Table of Contents

- eSignatures Glossary
  - eSignatures Glossary
    - Actor
    - Administrator
    - Advanced electronic signature
    - API
    - API user
    - Asynchronous signing
    - Basic electronic signature
    - Browser Package
    - Callback URL
    - Choice of signing
    - Configuration Index
    - Contact
    - ContactGroup
    - Document
    - Document group
    - Document Portal
    - DSS
    - Elements
    - Ending
    - Face to face signing
    - Initiator
    - Instant Package
    - Legal notice
    - Mandated signing
    - Markers
    - Modal
    - Notification Callback URL
    - One-time password (OTP)
    - One-time URL (OTU)
    - Package
    - PDF
    - PDF/A

PersonGroup
Poison queue
Qualified electronic signature
QuickSigning
Receiver
Redirect URL
Server signing
Session
Signature policy
Signer
Signer Portal
Stakeholder
Status
Synchronous signing
Tenant user
Theme
Unique identifier
User
User Management
WebPortal
Worker
WYSIWYS
XML signing

eSignatures Glossary

In this section you'll find an eSignatures glossary.

Click the key words in the table of contents on the left for their definition.

Actor

Definition

In the eSignatures API, an *actor* is an object that provides information about what the stakeholder must do: sign the document or simply receive the signed document. The Actor object only contains information relevant to the action the actor will perform (sign or receive).

All actors belonging to a stakeholder are grouped in an **Actors** array.

Administrator

Definition

An *administrator*, or *admin* in short, is a user who is part of the **Administrators** user group.

An admin has access to the Config Index and has the required permissions to configure an eSignatures environment.

In the WebPortal an admin by default has access to the Document Portal, Signer Portal and User Management Portal.

User Types

The following user types are available in eSignatures:

- User
- API user
- Tenant user
- Administrator

Advanced electronic signature

Definition

An *advanced electronic signature* is also called a *digital signature*.

Advanced electronic signatures must meet specific requirements providing a higher level of signer ID verification, security, and tampersealing (meaning the document cannot be changed once it is signed).

More information

For more information, consult the following sources:

- Documentation topic: What is the legal value of a digital signature?
- Connective whitepaper: The power of an electronic signature?

API

Definition

API stands for *Application Programming Interface*. The eSignatures API allows API users to set up signing flows and have eSignatures communicate with other systems.

The eSignatures API can be integrated in other software to use eSignatures features.

API user

Definition

An *API user* is a user who has access to the eSignatures API.

Per eSignatures environment there is 1 user who has access to the API. The API user's credentials do not provide access to the WebPortal.

User Types

The following user types are available in eSignatures:

- User
- API user
- Tenant user
- Administrator

Asynchronous signing

Definition

Asynchronous signing is the process of signing documents in the background while the signer may do other tasks.

How does asynchronous signing work?

Once the signing process has started, a signer may close the signing session and work on other things. If the signer has access to the Web Portal, they can for instance upload new documents and send them for signing, while the signing of the previous document is still ongoing. They are no longer blocked until eSignatures is finished with the previous document.

Prerequisites

- eSignatures 5.2.0 or higher.
- The Connective WebSigner Worker must be installed and configured correctly. Consult **Connective - eSignatures 5.3.1 - Installation Documentation** to learn how to do so.

Limitations

- Asynchronous signing works with the following signing methods: manual, SMS OTP, Mail OTP, iDIN, pincode (through OpenID Connect), biometric.
- When using the eSignatures API, do *not* define a RedirectURL. The purpose of a RedirectUrl after all is that the signer is redirected to another URL after signing, so the signing session cannot be closed manually.

Queueing mechanism

In case asynchronous should fail, the documents end up in a poison queue, from which they can be resubmitted for processing.

Basic electronic signature

Definition

An *basic electronic signature* simply captures a person's intent to agree to the content of an electronic document or a set of data. It can be a signature manually drawn on a desktop screen, but also merely the image of your signature pasted in a Word document, or your mail signature.

A basic electronic signature is **not** a digital signature.

More information

For more information, consult the following sources:

- Documentation topic: What is the legal value of a digital signature?
- Connective whitepaper: The power of an electronic signature?

Browser Package

Definition

The *browser package* is the component a signer needs to install on Windows or macOS in order to use any signing method that requires additional hardware.

The signing methods that require additional hardware are currently eID, Manual+eID, BeLawyer and Biometric.

More information

See **Why do I need to install the Connective browser package?** for more information on how it works and how to install it.

Callback URL

Definition

A *callback URL* is a REST API URL to contact an external system and inform it a status change has occurred. Depending on how the external system is set up, it will take certain actions after it has been informed.

More information

For more information, see How does a callback URL work?.

Choice of signing

Definition

Choice of signing is the feature by which a signer may choose from at least 2 signing methods to sign their document.

When the signer has made their choice, they will only use that signing method in the current signing session.

Configuration Index

Definition

The *Configuration Index*, or *Config Index* in short, is a Web Portal where administrators and tenant users can configure the behavior of their eSignatures environment.

The Config Index consists of three parts:

- **Configuration**: contains all settings to configure eSignatures.
- **DSS**: contains all settings to set up and configure a DSS connection.
- **Theme**: contains all settings to rebrand an eSignatures environment.

Prerequisites

To access the **Configuration** and **DSS** sections of the Config Index, a user needs the following elements:

- An on-premise eSignatures environment. Hosted and Cloud clients do not have access. To modify their configuration, they need to contact Connective.
- Admin credentials to the Config Index.

