Agreements PDF handling

2021-10-26

REVISIONS

- 2021-10-04: Initial version.
- 2021-10-26: Add a section about digital signatures.
- 2022-03-11: Reviewed text for release

OVERVIEW

This document provides PDF specific documentation for the [Tenant integration: Agreements](#) functionality provided by the [Tenant Management API](#).

It is directed at tenants and integrators as well as those supporting them.

DOCUMENT LIFECYCLE

The lifecycle of an agreement document is:

1. An agreement document is uploaded.
2. The document is validated.
3. If the document passes validation a preprocessed document is produced.
4. This preprocessed document is shown to signers.
5. Once the document has been signed, a final signed covenant document is produced.

The following sections detail these steps.

PDF RECOMMENDATIONS

The PDF file format is flexible and complex, which may lead to surprising results when PDF documents are manipulated and later viewed in different readers.

To ensure a consistent experience across PDF readers (which is an important quality for an agreement document), it is advisable to check the document to be signed, or the document template used to generate documents, by opening it in various PDF readers and verify that they render identically. A suggested list of readers are:

- Adobe Acrobat Reader

- Google Chrome
- Pdf.js - this browser based PDF reader is used to show agreement documents to signers. An [example viewer](#) can be used, where a new document can be uploaded using the buttons in the top right corner.
- Preview (macOS built in PDF reader)

Finally, before starting to use a new document type in production, the [Sandbox environment](#) can be used to check how agreement documents are shown to the user and how the final signed covenant document looks.

Besides the above visual checks, there are constructions allowed by the PDF file format which are unsuitable for agreements:

- **Editable form values**: An editable form value can be changed by mistake in an opened document. This may mislead a signer, who may believe that the locally updated form values reflect the signed agreement. If the agreement document is the result of a filled form, the document to be signed should be "flattened" before signing, thereby making the document read-only. The Tenants Agreement API will do that automatically for uploaded documents.
- **Annotations**: PDF annotations are markup such as review comments, highlights or the above mentioned user data entered into interactive form fields. Any annotations that remain after flattening will be removed by the Tenants Agreement API to avoid the agreement from being rendered differently by different readers.
- **JavaScript**: While PDF documents in general support dynamic functionality using JavaScript it should be avoided in agreement documents. Besides making the document dynamic, the level of JavaScript support varies between PDF readers.
- **Multimedia and 3D models**: The support of these are limited across PDF readers.
- **Non-standard, non-embedded fonts**: A document may look different depending on what fonts the viewing user has installed if a non-standard font is not embedded in it.
- **Links to or embedding of external resources**: An agreement document should be self-contained and not depend on external resources which may change over time.
- **Attachments**: Support for attachments varies across PDF readers and many users do not know that PDF documents can contain attachments. If an attachment needs to be used to sign information which cannot be expressed in the document text itself, the existence of the attachment needs to be clearly expressed in the document text, preferably with an embedded link to the attachment.

PDF VALIDATION

When an agreement PDF is posted to the [Create Tenant](#) endpoint, the first thing happening is that the file is validated - it is checked that the file can be parsed as a PDF document and that it is not password protected.

If this is not the case, due to the file being broken (or not a PDF file at all), a HTTP 400 Bad Request response with an associated error code will be obtained.

If this error occurs the file needs to be checked for validity by the supplier of the PDF agreement file. This can be done in several ways:

- Open the file in Adobe Acrobat Reader and check for any error messages detailing the error.
- Upload the file at an online PDF validation tool such as the [pdf-online.com validator](#) and study the error messages displayed there.
- Check for rendering errors in various PDF readers.

Contact Kivra support if none of the above checks explains the validation error from the Create Tenant API call.

PREPROCESSING

Once the PDF document passes the validation described above, the file is preprocessed. This preprocessing step is done to

1. Correct common PDF file format problems by automatically repairing them.
2. Make form fields read-only by flattening them into the document.
3. Remove annotations which may cause the document to look different depending on the PDF reader that is used to view them.
4. Ensure that the document being shown to signers renders identically as the final produced covenant document.

While the exact byte content of the preprocessed document will differ from the originally uploaded document, it will look identical except for possibly removed annotations.

THE COVENANT DOCUMENT

Once every signing party has signed the document a final signed covenant document is produced.

This covenant document is constructed by applying the following changes to the signed agreement:

- A footer that indicates that the agreement has been signed is added to each page.
- The content of the agreement is scaled down to make place for that footer.
- An appendix with information about the signing parties is added.
- The agreement document is added as a PDF attachment, named original.pdf.
- A json metadata file, that contains machine readable information about the BankID signatures, is added as a PDF attachment, named verification.json.
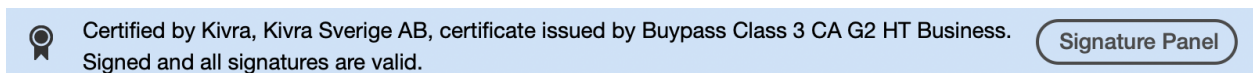- Finally, the document is signed with a digital signature (see the following section).

If there are non-trivial visual differences between the final covenant document and the original document, check that you are following the PDF recommendations in this document. Contact Kivra support if these visual differences still remain after ensuring that the original PDF document is sound.

PAdES DIGITAL SIGNATURES

The covenant document is digitally signed according to the PAdES (Advanced Electronic Signatures) specification in order to prove three aspects:

1. Authenticity: That the covenant document was indeed created by Kivra Signatures
2. Integrity: That the covenant document has not been altered
3. Timestamp: That the covenant was signed and valid at a specific time

Users can open the document in Acrobat Reader in order to check the authenticity and integrity of a document that claims to come from Kivra Signatures. If the document indeed was created by Kivra Signatures (after introduction of digital signatures), a box at the top of the document will be shown that proves that the document was indeed created by Kivra Signatures and not modified afterwards:

> ⚜ Certified by Kivra, Kivra Sverige AB, certificate issued by Buypass Class 3 CA G2 HT Business. Signed and all signatures are valid. 　　Signature Panel

The authenticity of documents can also be verified programmatically if desired.

Digital signatures provide a strong enough guarantee that it sometimes, depending on the context, can replace handwritten signatures in official documents such as annual reports for companies. Contact us at Kivra for more information regarding the legal validity of signatures.

PAdES is a standard for digital signatures adopted by the European eIDAS standard, meaning that agreements signed according to PAdES are legally binding in all EU member states since July 2014. For more information on PAdES: Wikipedia article or EU e-Signature page.

NOTE: Digital signatures went live 2022-03-15. For a document created before this date, contact Kivra to have the authenticity validated.