



Zpider User Guide

Table of Contents

1. Getting Started	1
1.1. Installation	1
1.2. Running Zpider	1
2. Syntax	2
2.1. AsciiDoc Syntax	2
2.2. Zpider Additions	2
3. Levels	4
3.1. Definitions	4
3.2. Controlling Inclusion/Exclusion	4
3.3. Removing Empty Sections	4
3.4. Examples	5
4. IDs	6
4.1. ID Attributes	6
4.2. ID Filtering	6
4.3. Examples	6
4.4. ID Versioning	6
4.5. Examples	6
5. Tags	7
6. Usage	8
6.1. Commands	8
6.2. Common Options	9

1. Getting Started

1.1. Installation

Zpider is installed from the Zenterio PPA using

```
sudo apt-get install zenterio-zpider
```

The Zpider installation requires Docker to work and will install the docker-ce package.

*To get Docker to work the user needs to be a member of the **docker** group.*

```
sudo adduser <user> docker  
reboot
```

The group will take effect after reboot (sometimes a logout/login is enough)

1.2. Running Zpider

The easiest way to run Zpider is to

```
cd <documentation_root_directory>  
zpider pdf <name of .adoc file>
```

Example with how to generate a PDF from this User Guide

```
cd documentation/zpider  
zpider pdf --output-pdf user_guide.pdf user_guide.adoc
```

Currently it's only possible to store the output file inside the current working directory because it is generated inside a Docker container that doesn't have access to anything else.

2. Syntax

Zpider is based on the [AsciiDoc](#) markup language and it uses the [AsciiDoctor](#) tool to generate the documentation.

2.1. AsciiDoc Syntax

The [AsciiDoctor User Guide](#) contains description of the complete AsciiDoc syntax including AsciiDoctor specific functionality.

2.2. Zpider Additions

Zpider provides some added functionality through AsciiDoctor plugins.

2.2.1. Level Syntax

Levels are marked with the [AsciiDoctor Role](#) syntax.

Levels always begin with a dot (.).

*Example of how to make a title as a Level with the name **mylevel**.*

```
[.mylevel]  
== My Level Title
```

2.2.2. Tag Syntax

Tags are set using the **tags** attribute using the [AsciiDoctor Element Attributes](#) syntax. The tags should be explicitly quoted with a comma between each tag.

Example of how to set tags on a title.

```
[tags="tag1,tag2"]  
== My Tagged Title
```

2.2.3. ID Syntax

IDs are used to be able to create references to titles. This uses the builtin AsciiDoctor ID functionality. For more information see [AsciiDoctor Cross References](#)

IDs always begin with a hash (#).

It's recommended to use explicit ID assignment to be able to rename titles without affecting references in other places of the documentation.

Example of how to set an ID on a title.

```
[#my-id]  
== My Tagged Title
```

2.2.4. Combining Level, Tag and ID

It's important to combine the different types of metadata in the correct way.

The order should be:

1. IDs
2. levels
3. Tags

Example of how to combine multiple IDs, levels and tags.

```
[#my-id1#my-id2.my-level1.my-level2, tags="tag1,tag2"]  
== My Tagged Title
```

The comma before the tags marks the beginning of the attribute list. IDs and Tags should not have any separator between them.

3. Levels

Zpider adds the possibility to define document *levels* and to be able to specify that the document generation should include and/or exclude different types of content for a specific level.

Levels are controlled through the options [levels.ids](#), [levels.<level>.includes](#), [levels.<level>.excludes](#) and [magic.titles](#).

For each level, it's possible to control the inclusion and exclusion of the content types summary, other and magic titles.

3.1. Definitions

level

A **level** is defined with the [levels.ids](#) option.

level section

A section marked with one of the defined **levels**. See [Section 2.2.1, "Level Syntax"](#).

summary

A **summary** is the first paragraph of a **level section**.

magic title

A **magic title** is defined with the [magic.titles](#) option.

magic title section

A section with a title that matches one of the **magic titles**.

other

Any content that doesn't match any of the other content types.

all

Everything except **level sections**.

3.2. Controlling Inclusion/Exclusion

For each **level** the inclusion and exclusions are defined with the [levels.<level>.includes](#), [levels.<level>.excludes](#) options.

Rules

- Includes are by default **all**.
- Excludes override includes.

