

# Exercise 4 - How to Populate a Word Template with a Table of Contents

In this exercise, you'll learn how to populate a Word template with a table of contents.

Open an empty word document, let's get to the exercise.


Firstly, we are going to create a cover page, you need to go to tab *Insert* -> *Cover Page* and select one of your choice.

 The Cover Page is populated from the document properties. You can:

1. Write your desired text directly on the cover page;
2. Insert it into the correct document property or;
3. Apply a mapping to the document property.

You'll see examples of this in this exercise example template.

After creating the cover page, the second page is blank. We are going to insert a table of contents. you need to go to tab *References* -> *Table of Contents* select the *Automatic Table 1*.

 The table of contents is going to three levels:

The first level is detected by Heading 1

The second level is detected by Heading 2

The third level is detected by Heading 3

Our template will export a **list of Bugs**, and then a **list of Stories** (Issue types). These will be our **first level** on the TOC (Table of Contents).

For each issue exported, in both issue types, it will be printed its **Key**, which will be the TOCs **second level**, and its **comments** and **linked issues**, which will be the TOCs **third level**.

Below the Table of Contents, we are going to insert a page break and continue with our template.

Now we have to insert a title. Name it "**Bugs**" and set up the style **Heading 1**. Below it, we are going to create the following section that will list all the issues being exported of the type Bug.

```
&{for issues|filter=%{"${IssueTypeName}'.equals('Bug')"/>
```

We'll insert a new title here to identify each issue in the TOC. Name it "**\${Key}**" and set the style **Heading 2**.

We'll want to extract the issue summary, so we need to put on the template, the correspondent mapping.

Now, for each issue, we'll want to export its linked issues. However, we'll only want to print this information in the final document if there are issue links. Insert the following conditional block:

```
#{if (%{"${LinksCount} > 0})}
```

The title for this iteration can be "**Links**" and set up the style **Heading 3**.

Insert the Links table Header:

Key	Summary	Link Type
-----	---------	-----------

And below it you put the **&{for links}** or **&{for <VariableName>=LinksCount}** statement, so the Header is printed only one time.

With that done, you create another table where the Issue Links will be populated:

\${Links[n].Key}	\${Links[n].Summary}	\${Links[n].LinkType}
------------------	----------------------	-----------------------

Finally close the Links statement using the mapping **&{end}** and the Conditional Block statement with another **&{end}**.

Now, we are going to repeat the same process to iterate the issue comments.

Insert the conditional block to know if there are comments on the issue

```
#{if (%{"${CommentsCount} > 0})}
```

The title for the second iteration can be "Comments" and set up the style **Heading 3**.

Then the Links table Header:

Author	Body	Date
--------	------	------

And below it you put the **#{for comments}** or **#{for <VariableName>=CommentsCount}** statement, so the Header is printed only one time.

With that done, you create another table where the Issue Comments will be populated:

<code>\${Comments[n].Author}</code>	<code>\${Comments[n].Body}</code>	<code>\${dateformat("yyyy-MM-dd hh:mm"):Comments[n].Created}</code>
-------------------------------------	-----------------------------------	---

Finally close the Links statement using the mapping **#{end}** and the Conditional Block statement with another **#{end}**.

We must finish the section of Bugs with **&{end}**.

We need to repeat the process above, but now for the stories. Start by inserting a page break.

Insert a title named "Stories" and set up the style **Heading 1**. Below it, We are going to create the following section will list the issues of the type Story.

**&{for issues|filter=%{'\${IssueTypeName}'.equals('Story')}}**

Insert the second level of the TOC with the issue **"\${Key}"** and extract the issue Summary on the next line of the document.

Repeating the previous steps to iterate the issue links and comments, insert:

**#{if (%{'\${LinksCount}' > 0})}**

The title for this iteration should be "Links" and set up the style **Heading 3**.

Insert the Links table Header:

Key	Summary	Link Type
-----	---------	-----------

And the **#{for links}** or **#{for <VariableName>=LinksCount}** statement. Insert the table to export the links:

<code>\${Links[n].Key}</code>	<code>\${Links[n].Summary}</code>	<code>\${Links[n].LinkType}</code>
-------------------------------	-----------------------------------	------------------------------------

Close the iteration with **#{end}** and the conditional block with another **#{end}**.

Insert the comments sections:

**#{if (%{'\${CommentsCount}' > 0})}**

The title for this iteration should be "Comments" and set up the style **Heading 3**. Insert the Comments table Header:

Author	Body	Date
--------	------	------

And the **#{for comments}** or **#{for <VariableName>=CommentsCount}** statement. Insert the table to export the comments:

<code>\${Comments[n].Author}</code>	<code>\${Comments[n].Body}</code>	<code>\${dateformat("yyyy-MM-dd hh:mm"):Comments[n].Created}</code>
-------------------------------------	-----------------------------------	---

Close the iteration with **#{end}** and the conditional block with another **#{end}**.

Finish the section of Stories with **&{end}**.