To access the **Theme** section of the Config Index, a user needs:

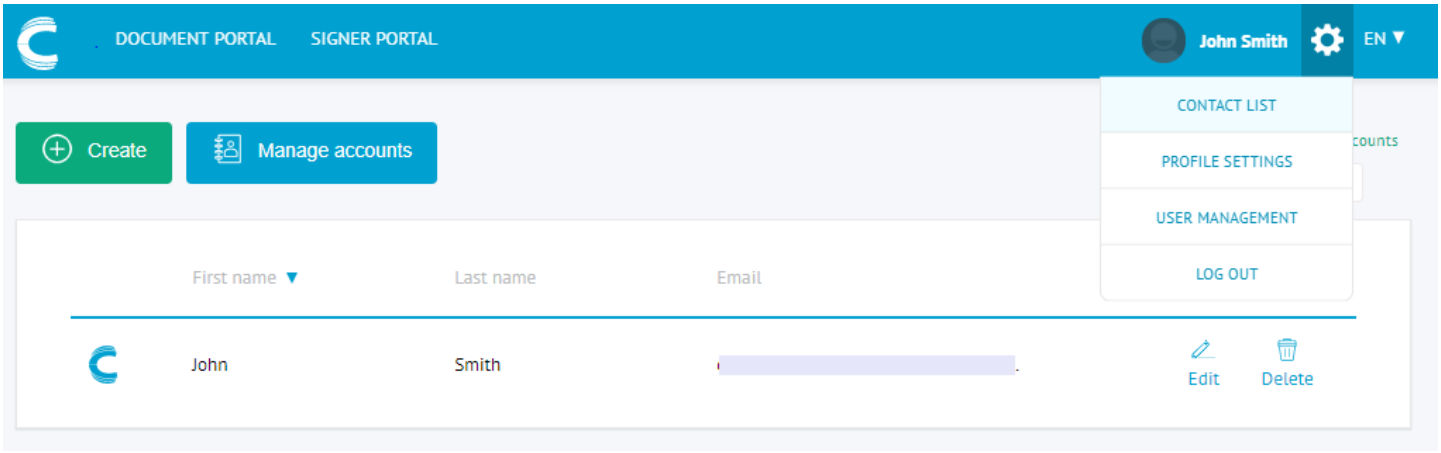
- Tenant user credentials

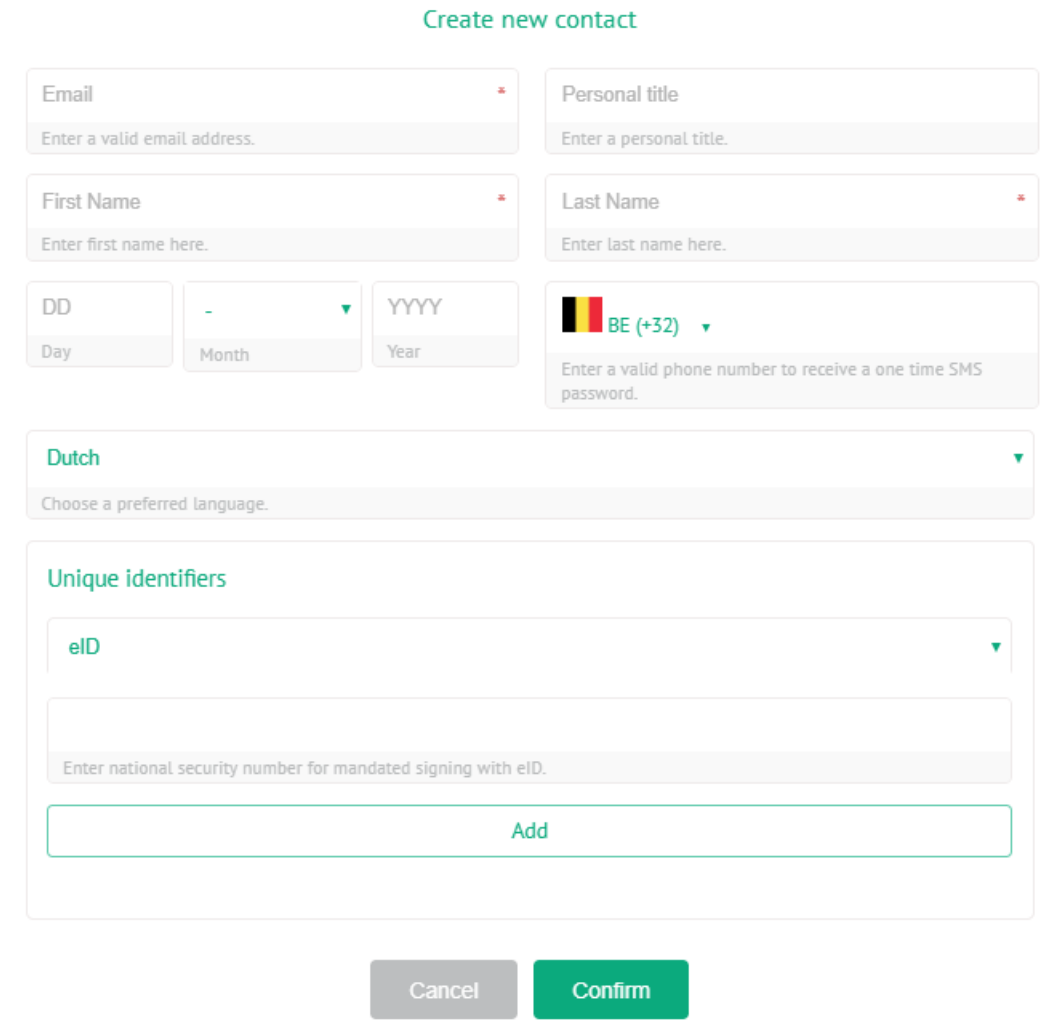
Contact

Definition

A *contact* in the eSignatures WebPortal is a person who can sign documents and receive signed documents. A contact contains at least the person's email address, first name and last name. Additionally, a contact may also contain a person's birth date, mobile phone number and the unique identifier of their eID or BeLawyer card.

