A-Trust Gesellschaft für Sicherheitssysteme
im elektronischen Datenverkehr GmbH
Landstraßer Hauptstraße 5
A-1030 Wien

https://www.a-trust.at
E-Mail: office@a-trust.at

A-Trust signature server REST interface

Developer Manual

Version: 2.0
Date: May 4, 2023

Contents

Datum	Rev	Autor	Änderungen
19.10.2017	0.1	Joel Chinnow	First draft
23.10.2017	0.2	Joel Chinnow	Finalized batch signature
22.09.2021	0.3	Joel Chinnow	Removed WBPK and minor fixes
28.10.2021	0.4	Joel Chinnow	Add seal signature
25.08.2022	0.5	Joel Chinnow	Add template format description
04.05.2023	2.0	Joel Chinnow	New API version

Table 1: document history

1 Overview

This Document describes the REST interface of the signature server.

1.1 Quick reference

URL	Method	Functionality
signaturebatches	POST	Open a new batch
signaturebatches/<ticket>/documents	POST	Add a document to a given batch
signaturebatches/<ticket>/mobilesignature	POST	Sign the batch with Handy-Signatur
signaturebatches/<ticket>/sealsignature	POST	Sign the batch with a.sign Seal
signaturebatches/<ticket>/documents/<id>	DELETE	Get the signed document
signaturebatches/<ticket>/certificate	GET	Get the used certificate
templates	POST	Add template
templates	GET	List templates
templates/<id>	GET	Get template
templates/<id>	PUT	Replace template
templates/<id>	DELETE	Delete template

1.2 API Key

Every call has to be authorized with an API key. Therefore, the X-API-KEY HTTP header has to be set to the corresponding API key. An API key for testing can requested from A-Trust.

1.3 Generic return codes

The following return codes can be returned for every REST call. Therefore, they are not explicitly mentioned.

- 401 UNAUTHORIZED Call not allowed
- 500 INTERNAL SERVER ERROR Error handling the request

2 Batch signature

Signing a batch is done with the following steps. First, the batch is opened. Second, the documents to be signed are added to the batch. At least one document is required, but up to 200 documents can be added. Afterwards, the batch has to be closed. This can be done either by starting a mobile signature or by starting a seal signature. Finally, the signed documents can be downloaded.

2.1 State - initial

2.1.1 Create a new signature batch

To be able to sign documents, it is required to open a batch. This can be done with a HTTP POST call. Two HTTP multipart/form-data parameter are supported:

- RedirectUrl The user will be redirected to the URL after a successful signature
- ErrorUrl The user will be redirected to the URL when the signing process fails

```
POST /v2/signaturebatches HTTP/1.1
Host: testbox.a-trust.at
Cache-Control: no-cache
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW
x-api-key: YOUR_API_KEY

------WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="RedirectUrl"

https://success.a-trust.at
------WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="ErrorUrl"

https://failure.a-trust.at
------WebKitFormBoundary7MA4YWxkTrZu0gW--
```

Listing 1: Open batch request

After a successful HTTP request, it is possible to add documents to the batch. When opening the batch, a time-out of 60 minutes is set. If the time-out is reached, the batch and the corresponding documents are removed.

If the creation of the batch was successful, the server responds with HTTP status 201. The response contains a location header with the URI of the batch (ticket).

```
Status: 201 Created
Location: https://testbox.a-trust.at/v2/signaturebatches/TICKET
```

Listing 2: Open batch response

Additional errors:

400 Bad Request Missing parameter

2.2 State - batch open

2.2.1 Add document

This call is used to add documents to an open signature batch. It is required to add at least one document. The maximum number of documents supported by the Handy-Signature is 200. There are three options to use this call: The first option is just to use the required parameter (location, reason and document). This will result in an invisible signature. As second option, the ID of the template can be set additionally with the template Parameter. In this case, a figurative mark is place on the PDF. It is placed on the default location as defined in template. As third option, the placement options for the figurative mark can be given explicitly. Hereby, all placement parameters are required. After all documents have been added, the batch can be signed with Handy-Signature or (if configured) with a.sign Seal.