We have completed our template. However, you need to go to the table of contents. The table is empty, so we must do these steps:

1. Select the table of contents
2. Right click and click on Updated Field
3. Update entire table
4. Click on OK.



The table of contents will show us the three levels that you have created along of template.



We can create more levels using different styles.

Also, you can iterate more iterations for example(Attachments, worklogs, etc..)

Below there is a sample of how the mappings will be displayed in a Word template:

<div><div></div><div><hr/><div><div>#{PROJECTNAME}</div></div><hr/></div><div>Bugs and Stories</div><div></div></div>	<div>Contents</div> <div>Bugs.....2</div> <div>#{Key} .....2</div> <div>Links .....2</div> <div>Comments .....2</div> <div>Stories .....3</div> <div>#{Key} .....3</div> <div>Links .....3</div> <div>Comments .....3</div>
<div>#{DATEFORMAT("YYYY-MM-DD HH:MM"):CURRENTDATE}</div> <div>XPORTER ACADEMY</div>	

## Bugs

&{for issues|filter=%('\${IssueTypeName}'.equals('Bug'))}

\${Key}

\${Summary}

#{if (%\${LinksCount} > 0)}

Links

Key	Summary	Link Type
-----	---------	-----------

#{for links}

\${Links[n].Key}	\${Links[n].Summary}	\${Links[n].LinkType}
------------------	----------------------	-----------------------

#{end}

#{end}

#{if (%\${CommentsCount} > 0)}

Comments

Author	Body	Date
--------	------	------

#{for comments}

\${fullname:Comments[n].Author}	\${Comments[n].Body}	\${dateformat("yyyy-MM-dd hh:mm");Comments[n].CreatedDate}
---------------------------------	----------------------	--

#{end}

#{end}

&{end}

## Stories

&{for issues|filter=%('\${IssueTypeName}'.equals('Story'))}

\${Key}

\${Summary}

#{if (%\${LinksCount} > 0)}

Links

Key	Summary	Link Type
-----	---------	-----------

#{for links}

\${Links[n].Key}	\${Links[n].Summary}	\${Links[n].LinkType}
------------------	----------------------	-----------------------

#{end}

#{end}

#{if (%\${CommentsCount} > 0)}

Comments

Author	Body	Date
--------	------	------

#{for comments}

\${fullname:Comments[n].Author}	\${Comments[n].Body}	\${dateformat("yyyy-MM-dd hh:mm");Comments[n].CreatedDate}
---------------------------------	----------------------	--

#{end}

#{end}

&{end}

This Template has:

- A header with an image
- Headings
- Text styling
- Table
- Table of Contents
- Cover Page

Below there is a sample of how the generated file will be populated:



# XPORTER ACADEMY

Bugs and Stories



2019-12-09 09:51  
XPORTER ACADEMY

## Contents

Bugs.....	2
XA-4.....	2
Links .....	2
Comments .....	2
XA-3.....	2
Links .....	2
Comments .....	2
Stories .....	3
XA-7.....	3
XA-5.....	3
Links .....	3
XA-2.....	3
Links .....	3
XA-1.....	3
Links .....	3
Comments .....	3

## Bugs

### XA-4

Reports are being generated too quickly

#### Links

Key	Summary	Link Type
XA-2	Analyze the possibility of having a cool feature	blocks
XA-5	Analyze the possibility of having schedule reports	causes

#### Comments

Author	Body	Date
Mário Jorge	There <u>are</u> no solution for this. We aim to be fast.	2019-12-09 09:47

### XA-3

Some fields are too awesome to be retrieved

#### Links

Key	Summary	Link Type
XA-1	Create a service to request template	blocks
XA-2	Analyze the possibility of having a cool feature	blocks

#### Comments

Author	Body	Date
Mário Jorge	This should be resolved as soon as possible.	2019-12-09 09:46

## Stories

### XA-7

Feature to be done on the epic

### XA-5

Analyze the possibility of having schedule reports

#### Links

Key	Summary	Link Type
XA-1	Create a service to request template	created by
XA-4	Reports are being generated too quickly	is caused by

### XA-2

Analyze the possibility of having a cool feature

#### Links

Key	Summary	Link Type
XA-3	Some fields are too awesome to be retrieved	is blocked by
XA-4	Reports are being generated too quickly	is blocked by

### XA-1

Create a service to request template

#### Links

Key	Summary	Link Type
XA-3	Some fields are too awesome to be retrieved	is blocked by
XA-5	Analyze the possibility of having schedule reports	created

#### Comments

Author	Body	Date
Rafael Videira	When will this feature be ready?	2019-12-05 03:29



If you like this exercise, please share your opinion on the page by just leaving a comment or a 👍. Your opinion is very important to us.

Thank you in advance. Enjoy our product. 😊

Here's the **Exercise 4** related files:

File	Description
<a href="#">Exercise 4 - Template</a>	<b>Exercise 4</b> - Sample template file
<a href="#">Exercise 4 - Generated</a>	<b>Exercise 4</b> - Sample generated file