To add a person as signer or receiver (or both), a contact must be created for that person. This is done in the **Contact List** section of the WebPortal.





Attention: a contact is not to be confused with a user. A contact does not necessarily have access to the eSignatures WebPortal.

More information

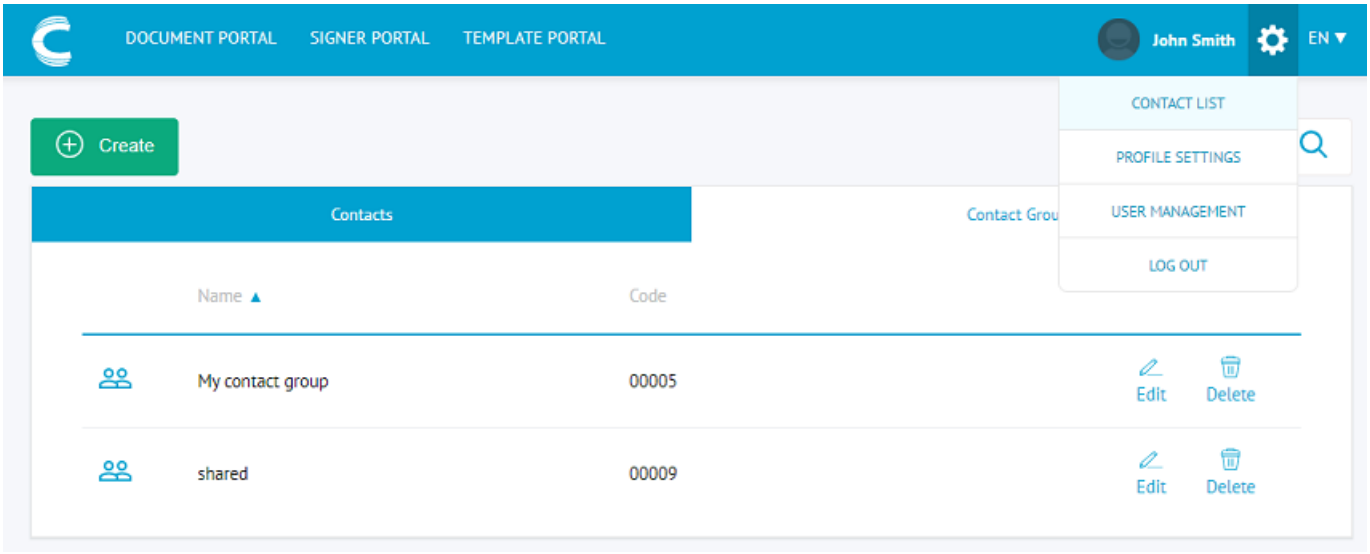
For more information about creating contacts, see the Managing contacts and contact groups topic in the User Documentation.

Contact Group

Definition

A *contact group* in the eSignatures WebPortal is a collection of contacts. A contact group can be added to a signing field, in the same manner as a single contact. The difference is that any member of the contact group can sign for the entire group.

Creating contact groups is done in the **Contact List** section of the WebPortal.



Tip: when a contact group has been created, a code is assigned to it. This code can be used in the API to send packages to the required contact group.

Attention: a *contact group* is not to be confused with a *person group*.

More information

For more information about creating contacts, see the Managing contacts and contact groups topic in the User Documentation.

Document

Definition

In eSignatures, a *document* is a computer file that will be signed digitally. eSignatures supports the following file formats: .pdf, .docx, .doc, .txt and .xml*.

In the eSignatures API, a document is always part of a package or instant package.

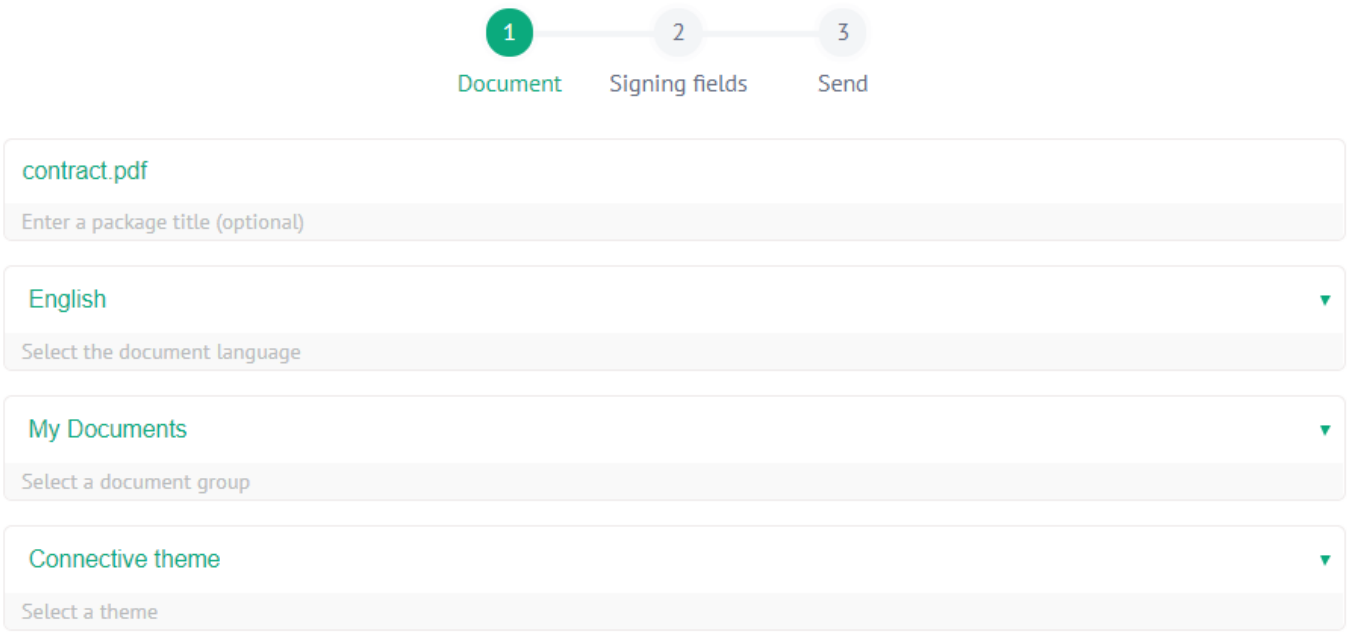
*.xml files can only be sent through the API, not via the WebPortal.