Name	Type	Description
document	File	The document to sign
location	string	Location of the signature
reason	string	Reason of the signature
template	string	ID of the template
x	string	X0 in userspace units
y	string	Y0 in userspace units
w	string	X1 in userspace units
h	string	Y1 in userspace units
page	string	page of the signature seal

A simple request looks like:

```
POST /v2/signaturebatches/TICKET/documents HTTP/1.1
Host: testbox.a-trust.at:443
Cache-Control: no-cache
Postman-Token: c023c7c8-eac1-d730-1ec3-c7dc78461d4b
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW
x-api-key: YOUR_API_KEY

------WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="document"; filename="diagnose.pdf"
Content-Type: application/pdf

------WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="location"

The location of the signature
------WebKitFormBoundary7MA4YWxkTrZu0gW
Content-Disposition: form-data; name="reason"

The reason of the signature
------WebKitFormBoundary7MA4YWxkTrZu0gW--
```

Listing 3: Add document request

A successful request is answered with 201 CREATED. The answer contains a location

header with the URL of the added document.

```
Status: 201 Created
Location: https://testbox.a-trust.at/v2/signaturebatches/TICKET/documents/0
```

Listing 4: Add document response

Additional errors:

400 Bad Request Missing / invalid parameter

404 Not Found Ticket not found

415 Unsupported Media Type Problems with PDF parsing

413 Request Entity Too Large PDF size above the limit

424 Failed Dependency Batch already signed

2.2.2 Sign batch with Handy-Signature

To close the the batch and sign the contained documents, an other REST call is required:

```
POST /v2/signaturebatches/TICKET/mobileSignature HTTP/1.1
Host: testbox.a-trust.at
Cache-Control: no-cache
x-api-key: YOUR_API_KEY
```

Listing 5: Sign batch request

It is possible to configure the parameter of the Handy-Signature iframe with the multipart/form-data parameter handysignaturparameter. Setting it to ẅidth=300&height=400&backgroundcolor=blue define the width, height, backgroundcolor and fill the mobilephonenumber field. The Details can be found on the A-Trust Website https://labs.a-trust.at/developer/Handy-Signatur.aspx.

A successful request is answered with 201 CREATED. The answer contains a location header with an URL (which includes the required session id) to start the Handy-Signature (or ID Austria) process:

```
Status: 201 Created
Location:https://www.handy-signatur.at/mobile/https-security-layer-request/main.aspx?sid=HS_SESSION_ID
```

Listing 6: Sign batch response

Additional errors:

404 Not Found Ticket not found

424 Failed Dependency No documents added

2.2.3 Sign batch with a.sign Seal

Instead of Handy-Signature, the documents can be signed with a corporate certificate (a.sign Seal). Therefore, instead of mobilesignature, the sealsignature REST call should be used. It is possible to use one certificate per API Key. The configuration has to be done by A-Trust.

This call closes the the batch and signs the documents. These can be retrieved in the same way as after a Handy-Signatur, no further action is required. A sample request looks like:

```
POST /v2/signaturebatches/TICKET/sealsignature HTTP/1.1
Host: testbox.a-trust.at
Cache-Control: no-cache
x-api-key: YOUR_API_KEY
```

Listing 7: Sign batch request

A successful request is answered with 201 CREATED:

```
Status: 201 Created
```

Listing 8: Sign batch response

Additional errors:

404 Not Found Ticket not found

424 Failed Dependency No documents added

2.3 State - signature finished

2.3.1 Get a signed document

When the signature process is finished, the signed documents can be received with an HTTP DELETE call. It has to contain the ticket and the document id:

```
DELETE /v2/signaturebatches/TICKET/documents/DOCUMENT_ID HTTP/1.1
Host: testbox.a-trust.at:443
Cache-Control: no-cache
x-api-key: YOUR_API_KEY
```

Listing 9: Get signed document request

When the call was successful, the server responds with 200 OK. Th signed document is attached.

Additional errors:

404 Not Found Ticket / document id not found

2.3.2 Get signer certificate

After a signature, the signer certificate can be requested.

```
GET /v2/signaturebatches/TICKET/certificate HTTP/1.1
Host: testbox.a-trust.at:443
Cache-Control: no-cache
x-api-key: YOUR_API_KEY
```

Listing 10: Get certificate request

If successful, the server responds with 200 OK and an attached certificate.

Additional errors:

404 Not Found Ticket not found

3 Templates

The following calls can be used to manipulate the available templates on the signature server.