3.3. Removing Empty Sections

After excluding content, empty sections are removed by default. This can be controlled with the [keep-empty-sections](#) option.

3.4. Examples

Document with only summary on feature level

```
levels.ids: [feature]  
levels.feature.includes: [summary]
```

Document that excludes Customization title on feature level

```
magic.titles: [customatization]  
levels.ids: [feature]  
levels.feature.excludes: [customization]
```

4. IDs

Zpider supports filtering sections using ID attributes. The ID attributes can be any string.

4.1. ID Attributes

ID Attributes are configure on Sections using [asciidoc](#) attributes.

```
[.level,id="my-level-id"]  
== Section with ID
```

4.2. ID Filtering

Which IDs that should be included in the document can be configured with the [include.ids](#), [exclude.ids](#) options.

The following rules are used

1. By default everything is included.
2. Specifying includes implicitly excludes everything not explicitly included.
3. Explicit excludes override includes.
4. Includes will include parent sections unless they are explicitly excluded.

4.3. Examples

Document with some IDs included

```
include.ids: [feature1, feature2]
```

Document with some IDs excluded

```
exclude.ids: [feature1, feature2]
```

4.4. ID Versioning

IDs can be versioned. By default, only the latest version is included in the generated document. What versions to include can be controlled using the [max.allowed.version](#) and [keep.all.versions](#) options.

4.5. Examples

```
[.level,id="my-feature",version="1"]  
== Section with ID version 1  
  
[.level,id="my-feature",version="2"]  
== Section with ID version 2
```


5. Tags

Zpider will in future versions have the possibility to specify tags on titles and to control inclusion/exclusion of the titles depending on the tags.

Currently the only thing existing for tags is the specification of how to add them to the document. This is described in [Section 2.2.2, “Tag Syntax”](#).

6. Usage

6.1. Commands

6.1.1. HTML

Command that generates HTML output. The HTML file is entirely self-contained with all images etc. inlined using data URIs.

6.1.1.1. `output.html` (`--output-html`)

The output HTML file.

default

The name of the `adoc.file` argument but with `.html` instead of `.adoc`.

6.1.1.2. `adoc.file` (`ADOC_FILE`)

The adoc file to convert to HTML. On command line this is given as a positional argument.

6.1.2. PDF

Command that generates PDF output

6.1.2.1. `output.pdf` (`--output-pdf`)

The output PDF file.

default

The name of the `adoc.file` argument but with `.pdf` instead of `.adoc`.

6.1.2.2. `pdf.fonts` (`--pdf-fonts`)

Option that specifies which directory that includes fonts.

default

Internal zpider fonts.

6.1.2.3. `pdf.styles` (`--pdf-styles`)

Option that specify which style YAML file that should be used.

default

Internal zpider Zenterio style.

6.1.2.4. `adoc.file` (`ADOC_FILE`)

The adoc file to convert to PDF. On command line this is given as a positional argument.

6.1.3. YAML

Command that generates YAML output.

6.1.3.1. output.yaml (--output-yaml)

The output YAML file.

default

The name of the adoc.file argument but with .yaml instead of .adoc.

6.1.3.2. adoc.file (ADOC_FILE)

The adoc file to convert to YAML. On command line this is given as a positional argument.

6.2. Common Options

6.2.1. levels.ids (--levels-ids)

Specify what levels are included in the document structure.

6.2.2. levels.<level>.includes (--levels-<level>@includes)

Specify what should be included on the <level>.

6.2.3. levels.<level>.excludes (--levels-<level>@excludes)

Specify what should be excluded on the <level>. This overrides includes.

6.2.4. magic.titles (--magic-titles)

Titles that should be specially handled. These can be included and/or excluded by using the title name.

6.2.5. keep.empty.sections (--keep-empty-sections)

By default empty sections will be removed. By setting this config option to true, empty sections after the include/exclude step, will be kept.

6.2.6. include.ids (--include-ids)

Include these IDs.

6.2.7. exclude.ids (--exclude-ids)

Exclude these IDs.

6.2.8. max.allowed.version (--max-allowed-version)

If provided this is the highest version that will be kept in the generated document.

6.2.9. keep.all.versions (--keep-all-versions)

If set all versions of a section with a given ID will be included in the generated document. Otherwise, only the latest version is kept.