

iText DITO 1.4

Release Preview Guide

July 2020

Disclaimer - This Release Preview Guide contains highlights that are expected to be included in the next minor release of iText DITO. It is not meant to be complete, nor does it guarantee that all of the listed items will be fully implemented.

Introduction

We are excited to announce the general availability of iText DITO 1.4. This latest release of iText's high-convenience PDF document generator contains several new template design features and extended API capabilities.

Barcodes

The hero feature of the 1.4 release is barcode generation. The new version of iText DITO can automatically generate two-dimensional and three-dimensional codes based on plain text or number values in the JSON payload.

Conditional Formatting of Alphanumeric Data Bindings

The previous release saw the introduction of conditional content in iText DITO. This release adds conditional text formatting. This allows you to style data bindings depending on their value. So, if you want to emphasize negative numbers for instances by printing them bold, red and underlined, iText DITO can do that for you.

Formatting Preservation

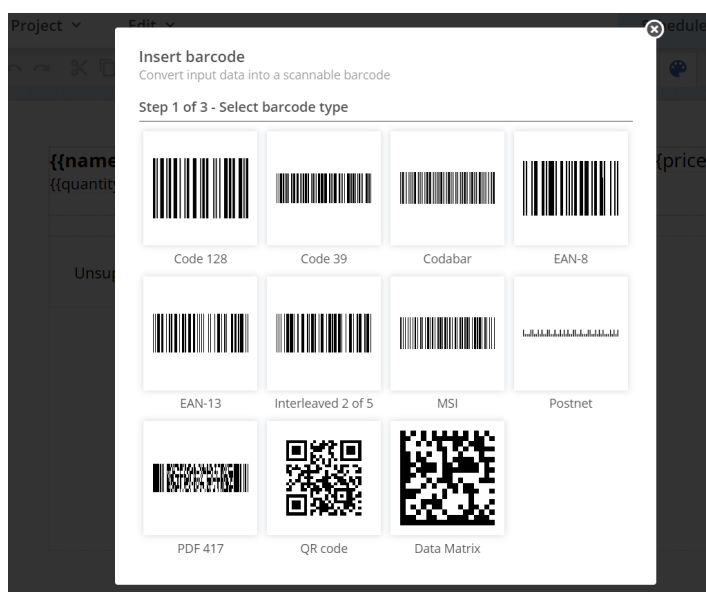
There's more text formatting news in this release. iText DITO 1.4 can read and apply HTML tags that are contained in the JSON payload. Although the list of supported tags is still limited, this new feature allows you to pass formatting tags along with the actual data that will be merged.

Password Protection

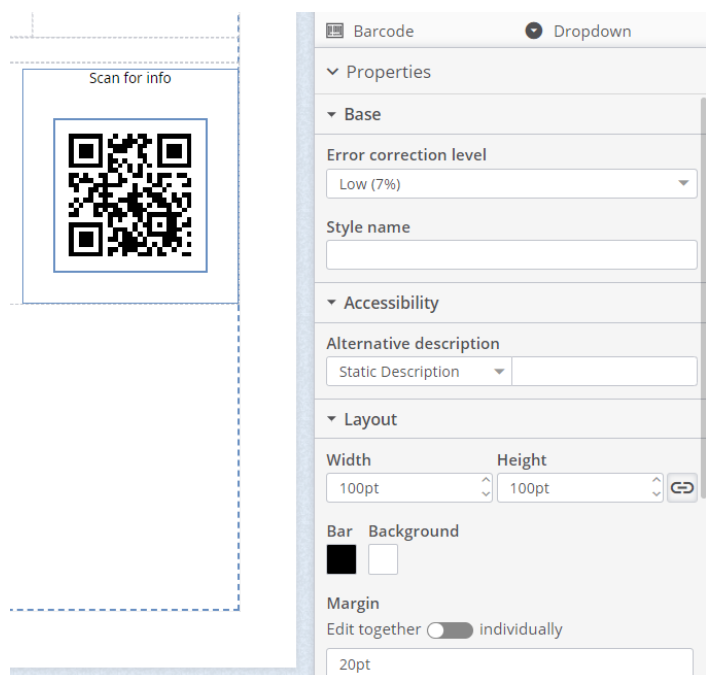
On the API side, iText DITO 1.4 can protect generated PDFs with a password. The password can protect the file either from being opened or edited. This new feature is available for both the Java and the Docker API.

Barcodes

iText DITO 1.4 can generate two-dimensional and three-dimensional barcodes from input data that is passed through the JSON payload. A wizard walks you through three steps to insert such a scannable barcode in your template. In the first step you select the type of barcode you would like to insert. In the second step you bind the barcode to a field in the data collection. In the third step you can set detailed configurations for the barcode.



The barcode elements you insert can be edited after insertion. If you want to, you can still switch to another type for instance. In terms of size and position, barcodes behave like images. So you can freely move them around or resize them according to your design requirements.