3.1 List templates

The call gets a list of all available templates for the API Key. A request looks like:

```
GET /v2/templates
Host: testbox.a-trust.at:443
Cache-Control: no-cache
x-api-key: YOUR_API_KEY
```

Listing 11: List template request

In case of a successful call, the server responds with 200 OK. The list of templates is returned in JSON format:

```
{
"templateList":[
   {"id":"1", "description":"Default Template"},
   {"id":"2", "description":"Another Template"},
   {"id":"3", "description":"Next Template"}
]
}
```

Listing 12: List template response

3.2 Add template

The call adds a template on the signature server. During the upload, it is validated against the corresponding xsd file. If the validation is successful, the template is stored and an id returned. A request looks like:

```
POST /v2/templates
Host: testbox.a-trust.at:443
Cache-Control: no-cache
x-api-key: YOUR_API_KEY
```

Listing 13: Add template request

In case of a successful call, the server responds with 201 CREATED. The location header contains the URL of the template.

```
Status: 201 Created
Location: https://sigserver.at/v2/templates/42
```

Listing 14: Add template response

Besides the standard error codes, the following return codes are possible:

400 Bad Request No template given

415 Unsupported Media Type The template could not be validated

3.3 Get template

The call can be used to download a template with a given id.

```
GET /v2/templates/<id>
Host: testbox.a-trust.at:443
Cache-Control: no-cache
x-api-key: YOUR_API_KEY
```

Listing 15: Get template request

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Content-Type: application/xml; charset=utf-8


<xml version="1.0" encoding="UTF-8"?>\r\n\<SignatureTemplateStyle...
```

Listing 16: Get template request

In case of a successful call, the server responds with 200 OK.

Besides the standard error codes, the following return codes are possible:

404 Not found The template could not be found

3.4 Replace template

The call can be used to change a template with a given id.

```
PUT /v2/templates/<id>
Host: testbox.a-trust.at:443
Cache-Control: no-cache
x-api-key: YOUR_API_KEY
```

Listing 17: Replace template request

```
200 OK
```

Listing 18: Replace template request

In case of a successful call, the server responds with 200 OK.

Besides the standard error codes, the following return codes are possible:

400 Bad Request No template given

404 Not found The template could not be found

415 Unsupported Media Type The template could not be validated

3.5 Delete template

The call can be used to delete a template with a given id on the server.

```
DELETE /v2/templates/<id>
Host: testbox.a-trust.at:443
Cache-Control: no-cache
x-api-key: YOUR_API_KEY
```

Listing 19: Delete template request

```
200 OK
```

Listing 20: Delete template request

In case of a successful call, the server responds with 200 OK.

Besides the standard error codes, the following return codes are possible:

404 Not found The template could not be found

4 Template format

Templates are defined as XML files, e.g.:

```
<?xml version="1.0" encoding="UTF-8"?>
<SignatureTemplateStyle>
<Description author="A-Trust GmBh" name="Sisi" description="A-Trust
    Standard Signatur Siegel" />
<Position type="relative" x="40.0" y="40.0" w="50.0" h="50.0" page="1" />
<B64bgImage>
iVBORw0KGgoAAAANSUhEUgAAG [...] jhz/DwF8ylxAZ19
</B64bgImage>
<Fields>
<Image x="150.0" y="0.0" w="800.0" h="200.0">
/9j/4AAQSkZJRgAB [...] ABRRRQAUUUUAFFFFAH/9k=
</Image>
</Fields>
</SignatureTemplateStyle>
```

The individual elements are described as follows:

4.1 Description

This is where meta-information, including author, name and description, is stored. The name is used by *ListTemplate*.

4.2 Position

This is where the position of the signature image in the PDF document is defined. The x and y coordinates define a corner of the signature image. It should be noted that the coordinate origin is the lower left corner of the PDF file. W and h define the image's second corner. These coordinates are expressed as percentages (page height / page width). The page onto which the signature image is to be placed has to be defined as well. Instead of 'relative', the position of the signature image may also be defined as 'absolute'.

4.3 B64bgImage

This is where the actual background image is saved as base64. Allowed file formats for background images are .png and .jpg.

4.4 Optional elements

4.4.1 Additional images

Further images may be added to the background image. These are defined in the <Fields> tags:

