# Digital Signature Wrapper API Docs:

A FastAPI web app that allows digital signature process automation using an external digital signature service. We will take as a reference the **Yousign** digital signature service, the one that will be first implemented, but the idea is to build an abstraction layer in order to adapt the app behaviour to manage different external resources.

**Header:**
Access to the Yousign API is performed via **Bearer Authentication** that involves security tokens called bearer tokens. So every request has to include Authorization header with your API key to be authenticated:

*Authorization: Bearer {api_key}*

The api_key is created once for all from Yousign web platform and will be stored as app config. Production and Development configurations will include a specific api_key in order to call the corresponding Yousign service:

| Environment | Yousign Api Base URL | Web App Url |
|---|---|---|
| Staging | https://staging-api.yousign.com | https://staging-app.yousign.com |
| Production | https://api.yousign.com | https://webapp.yousign.com |

The API accepts JSON content in requests and returns JSON content in all of its responses. This format is also forced by the Header Content-Type:

*accept: application/json*

## POST /procedure:
A procedure is the digital signature process of inviting one or more participants (members) to sign one or more documents (files).
It's important to emphasize that files to be signed must be in **PDF** format and that it is quite possible to link several files to the same signature procedure.

**Request Payload:**
```
{
        "vendor": "indomio",
        "members": [
                {
                "firstname": "First",
                "lastname": "Signer",
                "email": "first.signer@example.it",
                "phone": "+393921234567",
                },
                {
                "firstname": "Second",
                "lastname": "Signer",
                "email": "second.signer@example.it",
                "phone": "+393927654321",
                }
        ]
}
```

**Response status code 200:**

```
{
        "procedure_id": "cf7a6583-8722-45cd-a146-906e08bc36e2",
        "vendor": "indomio",
        "status": "active",
        "signers": [
                {
                "firstname": "First",
                "lastname": "Signer",
                "email": "first.signer@example.it",
                "phone": "+393921234567",
                "status": "pending",
                "iframe_src":
"https://staging-app.yousign.com/procedure/sign?members=/members/3af46b58-8c3a-4877-9878-1044fc8e0db3",
                },
                {
                "firstname": "Second",
                "lastname": "Signer",
                "email": "second.signer@example.it",
                "phone": "+393927654321",
                "status": "pending",
                "iframe_src":
"https://staging-app.yousign.com/procedure/sign?members=/members/3af46b58-8c3a-4877-9878-1044fc8e0000",
                }
        ]
}
```

**Request workflow:**
**Step 1 – Upload Pdf file to be signed to get the {file_id}**

**Note:** It's important to emphasize that files to be signed must be in **PDF format** and that it is quite possible to link several files to the same signature procedure. Also is highly recommended to specify the ".pdf" extension at the end of your file.

```
curl --request POST \
    --url {base_api_url}/files \
    --header 'Authorization: Bearer {api_key}' \
    --header 'accept: application/json'
    --data '{
        "name": "to_be_signed.pdf",
        "content": base64_encoded_pdf.decode('utf-8'),
        }'
```

**Step 2 – Create the procedure**

```
curl --request POST \
    --url {base_api_url}/procedures \
    --header 'Authorization: Bearer {api_key}' \
    --header 'accept: application/json'
    --data '{
```

```json
    "name": "Procedure name",
    "description": "Procedure description",
    "members": [
            {
        "firstname": "First",
        "lastname": "Signer",
        "email": "first.signer@example.it",
        "phone": "+393921234567",
        "operationLevel": "custom",
        "operationCustomModes": [
          "sms"
            ],
    "fileObjects": [
            {
            "file": {file_id},
            "page": 2,
            "position": "230,499,464,589",
            "mention": "Read and approved",
            "mention2": "Signed by First Signer"
            }
     ]
    },
            {
        "firstname": "Second",
        "lastname": "Signer",
        "email": "second.signer@example.it",
        "phone": "+393927654321",
        "operationLevel": "custom",
        "operationCustomModes": [
          "sms"
            ],
    "fileObjects": [
            {
            "file": {file_id},
            "page": 2,
            "position": "230,499,464,589",
            "mention": "Read and approved",
            "mention2": "Signed by Second Signer"
            }
     ]
    }
    ],
    "config": {
    "webhook": {
            "member.finished": [
                    {
                    "url": {callback_api_endpoint},
                    "method": "GET",
                    "headers": {
                            "X-Custom-Header": "Yousign Webhook - Test value"
                            }
                    }
```

```
                ]
        },
        "email": {
                "member.started": [
                {
                "subject": "Hey! You are invited to sign!",
                "message": "Hello <tag data-tag-type=\"string\" data-tag-
        name=\"recipient.firstname\"></tag> <tag data-tag-type=\"string\" data-tag-
        name=\"recipient.lastname\"></tag>, <br><br> You have ben invited to sign a document,
        please click on the following button to read it: <tag data-tag-type=\"button\" data-tag-
        name=\"url\" data-tag-title=\"Access to documents\">Access to documents</tag>",
        "to": [
        "@member"
        ]
        }
        ]
        }
    }
}'
```

**Webhooks and email notifications:**

For a better integration between Yousign and your application, Yousign provides webhook, a server-2-server notification which permits to inform your system that a procedure or any resource is updated. Events that can be related to webhooks and email notifications are:

| Name | Description |
|---|---|
| procedure.started | Fired when a procedure is created (POST /procedures) |
| procedure.finished | Fired when a procedure is finished (all members have signed) |
| procedure.refused | Fired when a procedure is refused (a member have refused) |
| procedure.expired | Fired when a procedure expired (The expiresAt date is reached) |
| member.started | Fired when a member can sign |
| member.finished | Fired when a member have signed |
| comment.created | Fired when someone comment a procedure |

**Step 3 - Parse response in order to return only the essential information**

**GET /procedure/{procedure_id}**

**Response status code 200:**
```
{
        "procedure_id": "cf7a6583-8722-45cd-a146-906e08bc36e2",
        "vendor": "indomio",
        "status": "active",
        "signers": [
                {
                "firstname": "First",
```

```
                "lastname": "Signer",
                "email": "first.signer@example.it",
                "phone": "+393921234567",
                "status": "pending",
                "iframe_src":
"https://staging-app.yousign.com/procedure/sign?members=/members/3af46b58-8c3a-4877-9878-
1044fc8e0db3",
                },
                "firstname": "Second",
                "lastname": "Signer",
                "email": "second.signer@example.it",
                "phone": "+393927654321",
                "status": "pending",
                "iframe_src":
"https://staging-app.yousign.com/procedure/sign?members=/members/3af46b58-8c3a-4877-9878-
1044fc8e0000",
                }
        ]
}
```

### GET /procedure/status

callback endpoint used as webhook for receiving procedure status from yousign when a specific event occurs. Received informations can be used for triggering notifications.

### How to get signed pdf and proof files:

### GET signed pdf file from Yousign:

```
curl --request GET \
    --url {base_api_url}files/{file_id}/download \
    --header 'Authorization: Bearer {api_key}' \
    --header 'accept: application/json'
```

### GET proof file containing information about the completed procedure:

```
curl --request GET \
    --url {base_api_url}/members/{member_id}/proof \
    --header 'Authorization: Bearer {api_key}' \
    --header 'accept: application/json'
```