Document group

Definition

A *document group* is a repository to which an initiator uploads their documents.

The document group to which the documents will be uploaded is selected at step 1 of the upload process.



If multiple themes have been created for the eSignatures environment, the initiator can select in which theme the WYSIWYS must be shown.

Document Portal

Definition

The *Document Portal* is the part of the eSignatures WebPortal where users can upload documents and send them for signing.

A user who uploads and sends documents for signing is called an initiator in eSignatures terminology.

Prerequisites

To access the Document Portal, a user needs the following elements:

- An eSignatures account
- Valid credentials to the account

User actions

A user can do the following actions in the Document Portal:

- Upload documents, (And all related actions such as adding signers, adding signing methods, etc.)
- Sign documents
- Revoke documents
- Notify signers there are documents to be signed
- Delete signed documents

DSS

Definition

DSS stands for *Digital Signature Services*. DSS is Connective's digital signing service that actually signs the documents processed by eSignatures.

Connective's DSS is based on the European Commission's DSS open-source library.

Elements

Definition

In the eSignatures API, a *element* is an item that is placed on a document and may be assigned to an Actor

An element cannot exist on its own, it is always part of a document.

A document may contain multiple elements grouped in an Elements array.

Prior to eSignatures 6.3, one type of element is supported: SigningField.

As of eSignatures 6.3, two new element types have been added: TextBoxField and CheckBoxField.

SigningField elements determine where on the document a signature must be placed. SigningFields are always mandatory.

TextBoxField elements determine where on the document a textbox field will be placed. A textbox may contain a default, prefilled value, which can be modified afterwards by the end user. A textbox may be mandatory or optional.

CheckBoxField elements determine where on the document a checkbox field will be placed. A checkbox may be already checked by default, but can still be modified by the end user. Like textboxes, checkboxes may also be mandatory or optional.

Ending

Definition

A Package can be *Ended* by the initiator. With this action, the initiator indicates that the process has stopped.

An initiator can end the process both through the API as through the webPortal

Face-to-face signing

Definition

Face-to-face signing is the process by which an initiator lets one or more signers - who are physically present - sign their document(s) in the initiator's Document Portal.

Attention: when signing a document face-to-face, all signing fields for all signers of the document are enabled. So make sure each signer selects the correct signing field.

Initiator

Definition

An *initiator* is a user who uploads documents and sends them for signing. This may be done through the eSignatures API or through the WebPortal.

When sending documents through the API, the documents are added to the initiator's account in the WebPortal.

Instant package

Definition

An *instant package* is a package created with a single API call: Instant Package Creation.

Contrary to a "regular package", an instant package always contains 1 document.

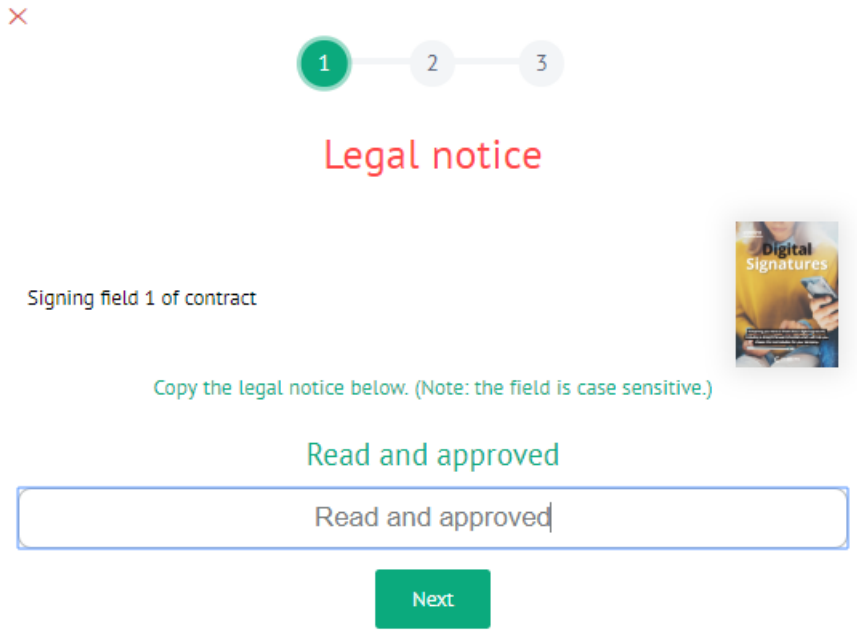
Apart from that, instant packages are processed just like regular packages.

Legal notice

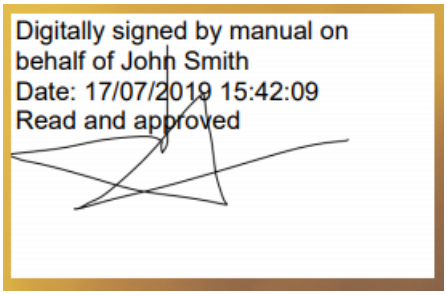
Definition

A *legal notice* is text an initiator may add to a document, forcing the signer to agree with a certain statement.

When a document contains a legal notice, the signer must retype the exact content of the legal notice before they are able to place their signature. Note that legal notices are case-sensitive.



