



We enable docx-as-code

**Advanced Template version: 1.0**

[[doc-title]]

[[doc-subtitle]]

[[doc-type]]

[[doc-company]]

[[doc-company-url]]

Document Number: [[doc-id]]

Version: [[doc-version]]

Release Date: [[doc-release-date]]



We enable docx-as-code

## Document History

Author Name: [[doc-author-name]]  
Author Contact: [[doc-author-email]]

## Document Revision

Version Number	Release Date	Status	Change Description	Author
[[doc-version]]	[[doc-release-date]]	Draft	[[doc-change-description]]	[[doc-author-name]]

## Intellectual Property Rights

This document is exclusive property of [[doc-company]] and all rights are reserved.



## Table of Contents

<b>1 Introduction.....</b>	<b>4</b>
1.1 References.....	4
<b>2 markdown ekstenzija.....</b>	<b>5</b>
2.1 drugi heading.....	5
<b>3 UC2 Make Your Website Docs Available Offline.....</b>	<b>6</b>
<b>Table of Tables</b>	
Table 1: Document references.....	4

### **Table of Figures**



## 1 Introduction

### 1.1 References

Referenced documents in creation of this document are listed in the Table 1.

Document Reference	Document Title	Version	Filename
[1]	[[ reference-doc-title ]]		[[ reference-doc-filename ]]
[2]			
[3]			
[4]			
[5]			
[6]			
[7]			
[8]			

Table 1: Document references



We enable docx-as-code

## 2 markdown ekstenzija

### 2.1 drugi heading



### 3 UC2 Make Your Website Docs Available Offline

- ✓ *Do you use some of the Jamstack Static Site Generators (SSG) - Jekyll, Hugo, Next.js, Gatsby etc?*
- ✓ *Do you think it is cool to make your website content available offline?*

If answers are: yes, yes respectively, you are at the right place.

In this section we will explain the workflow on how you can keep using your favorite SSG and still make your website content available offline. The workflow is outlined in the following UML diagram:

1. Register (if not already done) and login to our application
2. Import a Git project where you have Markdown based documentation.
3. Our application will connect to your repo and import it.
4. Upload your DOCX corporate template. For more details how to do it please check [Create DOCX Template](#).
5. Save your new doc in our application and push it to your Git repo.
6. Our App will autogenerate the DOCX structure from the SSG repo
7. Our App will push to your Git project uploaded DOCX template.
8. Our App will push to your Git project created DOCX structure (Table of Content).
9. Our App will push to your Git project new updated DOCX and PDF documents.
10. [If Using Git for Updates][If Updating the Content] You can update the content and commit the change to your repo.
11. Your repo through webhook service will notify our App about the change.
12. Our App will autogenerate updated DOCX and PDF.
13. Our App will push to your Git project new updated DOCX and PDF documents.
14. [If Changing DOCX Template] You can update the DOCX Template and commit the change to your repo.
15. Your repo through webhook service will notify our App about the new Template.
16. Our App will autogenerate updated DOCX and PDF.
17. Our App will push to your Git project new updated DOCX and PDF documents.
18. [If Changing Table of Content] You can update the Content structure and commit the change to your repo. There are different possibilities: add/remove chapters/ .md docs, change order/heading level
19. Your repo through webhook service will notify our App about the new Table of Content.
20. Our App will autogenerate updated DOCX and PDF.
21. Our App will push to your Git project new updated DOCX and PDF documents.

If using PuzzlesCloud for Updates

This option is still in development, coming soon.



We enable docx-as-code



Figure 1: Last Page