In this first iteration of the feature, iText DITO supports Code 128, Code 39, Codabar, EAN-8, EAN-13, Interleaved 2 of 5, MSI, Postnet, Planet, PDF 417, QR code and Data matrix. For all of these, type-specific configurations can be set by the template designer.

If something goes wrong during the generation of a barcode (e.g. when invalid data is passed through the JSON payload) a warning will be thrown at the API level. These warnings are visible in the PDF Production Result parameter.

Conditional Formatting of Alphanumeric Data Bindings

In iText DITO 1.4 you can format textual data bindings conditionally. This means that depending on a condition, or even a series of conditions, you can swap fonts, increase or decrease font size, character or word spacing, font color, font face and decorations.

Transaction	Transaction Date	Net Transaction Amount
PreApproved Payment Bill User Payment	1/6/2019	€ -29
General Currency Conversion	1/7/2019	€ -405,13
Account Hold for Open Authorization	1/10/2019	€ -21,40
Reversal of General Account Hold	1/10/2019	€ 21,40
PreApproved Payment Bill User Payment	1/10/2019	€ -21,40
Express Checkout Payment	1/11/2019	€ -100
General Currency Conversion	1/14/2019	€ -693,49
PreApproved Payment Bill User Payment	1/15/2019	€ -39
PreApproved Payment Bill User Payment	1/19/2019	€ -9,99
General Currency Conversion	1/21/2019	€ -699,68
PreApproved Payment Bill User Payment	1/26/2019	€ -9,99
General Currency Conversion	1/28/2019	€ -703,10
Mass Pay Payment	1/1/2019	€ 52,13
Express Checkout Payment	1/4/2019	€ -27
PreApproved Payment Bill User Payment	1/7/2019	€ -470,93
General Currency Conversion	1/7/2019	€ 445,80
PreApproved Payment Bill User Payment	1/14/2019	€ -770,63
General Currency Conversion	1/14/2019	€ 770,63
PreApproved Payment Bill User Payment	1/21/2019	€ -770,63
General Currency Conversion	1/21/2019	€ 770,63
PreApproved Payment Bill User Payment	1/28/2019	€ -770,63
General Currency Conversion	1/28/2019	€ 770,63

The feature is built in such a way that conditional text formatting configurations can be saved as a reusable style. Imagine that you have a whole list of data bindings with numeric values spread across your template and you want all of the resulting numbers in the generated documents to turn red if they are negative. In that case, you can create a Red if negative style that can be reused easily. In this example the condition to be evaluated (negative or not) applies to the same value as the formatting, so in the condition wizard we can use the referential value `$this`.

Transaction Date

Conditional formatting

Apply different formatting based on conditions

Style name

Red if negative

Conditions

+

↑

↓

\$this < 0

Default

Formatting

Text style

Font

inherit

Size

inherit

Char Spacing

inherit

Word Spacing

inherit

B

I

U

A

Save

▼ Data

Data Bind

Net

☐ Preserve formatted text

Display Format

num('\$ #,###.#0','en-be')

Bind Format

☒ Apply conditional formatting

Apply style

Red if negative

+ Create new style

Conditional styles can be more complex than illustrated by the red if negative example. They can consist of multiple predicate-formatting combinations. A single style could for instance define that numbers below a certain threshold should be printed in red, while values above another threshold should be printed with an increased font size and with emphasis.

Transaction Data

Conditional formatting

Apply different formatting based on conditions

Style name

Red if negative, Emphasized|f >200

Conditions

+

\$this < 0

\$this >= 200

Default

Formatting

Text style

Font

inherit

Size

16

Char Spacing

inherit

Word Spacing

inherit

B

I

U

A

Save

Transaction	Transaction Date	Net Transaction Amount
PreApproved Payment Bill User Payment	1/6/2019	€ -29
General Currency Conversion	1/7/2019	€ -405,13
Account Hold for Open Authorization	1/10/2019	€ -21,40
Reversal of General Account Hold	1/10/2019	€ 21,40
PreApproved Payment Bill User Payment	1/10/2019	€ -21,40
Express Checkout Payment	1/11/2019	€ -100
General Currency Conversion	1/14/2019	€ -693,49
PreApproved Payment Bill User Payment	1/15/2019	€ -39
PreApproved Payment Bill User Payment	1/19/2019	€ -9,99
General Currency Conversion	1/21/2019	€ -699,68
PreApproved Payment Bill User Payment	1/26/2019	€ -9,99
General Currency Conversion	1/28/2019	€ -703,10
Mass Pay Payment	1/1/2019	€ 52,13
Express Checkout Payment	1/4/2019	€ -27
PreApproved Payment Bill User Payment	1/7/2019	€ -470,93
General Currency Conversion	1/7/2019	€ 445,80
PreApproved Payment Bill User Payment	1/14/2019	€ -770,63
General Currency Conversion	1/14/2019	€ 770,63
PreApproved Payment Bill User Payment	1/21/2019	€ -770,63
General Currency Conversion	1/21/2019	€ 770,63
PreApproved Payment Bill User Payment	1/28/2019	€ -770,63
General Currency Conversion	1/28/2019	€ 770,63

Conditional styles are not limited to number-based conditions. You can also use textual or boolean-based conditions to define the style of a data binding. Consider a scenario for instance where the renewal date of a contract needs to be printed in bold if a true-false field that indicates whether renewal payment has been received, is false. Note that in this example the field that is evaluated in the condition is not the field that is conditionally formatted, so instead of using the referential `$this`, we selected a field (`$.renewal_payment_received`) from the tree.