```
<Image x="0" y="0" w="150" h="150">Base64Content</Image>
```

The image itself is written into the XML document encoded in base64. All acceptable image formats only support JPEG and PNG. The x and y coordinates indicate the upper left point of the image in the background image. The w and h coordinates define the image extents. If they do not match the height or width of the image itself, the image is stretched or compressed. All coordinates (x, y, w, h) are expressed in pixels.

Examples

The following examples are based on a background image with a resolution of 800x300 and an additional image with a resolution of 100x100.

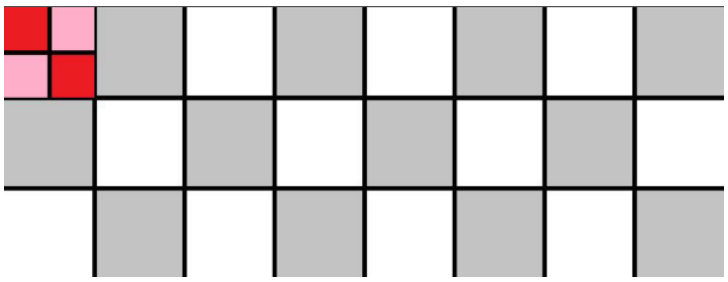

Figure 1: Original size image (X=0, Y=0, W=100, H=100)

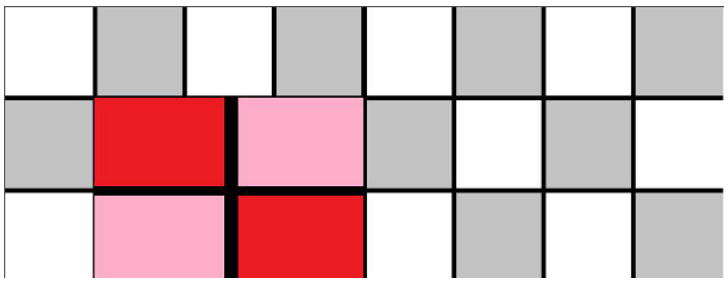

Figure 2: Stretched image (X=100, Y=100, W=300, H=200), as the original image measured 100x100 in size

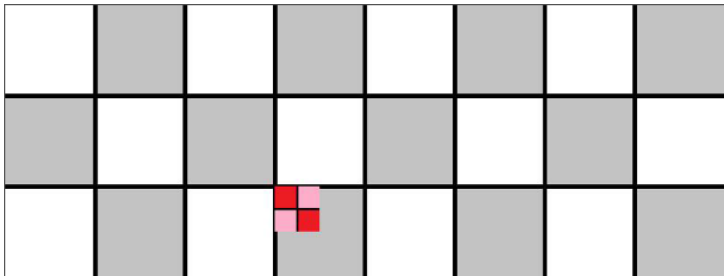

Figure 3: Compressed image (X=300, Y=200, W=50, H=50), as the original image measured 100x100 in size

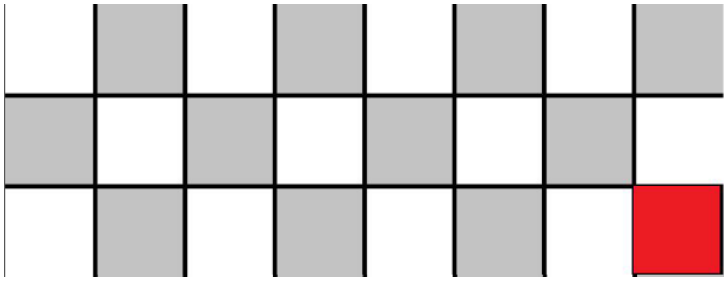

Figure 4: Image out of range* (X=700, Y=200, W=200, H=200) Only the upper left part of the image is shown here; the remainder is outside the visible range.

4.4.2 Additional text

The text written on the signature image can also be defined. There are different text types: 'static' and 'dynamic'. The following attributes are needed:

- type – static, dynamic (signer, location, reason, timestamp may also be used if no additional text is to be used with one of the dynamic elements)
- x – The x position on the background image (coordinate origin is at the top left-hand corner)
- y – The y position on the background image (coordinate origin is at the top left-hand corner)
- w – width of the text field
- h – obsolete
- center – If the text is smaller than the text field, it can be centred using True.
- wrap – obsolete
- fgColor – as RGB values, e.g. (255, 255, 255)

- bgColor – as RGB values, e.g. (255, 255, 255)
- textSize - the size of the text (optional)

Type explained

- static - static text
- dynamic - Dynamic information can be written onto the signature image. For now, the following information is supported:
  - %timestamp – The time at which the signature is implemented (in the format DD.MM.YYYY HH:MM MET)
  - %signer – The signer implementing the signature (common name from the signature's certificate)
  - %location – The location where the signature is placed (from the method call)
  - %reason – The reason for the signature (from the method call)
- signer - short form for signer
- location - short form for location
- reason - short form for reason
- timestamp - short form for timestamp

Examples

A text field with text and the signer:

```
<Label type="dynamic" x="50.0" y="50.0" w="0.0" h="0.0" center="false"
    wrap="false" fgColor="(0,0,0)" bgColor="(255, 255, 255)
    textSize="26">Signiert von: %signer</Label>
```

A text field with only the signer:

```
<Label type="signer" x="50.0" y="50.0" w="0.0" h="0.0" center="false"
    wrap="false" fgColor="(0,0,0)" bgColor="(255, 255, 255) textSize="26"/>
```