



UWP



Import and export



WINDOWS > GUIDES > ANNOTATIONS > IMPORT AND EXPORT

Import and export annotations from Instant JSON in UWP

Instant JSON is our approach to bringing annotations and bookmarks into a modern format while keeping all important properties to make the Instant JSON spec work with PDF. It's fully documented and supports long-term storage.

Instant JSON stores PDF changes like annotations and bookmarks in a separate JSON file. This means that a PDF document will only need to be transferred once and all changes will be added as an overlay to the existing PDF. This approach significantly reduces the bandwidth since you only need to transfer the JSON instead of the complete PDF.

Conceptually, Instant JSON defines a list of `skippedPdfObjectIds`. These point to the PDF's internal object IDs for annotations. Whenever an object ID is marked as *skipped*, it'll no longer be loaded from the original PDF. Instead, it could be defined inside the `annotations` array with the same `pdfObjectId`. If this is the case, the PDF viewer will display the new annotation, which signals an update to the original one. If an object ID is marked as *skipped* but the `annotations` array doesn't contain an annotation with the same `pdfObjectId`, it'll be interpreted as a *deleted* annotation. An annotation inside the `annotations` array without the `pdfObjectId` property is interpreted as a newly created annotation.

All annotations in the `annotations` array have a unique `id` field. For updated annotations that were in the original PDF, this field will be the stringified `pdfObjectId`. Newly created annotations will get a newly generated ULID.

An "empty" Instant JSON contains neither `skippedPdfObjectIds` nor `annotations`, which means the original PDF is untouched. All annotations in the initial PDF are still shown.

Instant JSON also defines a list of `skippedPdfBookmarkIds` that follow the same principals of `skippedPdfObjectIds`, but instead they contain a list of bookmarks that no longer need to be



ASK AI

from the original document.

The format

We use Flow type declarations to specify the format of Instant JSON:

```
1 declare type InstantJSON = {
2   format: "https://pspdfkit.com/instant-json/v1",
3   pdfId?: {
4     permanent: string,
5     changing: string
6   },
7   skippedPdfObjectIds?: number[],
8   annotations?: Object[],
9   bookmarks?: Object[],
10  skippedPdfBookmarkIds?: string[],
11  formFieldValues?: Object[]
12 };
```

format

This is a literal string that includes the version information.

pdfId

This optional key contains an object of a `permanent` PDF ID and a `changing` PDF ID. According to the PDF spec, a PDF document must contain these IDs. We use the `permanent` ID to verify that the PDF you've opened together with this Instant JSON is indeed the correct one. The `changing` PDF ID will be updated whenever the PDF file is updated (for example, when saved with different annotations). Since Instant JSON only works with an immutable PDF, the state will be invalid when used with a changed PDF.

Not every PDF will have a valid `permanent` or `changing` ID. As such, this field might not be set. We recommend you take care to always use the same PDF.

skippedPdfObjectIds

This is an array of PDF object IDs that will be ignored when importing annotations from the original PDF document. If this array is empty, the key shouldn't be set.

annotations

This is a list of new or updated annotations. Annotations follow the format for Instant JSON for annotations. When an annotation contains a `pdfObjectId`, it's considered to be an update to one of the annotations of the original PDF. For newly created annotations, this key won't be set.

bookmarks

This is a list that contains any new bookmarks added to the PDF. Bookmarks follow the format for Instant JSON for bookmarks.

skippedPdfBookmarkIds

This is an array of bookmark object IDs from the PDF that will be ignored when importing annotations from the original PDF document. If this array is empty, the key shouldn't be set.

formFieldValues

This is a list of modified form field values. Objects follow the [Instant form field value JSON format](#). This list won't be preset when no form field values have been modified.

Instant annotation JSON API

Annotation JSON is Instant's representation of a single annotation. This can be generated using the

`PSPDFKit.Pdf.Annotation.ToJson` method available in each annotation:

```
1 var textAnnot ... // Get the annotation from the document or from another
2 var textAnnotJson = textAnnot.ToJson();
```



For creating an annotation from the JSON, use `PSPDFKit.Pdf.Annotation.Factory.FromJson`, like so:

```
1 var fromJsonAnnotation = Factory.FromJson(textAnnotJson);
2
3 // Create the annotation parsed from JSON.
4 await pdfView.Document.CreateAnnotationAsync(fromJsonAnnotation);
```



There are some limitations with Instant JSON, in that not *all* annotation types are currently supported, and only the properties that can be handled correctly across all of Nutrient’s supported platforms (iOS, Android, and Web) are serialized. For more information, check out the detailed [JSON Format](#) guide.



There’s currently no dedicated API like the Instant Annotation JSON API for importing and exporting bookmarks alone. Use the Instant Document JSON API above for this purpose.

Was this helpful?



Questions? [Contact us](#)