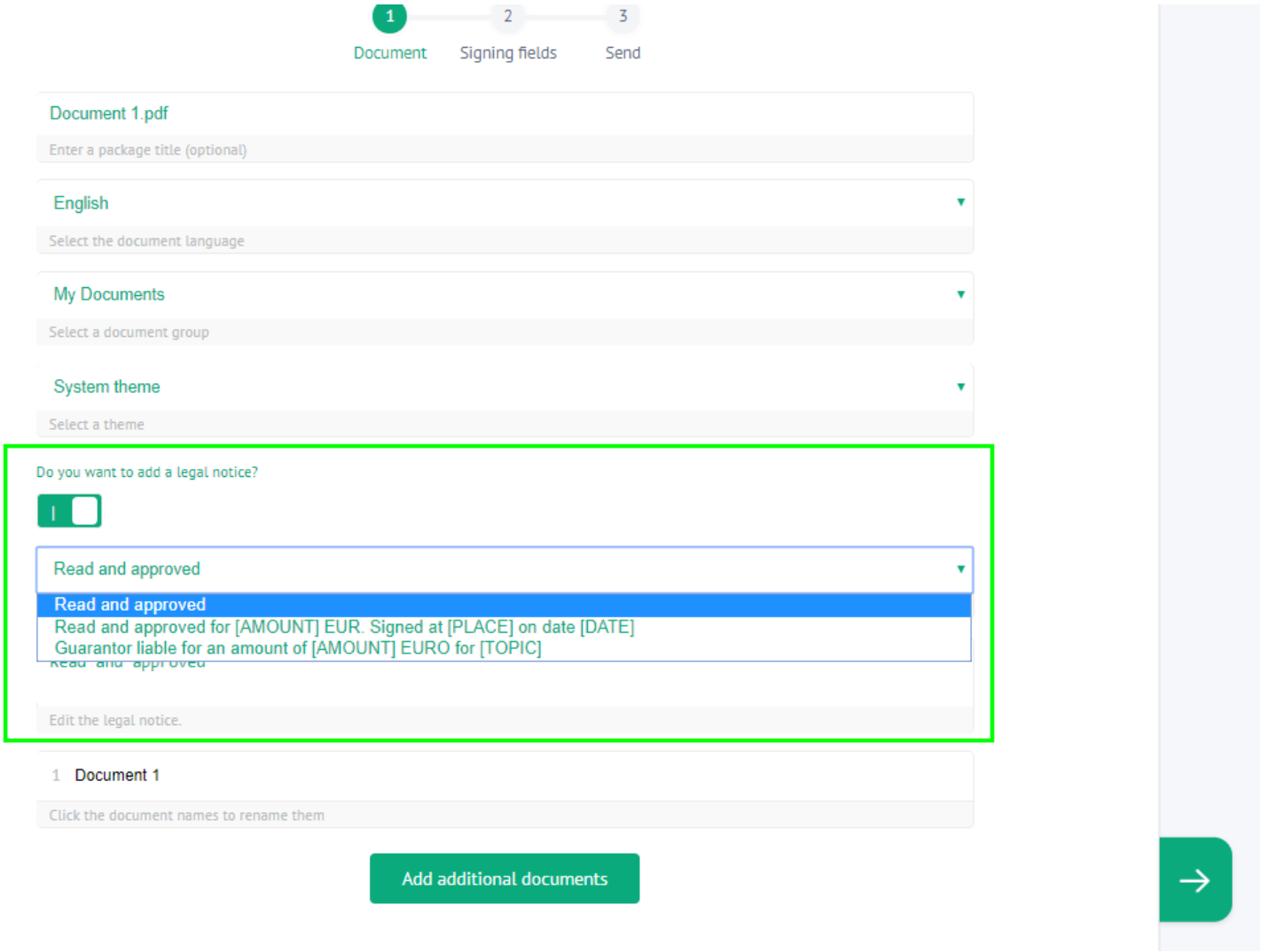
When the document is signed, the content of the legal notice is added to the signature field.



Configuration

In the Config Index, the admin can configure 3 legal notices initiators will be able to choose from at step 1 of the upload process. Note however that each type of legal notice can still be edited.

When adding a legal notice through the API, the API user can also choose from the 3 configured legal notices, or add a custom one.


1
Document
2
Signing fields
3
Send
Document 1.pdf
Enter a package title (optional)
English
Select the document language
My Documents
Select a document group
System theme
Select a theme
Do you want to add a legal notice?
Read and approved
Read and approved
Read and approved for [AMOUNT] EUR. Signed at [PLACE] on date [DATE]
Guarantor liable for an amount of [AMOUNT] EURO for [TOPIC]
Edit the legal notice.
1 Document 1
Click the document names to rename them
Add additional documents

Mandated signing

Definition

Mandated signer validation, also commonly referred to as *mandated signing*, is an extra check eSignatures can do to verify if a signer is mandated to sign during a particular signing session.

More information

For more information, see:

- How does mandated signer validation work? (**prior to esIgnatures 6.2**)
- How does mandated signer validation work? (**as of esIgnatures 6.2**)