Conditional formatting

Apply different formatting based on conditions

Style name

Bold for unpaid renewals

Conditions

+ ↑ ↓
 {\$.renewal_payment_received} [edit] [delete]
 Default

Formatting

Text style

Font inherit Size inherit **B** *I* U A

Char Spacing

inherit

Word Spacing

inherit

Save

Condition Settings

Define a condition that will display or format the selected content

Wizard ☐ Syntax

Field

\$.renewal_payment_received

Use as

True/False (Boolean)

Operator

Is equal to

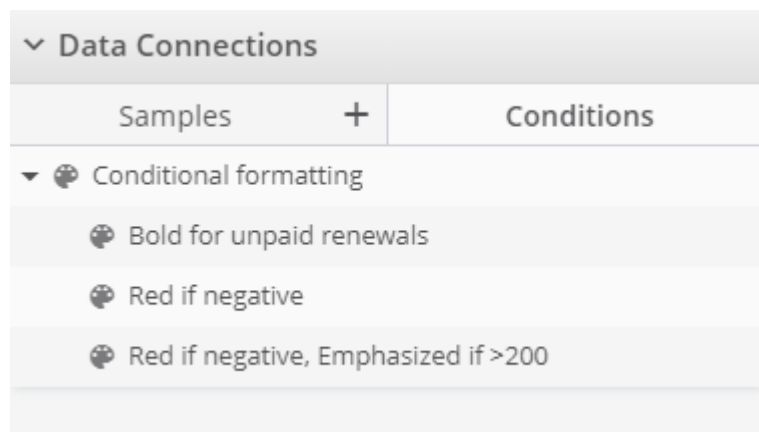
Value

False

Back

Save

In data mapping mode, you can manage the conditional formatting styles in the Conditions pane of the Data Connections panel.



Text Formatting Preservation

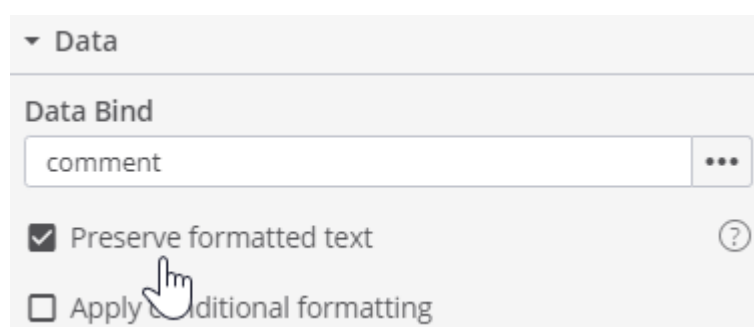
The iText DITO Editor has a broad spectrum of text formatting capabilities, but sometimes the data that you pass to the PDF producer service may already contain formatting tags. Typically rich text fields in the user interfaces of your systems of record, may store content along with layout in HTML format. iText DITO 1.4 can read a selection of HTML layout tags and apply them. Supported tags in the first iteration of this feature are:

-
- <i>
- <u>
-
-

Supported styles are:

- font-family
- font-weight
- font-style
- font-size
- color

Unsupported tags and tags with unsupported style declarations will be replaced with an empty string. All attributes except style will be ignored and removed from the tag. To activate formatting preservation, you tick the box in the data bind configuration. If you leave the box unticked all tags will be considered flat text and printed.



Password Protection for Generated PDFs

In iText DITO 1.4 it is possible to protect generated PDFs with a password. Password protection configurations are handled at the API level. You can set a user password (prompting the user to fill out a password to open the document), an owner password (prompting the user to fill out a password to interact with the document), or both, and set the appropriate protection levels through the newly added Encryption Properties of the PDF Producer operation. The permission parameters you can configure are:

- Content copying;
- Editing;
- Printing;
- Screen reader access.

Minor Improvements

iText DITO 1.4 also introduces a number of smaller improvements:

- Broadening of the pdfProductionResult operation to throw warnings for a variety of template-related issues and visualization of such warnings in preview mode;
- Extended browser support for the iText DITO Editor: the 1.4 Editor application supports Microsoft Edge 79.0 and higher;
- Support for boolean values in conditional content and conditional formatting wizards;
- Extension of the conditional formatting options with strikethrough text decoration;
- Widow and orphan control through CSS.