

参数	说明	是否必填
documentIndex	ofd文档中document的索引，默认从0开始	是
pageIndex	ofd文档中页码，默认从0开始	是

```
pageHeight(documentIndex, pageIndex)
```

获取OFD文档对应页的实际size以及渲染的size

此方法需要在parseOfdDocument success回调后使用

参数	说明	是否必填
documentIndex	ofd文档中document的索引，默认从0开始	是
pageIndex	ofd文档中页码，默认从0开始	是
width	预期渲染的宽度，像素值 ，如800	否

```
pageSize(documentIndex, pageIndex, width)
```

一次性渲染OFD对应文档的所有页，适合页数少

此方法需要在parseOfdDocument success回调后使用

参数	说明	是否必填
documentIndex	ofd文档中document的索引，默认从0开始	是
width	预期渲染的宽度，像素值 ，如800	否

```
renderOfd(documentIndex, width).then(divs=>{
    // do something
})
```

渲染OFD对应文档的对应页

此方法需要在parseOfdDocument success回调后使用

参数	说明	是否必填
documentIndex	ofd文档中document的索引，默认从0开始	是
pageIndex	ofd文档中页码，默认从0开始	是
width	预期渲染的宽度，像素值 ，如800	否

```
renderOfdByIndex(documentIndex, pageIndex, width).then(div => {
    // do something
}})
```

打开一个基础的内置ofdview

此方法直接使用即可

html需要添加一个id为xxx的div容器，其中overflow必须设置为auto，width和height必须设置

```
<div id="OfdView" style="overflow:auto;width:800px;height:1000px"></div>
```

参数	说明	是否必填
ofd	OFD文件。支持File、ArrayBuffer及url	是
secret	看使用方式	是
digest	看使用方式	是
container	document.getElementById('xxx')	是
loadingContainer	页面加载元素，自定义	否
width	期望渲染的每页宽度	否
waterText	水印文字	否
signaturesCallback	签章列表，返回signatures列表，后续介绍	否
signatureClickCallback	用于点击签章响应回调，返回该签章对应的页数和id	否
signatureClickCallback	用于点击签章响应回调，返回该签章对应的页数和id	否
parserOFDSuccess	解析ofd成功预览页面前回调	否
parserOFDFail	解析ofd失败回调	否

```
openOFDBaseViewer({
        ofd: xxx,
        secret: 'xxxx',
        digest: 'xxxx',
        container: document.getElementById('OfdView'),
        loadingContainer: xxx,
        parserOFDSuccess() {
            console.log('解析成功')
        },
        parserOFDFail() {
            console.log('解析失败')
        },
        signaturesCallback(signatures) {
          //所有签章列表的信息
          console.log(signatures)
        },
        signatureClickCallback (evt) {
          //verifySignature方法后续介绍
          verifySignature({
            pageRef: evt.pageRef,
            signatureId: evt.signatureId,
            signatureCallback(data) {
              //验章的数据返回
              console.log(data)
            },
            hashFileCallback(ret) {
              //摘要验证的结果
              console.log(ret)
            }
          })
        }
      })
```