Markers

Definition

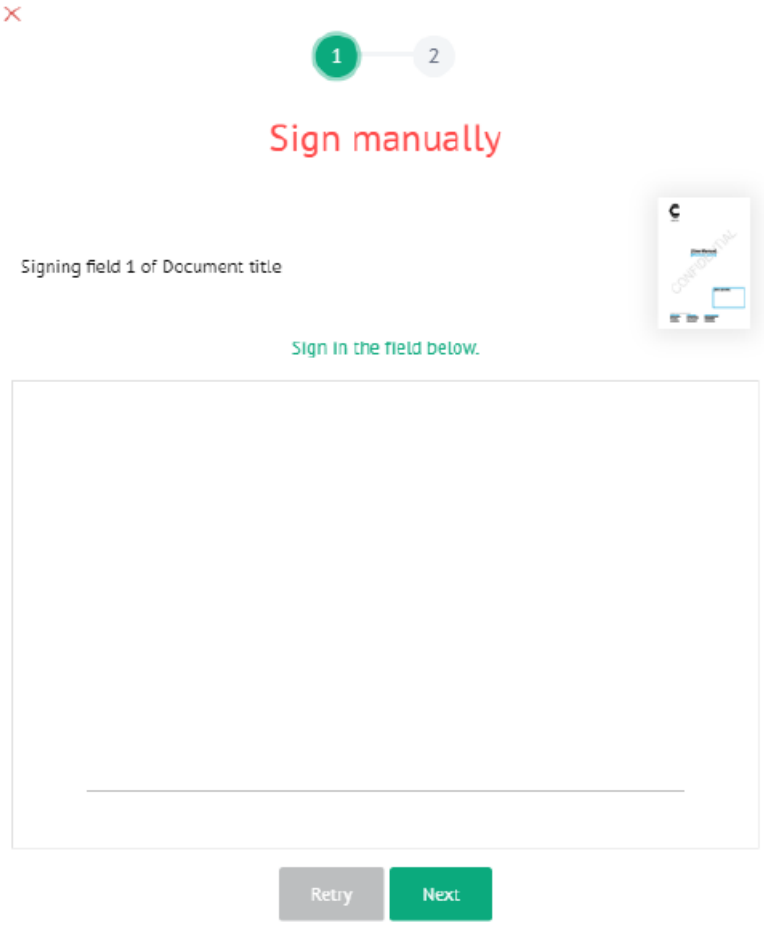
A text marker, commonly referred to as *marker* is a piece of text a user can add to a document to determine where a signature field must be placed, and which dimensions it must have.

The location in the document where the marker is placed, is where the signature field will be placed. Hence, markers are especially convenient in documents with a dynamic/variable structure.

Modal

Definition

In eSignatures terminology, a *modal* is a pop-up window. Modals are opened when a signer selects a signing method, or when a user creates a contact for instance.



Notification Callback URL

Definition

A *notification callback URL* is a parameter in the API that can be used to override the standard behavior of sending out emails when a user requests a new signing link or a new download link after they have expired.

The standard behavior is that eSignatures always sends a new email to the signer whenever they request a new signing link or download link. By means of a notification callback URL, eSignatures can trigger a remote service that will handle the request. How it handles the request depends entirely on its setup.

More information

For more information, see section 5.1.12 of the API documentation.

One-time password (OTP)

Definition

A *one-time password*, or OTP in short, is a password that is valid for only one signing session, during a (configurable) period of time.

The administrator can determine in the Config Index how long a one-time password stays valid, but once a signing session has started it is only valid during that signing session.

In eSignatures a one-time password is used for Mail OTP signing and SMS OTP signing.

One-time URL (OTU)

Definition

A *one-time URL*, or *OTU* in short, is a specially crafted web address that is valid for one-time use only. The links signers receive by email to sign their documents and the link they receive to download their documents (after they've been signed) are one-time URLs.

A soon as a signer has clicked a one-time URL, the URL becomes invalid and they will not be able to use it again.

When a user encounters an expired URL, they can always request a new email containing a new one-time URL to access their document.

Package

Definition

In eSignatures, a *package* is a collection of digital documents that need to be signed.

How to create a package?

In the WebPortal, a package is created automatically as soon as a user uploads multiple documents in one go.

In the API, a package is created by means of the Create Package call. Afterwards the user can add documents to the package using Add document to package calls.

Note that in the API you can also create an Instant Package - which always contains 1 document - using a single API call.

PDF

Definition

PDF stands for "Portable Document Format". PDF is a file format designed to present documents consistently across multiple devices and platforms.

Documents sent through the API or uploaded to the WebPortal are converted to PDF format. It is also possible to select PDF/A as output format.

PDF/A

Definition

PDF/A is an ISO-standardized version of the Portable Document Format (PDF) which restricts features that can be used in a regular PDF, optimizing it for long-term accessibility and reproducibility of the content.

In other words PDF/A is a long-term preservation format.

Different PDF/A formats

eSignatures supports the following types of PDF/A formats, both as input format and output format:

- PDF/A-1a
- PDF/A-2a

PDF/A-1a

PDF/A-1 is a constrained form of Adobe PDF version 1.4 intended to be suitable for long-term preservation of page-oriented documents for which PDF is already being used in practice. The ISO standard [ISO 19005-1:2005] was developed by a working group with representatives from government, industry, and academia and active support from Adobe Systems Incorporated.

PDF/A-2a

PDF/A-2 is a constrained form of Adobe PDF version 1.7 (as defined in ISO 32000-1).

How to check whether a file is PDF/A compliant?

- You can use the VNeeS checker application, which identifies non PDF/A-compliant files in the validation report.
- You can also check the PDF/A-specific metadata in the PDF/A file.

Person group

Definition

A **person group** is a list of persons an API user adds to a Set Process Information or Instant Package Creation call.

Any persons listed in the person group will be able to sign the package for the entire group. In this sense, person groups function in the same way as contact groups.

Note that person groups are only used in the API.

Poison queue

Definition

The *poison queue* is the queue is which poison messages end up when an asynchronous operation failed. A poison message is a message that has exceeded the maximum number of delivery attempts to eSignatures.

In the API, an API user can use different calls to retrieve the contents of the poison queue, resubmit the contents to the desired command queue, and clear the contents of the poison queue.

More information

For more information on how the queuing mechanism works in eSignatures, see the Queuing Mechanism section in the API documentation.

Qualified electronic signature

Definition

A *qualified electronic signature* or non-repudiation Digital Signature is the only electronic signature type to have special legal status in EU. Unlike the other signatures, the burden of proof lies with the party that disputes the signature(s), not with the initiator. This makes it legally equivalent to a written signature. It is backed by a certificate issued by a Qualified Trust Service Provider (QTSP) that is on the EU Trust List (EUTL) and thus certified by an EU member state.

