

# Identity Attestation API

Simple attestation API to allow multiparty document signing with validation.

**Default response content-types:** application/json

**Schemes:** http

## Summary

### Tag: SignedDocuments

Operation	Description
GET /doc	Get signed documents available in the system.
POST /doc	Create new signed documents
PUT /doc	Updates a signed document.

### Tag: Signer

Operation	Description
GET /signer	Get signers registered.
POST /signer	Create new signer
PUT /signer	Updates an existing signer.

## Paths

### GET /doc

Tags: SignedDocuments

Get signed documents available in the system.

### DESCRIPTION

This will return all the signed documents in a specific collection.

**REQUEST PARAMETERS**

Name	Description	Type	Data type
id	The unique identifier of the document to retrieve.	query	<i>number</i> (int)

**RESPONSES**

Uses default content-types: `application/json`

**200 OK**

An array of signed documents.

**ITEMS**

*SignedDocument*

**POST /doc**

Create new signed documents

Tags: SignedDocuments

**DESCRIPTION**

This will create new signed documents that will be validated.

**RESPONSES**

Uses default content-types: `application/json`

**201 Created**

The new document is created.

**PUT /doc**

Tags: SignedDocuments

Updates a signed document.

**DESCRIPTION**

This will update existing signed documents, used by participants and validators.

**RESPONSES**Uses default content-types: `application/json`**200 OK**

The document is updated/replaced.

**GET /signer**

Tags: Signer

Get signers registered.

**DESCRIPTION**

This will return all the signers in a specific collection.

**REQUEST PARAMETERS**

Name	Description	Type	Data type
id	The unique identifier of the signer to retrieve.	query	<i>string</i>

**RESPONSES**Uses default content-types: `application/json`**200 OK**

An array of signers.

**ITEMS***Signer***POST /signer**

Create new signer

Tags: Signer

**DESCRIPTION**

This will create a new signer profile, that can sign documents.

**RESPONSES**

Uses default content-types: `application/json`

**201 Created**

The new document is created.

**PUT /signer**

Updates an existing signer.

Tags: Signer

**DESCRIPTION**

This will update the signer profile.

**RESPONSES**

Uses default content-types: `application/json`

**200 OK**

The document is updated/replaced.

# Schema definitions

**SignedDocument:** *object*

## PROPERTIES

**signers:** *object[]*

Unique identifier representing a user who will participate in the signing / validation.

## ITEMS

*Signer*

**chainpoint:** *boolean*

**data:** *string*

Container field for associated data with the document.

**id:** *string*

The globally unique document index in the backend repository.

**Signer:** *object*

## PROPERTIES

**id:** *string*

Identifier that represents a signer that will participate in the transaction. This id is globally unique.

**name:** *string*

Friendly name of the signer.

**signingUrl:** *string*

Url for signing.