签章列表的信息介绍，返回时电子签章数组

```
[
{
    // object内容主体是ofd中signature.xml内容为主
    "ID": "2",
    "BaseLoc": "Doc_0/Signs/Sign_1/Signature.xml",
    "Signature": {
        "SignedValueLoc": "/Doc_0/Signs/Sign_1/SignValue.dat",
        "PicValue": "xxx", //印章图片的base64
        "PicType": "png", //印章图片的类型
        "width": 40, //印章图片的宽
        "height": 40, //印章图片的高
        "SignatureValid": true, //验签结果
```

```
        "HashValid": true, //待签名数据摘要比对结果
        "signatureHex": "", //签名值16进制
        "HashHex": "", //摘要16进制
        "SignCert": { //签章证书
            "Version": 2,
            "subject": "13010000008088",
            "issuer": "ShanXi Digital Certificate Authority",
            "SerialNumber": "57660EDEF5A2B846",
            "NotBefore": "2020-12-03 18:47:03",
            "NotAfter": "2030-12-01 18:47:03",
            "PublicKeyHex": "0470d46b354f7696a9e94f329d0e124830b08899d76538de21d0927
            "PublicKeyType": "ECC(256 Bits)",
            "SignAlgType": "SM3WithSM2Encryption",
            "SignHex": "3045022077a3596e7183a230313e5a22191fba3d508f1a13badd0e8fe827
        },
        "SealCert": { //制章证书
            "Version": 2,
            "subject": "xxxx",
            "issuer": "ShanXi Digital Certificate Authority",
            "SerialNumber": "3AD76211C9A8AA8C",
            "NotBefore": "2019-04-28 17:00:00",
            "NotAfter": "2029-04-25 17:00:00",
            "PublicKeyHex": "04d5922804795ec3fa682150c6ebaa180c0348f6a05abad48f6dd69
            "PublicKeyType": "ECC(256 Bits)",
            "SignAlgType": "SM3WithSM2Encryption",
            "SignHex": "304502203072e64d8b0300daa0ce082fa80127368900d0fb4e55a79902dc
        },
        "SignedInfo": {
            "Provider": { // 以下均为signatrue.xml里面的内容，不做介绍
                "Version": "2.0",
                "Company": "DianJu",
                "ProviderName": "DJ"
            },
            "Seal": { // 电子印章的属性
                "Property": {
                    "name": "xxxxx专用章",
                    "type": 1,
                    "createDate": "2020-12-03 18:52:42",
                    "validStart": "2020-12-03 18:47:03",
                    "validEnd": "2030-12-01 18:47:03"
                },
                "Header": {
                    "ID": "ES",
                    "version": 4,
                    "Vid": "dianju"
                },
                "esVersion": 4,
                "esID": "13010000008088"
```

```
        },
        // 以下均为signatrue.xml里面的内容，不做介绍
        "SignatureMethod": "1.2.156.10197.1.501",
        "SignatureDateTime": "20220518031019Z",
        "ReferencesCheckMethod": "1.2.156.10197.1.401",
        "References": [{
            "FileRef": "/Doc_0/Res_0/img2.jpg",
            "CheckValue": "aL+w/eeJ6w26FPkRFpAyskTDkgS6zZNyt99gHO/+Fl4="
        }, {
            "FileRef": "/Doc_0/Templates/Temp_0/Form.xml",
            "CheckValue": "NwZd9fxuUY6nPpT/FqSQ4FKOpcItpCXpEptuC5clmCA="
        }, {
            "FileRef": "/Doc_0/Templates/Temp_0/Content.xml",
            "CheckValue": "cS7aMSZvi9+MsAfhxE4ooOALeehZk7tSr1z65cweyok="
        }, {
            "FileRef": "/Doc_0/Forms.xml",
            "CheckValue": "5mpfL8C8JmOWM5taSFj0hamBurEuKPYSNqB3PUYT0sU="
        }, {
            "FileRef": "/Doc_0/Res_0/img6.jpg",
            "CheckValue": "HHZhpmcNLfjXa/72MFQZ4TeRcxj7+pBl2RYoAKIqdww="
        }, {
            "FileRef": "/Doc_0/DocumentRes_0.xml",
            "CheckValue": "gLBCNMd4/0tDUwzQ45RlDw4sO4rqjSyT5j2Skz+FP0E="
        }, {
            "FileRef": "/Doc_0/PublicRes_0.xml",
            "CheckValue": "GrRMHMCsnMrLotd2KnEOl3SyMDYh6/kU0bAUKrAFqI0="
        }, {
            "FileRef": "/Doc_0/Pages/Page_0/Content.xml",
            "CheckValue": "ow4g/snBkrW3/j2uE/hC1QggYJpKMf9AgXsN1224If4="
        }, {
            "FileRef": "/Doc_0/Tags/Tag_Custom.xml",
            "CheckValue": "OfQKTqLNnhFDW1ss6v+k/jM4b6sOFu0YIh33ZJphGLQ="
        }, {
            "FileRef": "/Doc_0/CustomTags.xml",
            "CheckValue": "RFvpXePnUq/MpZ4B6GP7XwrsVwFnzVi3S2OXLubWJ+4="
        }, {
            "FileRef": "/Doc_0/Res/27.gif",
            "CheckValue": "5bL7tjMuNLvHFMvZOv9kxaWAINePwTCYRhZ1CQPRh2M="
        }, {
            "FileRef": "/Doc_0/Annotations.xml",
            "CheckValue": "AE70HIIX7PHXQkIXg4KZU+yiyLQjsYh8DnenjXwgsYg="
        }, {
            "FileRef": "/Doc_0/Pages/Page_0/Annotation.xml",
            "CheckValue": "Z3OGtJk6TGy9NlS4GIskZs94IB2WFwQkSMJvroL34Zc="
        }],
        "StampAnnot": [{
            "ID": "3",
            "PageRef": 3,
```

```
                "Boundary": {
                    "x": 98.94,
                    "y": 118.77,
                    "w": 39.91,
                    "h": 39.92
                },
                "Clip": {
                    "x": 0,
                    "y": 0,
                    "w": 39.91,
                    "h": 39.92
                }
            }]
        }
    }
}
]
```

根据对应pageref和signatureId验章

参数	说明	是否必填
pageRef	签章关联的pageref	是
signatureId	签章id	是
signatureCallback	验章回调的数据，后续介绍	否
hashFileCallback	保护文件摘要回调结果，返回true或者false	否

```
verifySignature({
    pageRef: evt.pageRef,
    signatureId: evt.signatureId,
    signatureCallback(data) {
      //验章的数据返回
      console.log(data)
    },
    hashFileCallback(ret) {
      //摘要验证的结果
      console.log(ret)
    }
})
```

验章回调的数据

```
{
    "signer": "13010000008088", //签章人
    "provider": "DJ", //签章提供者
    "hashedValue": "", //摘要值16进制
    "signedValue": "", //签名值16进制
    "signMethod": "1.2.156.10197.1.501", //签名方法标识
    "signVersion": 4, //版本号
    "signatureVerify": true, //验签结果，true or false
    "sealID": "13010000008088", // 印章id
    "sealName": "xxx", // 印章名称
    "sealType": 1, // 印章名称
    "sealAuthTime": "2020-12-03 18:47:03 至 2030-12-01 18:47:03", // 印章有效时间
    "sealMakeTime": "2020-12-03 18:52:42", // 印章制作时间
    "sealVersion": 4 // 印章版本号
}
```

base_viewer的介绍

getViewerConfiguration方法尽量不动，这个是基于viewer.html的初始化
其中特别说明的是config中的部分参数

参数	说明	是否必填
secret	看使用方式	是
digest	看使用方式	是
loadingContainer	页面加载元素，自定义	否
url	支持http的ofd地址	否
waterText	水印文字	否
parserOFDSuccess	解析ofd成功预览页面前回调	否
parserOFDFail	解析ofd失败回调	否
onPageChanging	页码回调	否
onPageScale	缩放比例回调	否