More information

For more information, consult the following sources:

- Documentation topic: What is the legal value of a digital signature?
- Connective whitepaper: The power of an electronic signature?

QuickSigning

Definition

QuickSigning is the process by which a signer signs all documents within a package, using a single signing action.

Note: for QuickSigning to work, a number of conditions must be met.

Receiver

Definition

A *receiver* in eSignatures terminology is a person who receives a copy of the fully signed document. A receiver does not sign documents.

Prerequisites

For a person to be a receiver:

- In the WebPortal: a contact must be created in the WebPortal containing at least the person's email address, first name and last name.
- In the API:
    - An Actor of the type "receiver" must be added to the Stakeholder array in a Set Process Information call or Instant Package Creation call.
    - The Stakeholder array must contain at least the person's email address, first name and last name.

  Note: when you add an Actor of the type "signer" in the API, an actor of the type "receiver" will also be created per definition.

Redirect URL

Definition

A *redirect URL* is the web address a signer's web browser is redirected to after the signing process is finished. In other words, it's the web page that is opened after the signing is done.

Attention: a redirect URL is not to be confused with a callback URL.

Server signing

Definition

Server signing is the process by which instant packages are signed autonomously by eSignatures, without signer interaction.

How does server signing work?

An API user sets the **SigningTypes** parameter to "Server" and sends their document through an Instant Package Creation call. As soon as the InstantPackage is set to status 'Pending' it is signed autonomously by eSignatures.

The actor defined as signer in the InstantPackage call receives a download link via secure email to download their signed document. Note that the signer does not receive a mail to sign their document, since the signing is done autonomously.

Limitations

The following limitations apply to Server signing:

- Only available through the API.
- Can only be used if all signing locations in the InstantPackage use "Server" as **SigningTypes** parameter.
- Cannot be combined with choice of signing (since choice of signing requires user interaction)

Session

Definition

A *session* is the time frame for communication between two devices, two systems or two parts of a system.

Which sessions are used?

In eSignatures, the following sessions are used:

- Login session
- WYSIWYS session
- Signing session
- SMS code validity session
- Mail OTP code validity session
- One-time URL session

More information

Click here for more information about each session.

Signature Policy

Definition

A *signature policy* is a document containing a set of rules that detail the terms and conditions of how a valid signature should be created and validated.

How to use a signature policy?

Currently, a default signature policy is used for itsme signing only.

If clients want to implement a custom signature policy they need to contact Connective.

Signer

Definition

A *signer* is a person who digitally signs a document in the WYSIWYS, whether they access it through the eSignatures WebPortal, via email or in an app.

Note that a signer doesn't necessarily have access to the eSignatures WebPortal.

Signer Portal

Definition

The *Signer Portal* is the part of the eSignatures WebPortal where users can access their documents that need to be signed.

Prerequisites

To access the Signer Portal, a user needs the following elements:

- An eSignatures account
- Valid credentials to the account

User actions

A user can do the following actions in the Signer Portal:

- View the documents that require their signature
- Filter documents based on their status
- Sign documents
- Download signed documents

Stakeholder

Definition

In the eSignatures API, a *stakeholder* is an object that provides information about any person who is involved with a package.

All stakeholders involved with a package are grouped in a **Stakeholders** array in the API.

Status

Different objects can go through different statuses in eSignatures throughout the lifecycle.

Within the eSignatures portal, a couple of statuses have been merged together as the technical status offers no additional value to the end-user.

OBJECT	PORTAL / API	STATUS	DEFINITION
Package / Document	Portal API	Draft	A document/package is in draft status when it has been uploaded to the portal but hasn't been finalized yet. In the API, a package is in draft as soon as it has been created using a Create Package call. A draft document/package can still be edited and must be finalized before signers are able to sign it. A draft document/package can be deleted.
	Portal API	Pending	The difference between *Pending_Approval* and *Pending_Signing* is introduced as of API v4. Within the portal and API versions before API v4, all packages / documents on which an actor can still take an action are considered to be *Pending*. In the API a user can also set a package's status to pending using a Set Package Status call. A pending document/package can no longer be edited. The only modifications that can still be done are extend the expiration date, and remove allowed signing methods through the API. A pending document/package cannot be deleted.
	Portal API	In Progress	Only applies when using Asynchronous Signing methods. A Package / Document is *In Progress* when the signer has finalized the signing process but the back-end has not yet finalized the document.
	Portal API	Ending	A Package / Document is *Ending* when the initiator requested to End the process but the back-end has not yet finalized the document.
	Portal API	Finished	A Package / Document is *Finished* when all actors performed the necessary actions or when the initiator ended the flow. '*Finished*' is a final state. Consequently, a finished document/package can be deleted. *Finished* documents/packages can also be downloaded.
	Portal API	Expired	A document/package goes to expired status when at least one actordid not act it before the expiration date. An *expired* document/package cannot be deleted. To delete an *Expired* document/package, it should first be revoked To make an expired document/package available again, its expiration date can be extended through the API.
	Portal API	Failed	A document/package goes to *failed* status when an asynchronous operation has failed and left a message on the Poison Queue, from which they can be resubmitted for processing.
	Portal API	Revoked	A document/package goes to *revoked* status when it has been revoked by the initiator. A *revoked* document/package can no longer be acted upon. To make it available again, the Set Package Status must be used.

OBJECT	PORTAL / API	STATUS	DEFINITION
	Portal API	Rejected	A document/package goes to *rejected* status when at least one signer rejected to sign it. It is no longer available for signing to the other signers. When a actor rejects to sign / approve, the reject reason is mandatory. The reject reason can be checked in the Document Portal and via the Get Package Status call in the API. 'Rejected' is a final state. Consequently the document/package can be deleted
Actor	API	Draft	An Actor is in status *Draft* if the Document / Package to which the Actor is linked is in status *Draft*
	API	Waiting	An Actor is in status *Waiting* if the Document / Package to which the Actor is *Pending* but the Actor is not the next to take an action. Examples are: the Actor is a signer and there is an approver that needs to approve first, or sequential workflow is being used and the actor is not the next to take action
	API	Available	An Actor is in status *Waiting* if the Document / Package to which the Actor is *Pending* and the Actor is the next to take an action.
	API	Finished	An Actor is in status *Finished* if the Document / Package when the actor has finalized his action. **Note:** that the Document / Package could still be pending is other actors still need to take action
	API	Rejected	An Actor is in status *Rejected* if the actor rejected the action that needed to be taken.
	API	Skipped	An Actor is in status *Skipped* if the initiator ended the flow prior to the actor taking action
	API	In Progress	Only applies when the Actor is a signer and is using an asynchronous signing method. An actor is *In Progress* when the actor has finalized the signing process but the back-end has not yet finalized the document.
	API	Failed	An actor goes to *failed* status when an asynchronous operation has failed and left a message on the Poison Queue, from which they can be resubmitted for processing.
Field	API	Pending	A Field is in status *Pending* when the Document / Package is in status *Pending* and no action has been taken on that field
		Rejected	An Field is in status *Rejected* if the actor rejected the action that needed to be taken.
		Refused	An Field is in status *Refused* if the actor refused the action that needed to be taken.
		In Progress	Only applies when the Field is a signing field and is using an asynchronous signing method. An Field is *In Progress* when the actor has finalized the signing process but the back-end has not yet finalized the document.
		Failed	A Field goes to *failed* status when an asynchronous operation has failed and left a message on the Poison Queue, from which they can be resubmitted for processing.
		Finished	A field is in status *Finished* if the Document / Package when the actor has finalized his action. **Note:** that the Document / Package could still be pending if other Fields are still pending.

Synchronous signing

Definition

Synchronous signing is the process of signing documents by which the signer needs to wait until eSignatures is finished with signing the document.

Signing methods

Synchronous signing is mandatory for the following signing methods: BeID, ManualBeID, BeLawyer, Itsme.

Tenant user

Definition

A *tenant user* is currently a user who has access to the **Theme** settings of the Config Index. The **Theme** settings determine the look and feel of eSignatures.

User Types

The following user types are available in eSignatures:

- User
- API user
- Tenant user
- Administrator

Theme

Definition

A *theme* is a collection of branding settings that determine the look and feel of eSignatures. A theme can be used to make eSignatures fit your corporate brand.

Creation

Tenant users can create themes in the **Theme** section of the Config Index.

Application

Themes can be applied on 3 different levels:

1. Environment level
2. Document group level
3. Package level

Environment level

A theme applied on environment level determines the entire look and feel of the eSignatures environment. Every user and signer will see the same branding.

Document group level

A theme applied on document group level determines the branding of the WYSIWYS for all documents that are uploaded to that document group.

Package level

A theme applied on package level determines the branding of the WYSIWYS for an individual package.

Unique identifier

Definition

A *unique identifier* uniquely identifies a signer based on their national security number or LawyerID.

Prerequisites

- eSignatures 5.3.0 or higher

More information

- A unique identifier can be added to a contact when creating or editing the contact in the WebPortal or through the **MandatedSignerIds** parameter on Actor level in the API.
- A unique identifier is only required if you want to apply Mandated Signer Validation based on the **matchId** parameter to a signer who uses eID or BeLawyer signing.

User

Definition

A *user* is a person who has access to the eSignatures Web Portal to send documents for signing. A user who uploads documents to the Web Portal is also called an initiator.

User Types

The following user types are available in eSignatures:

- User
- API user
- Tenant user
- Administrator

More information

- A user is not to be confused with a signer or a contact. Click the links to consult the respective keys.
- When an admin enables the **IsUserRegistrationEnabled** parameter in the Config Index, people can register themselves as users of eSignatures.

User Management

Definition

The *User Management* section is the part of the eSignatures WebPortal where admins can manage the eSignatures users and their permissions.

More information

For more information, see User Management in the User Documentation.

WebPortal

Definition

The *WebPortal* is the user interface of Connective's digital signature solution "**eSignatures**".

The WebPortal encompasses the Document Portal, Signer Portal and WYSIWYS.

Prerequisites

To access the WebPortal, a user needs the following elements:

- An eSignatures account
- Valid credentials to the account

Worker

Definition

The *Worker* is short for the Connective WebSigner Worker.

The Worker is the eSignatures component that enables the following actions:

- **Asynchronous signing**

  This allows signers to close the signing session and continue with other things.

- **Asynchronous prerendering of documents**

  This allows for a faster signing time, since the signer no longer needs to wait for the document to load after they've clicked Start signing.

WYSIWYS

Definition

WYSIWYS stands for "What You See Is What You Sign".

The WYSIWYS is the part of the eSignatures WebPortal that is opened when a signer clicks the link they receive via email, or when a user clicks the **Sign** button in the Document Portal, Signer Portal or Connective app.

More information

Fore more information, see Signing step-by-step (End users) in the User Documentation.

XML signing

Definition

XML signing is the process of digitally signing .xml documents sent through the API.

Prerequisites

- eSignatures 5.2.0 or higher
- The admin must enable the **IsXmlSigningEnabled** parameter under **Signing Format Settings** in the Config Index.

Limitations

- XML signing is not available in the WebPortal.
- XML files must not contain more than 2 million characters per file. A package must not contain more than 15 .xml files.
- Packages currently can't contain both XML documents and PDF documents on which signatures will be placed. The type of package is determined by the first uploaded document.