Table Of Contents

OVERVIEW

speccer is a CLI tool for dealing with specs.

It is a frontend to the speclib.

SCENARIOS

Create Specification

Mike wants to create a specification. He gets a clean starting point with `speccer`.

Add comments

Mike receives comments for his specification. He wants to add them without hassle.

See and update specification

Mike wants to update his specification with is editor.

Script

Gill wants to change specifications with a script.

Import and export

Gill wants to read import and export specifications from different places.

View as HTML and markdown

Sue wants to read the specifications in HTML and be able to mail them as markdown.

Filter for documentation

Peter wants to create a documentation from the spec and wants the unneccessary informations to be filtered.

Filter for implementation

Anne wants to implement parts of the specification. She just wants to read the parts that are ready to be implemented.

DEFINITIONS

All definitions of the `speclib` apply

-name

The parameter passed via `-name` is the name of a spec property (like `TRANSLATIONS` with the command `property`) or the name of a paragraph property (like `STATE` with the command `meta`).

-usage

The parameter passed via `-usage` is the name of a filter provided by `speccer` for typical usage scenarios.

There are currently:

- **documentation** filters
  - PROPERTIES
  - META
  - COMMENTS
  - NONGOALS
  - CONTRADICTIONS
  - UNDECIDED
  - SPEC_END
  - RESOURCES
  - PLANNING
  - APPROVED
  - PARTLY_IMPLEMENTED
  - OBSOLET
- **approval** filters
  - META
  - COMMENTS
  - CONTRADICTIONS
  - UNDECIDED
  - SPEC_END
  - RESOURCES
  - PLANNING
  - OBSOLET
- **discussion** filters
  - OVERVIEW
  - APPROVED
  - PARTLY_IMPLEMENTED
  - OBSOLET
  - SPEC_END
- **implementation** filters
  - UNDECIDED
  - PLANNING
  - FULLY_IMPLEMENTED
  - OBSOLET

FEATURES

create a new spec

Go into a folder that should contain your spec and run

```
speccer -cmd create
```

Creates a file `spec.json` in the current folder and exports it as `spec.md` (Markdown) and `spec.html` (HTML).

If the `-language` parameter is set (to e.g. `en_US`) it creates the following files instead: - `spec_en_US.json` - `spec_en_US.md` - `spec_en_US.html`

Be aware that the new spec is not valid since there are missing properties. You can s validate a spec to see them:

```
speccer -cmd validate
```

---

validate a spec

Go into a folder that contains your spec and run

```
speccer -cmd validate
```

It it returns nothing your spec is valid.

---

recreate HTML and markdown export based on current spec.json

Go into a folder that contains your spec and run

```
speccer -cmd save
```

---

export markdown to stdout without overwriting json.md

Go into a folder that contains your spec and run

```
speccer -cmd markdown
```

You might also pass the usage option to filter out some data, e.g.

```
speccer -cmd markdown -usage "documentation"
```

---

export HTML to stdout without overwriting json.html

Go into a folder that contains your spec and run

```
speccer -cmd html
```

You might also pass the usage option to filter out some data, e.g.

```
speccer -cmd html -usage "documentation"
```

---

print property of spec

Go into a folder that contains your spec and run

```
speccer -cmd property -name COMPANY
```

to print the COMPANY property. The properties are all written in high caps and defined in `speclib`.

---

set property of spec

Go into a folder that contains your spec and run

```
speccer -cmd property -name COMPANY -set acme
```

to set the `COMPANY` property to `acme`. The properties are all written in high caps and defined in `speclib`.

---

unset property of spec

Go into a folder that contains your spec and run

```
speccer -cmd property -name PARENT -uset
```

to unset the `PARENT` property. Only the following properties can be unset:

- REQUESTEDBY
- RELATED
- TRANSLATIONS
- SUPERSEDEDBY
- RESOURCES
- PARENT
- PERSONS

---

add markdown text to a section

Go into a folder that contains your spec and write your text into a file, let's say `new.md` and then run

```
speccer -cmd add -sec SCENARIOS -resp 'Your Name' -set new.md
```

to add the content of the file to the SCENARIOS sections. The available sections are defined in the `speclib`.

---

show list of paragraphs of a section

Go into a folder that contains your spec and run

```
speccer -cmd positions -sec SCENARIOS
```

to see a list of all paragraphs of the SCENARIOS section. Each paragraph has the first line printed preceeded by a number indicating the position of the paragraph inside the section. This position can be used to print or change the paragraph.

---

show content of a paragraphs of a section

Go into a folder that contains your spec. You will need to know the position of the paragraph you want to see. Therefor you might want to first print all positions of the section (here SCENARIOS) with

```
speccer -cmd positions -sec SCENARIOS
```

then you can run

```
speccer -cmd text -sec SCENARIOS -at 3
```

to see the text for the 3rd paragraph in the SCENARIOS section.

If you want to include comments and meta data, run

```
speccer -cmd text -sec SCENARIOS -at 3 -with-comments -with-meta
```

---

change content of a paragraphs of a section

Go into a folder that contains your spec. You will need to know the position of the paragraph you want to see. Therefor you might want to first print all positions of the section (here SCENARIOS) with

```
speccer -cmd positions -sec SCENARIOS
```

then you can run

```
speccer -cmd text -sec SCENARIOS -at 3 > temp.md
```

to save the current text and comments for the 3rd paragraph in the SCENARIOS section to the file `temp.md`.

Now fire up your editor to change the content. Then run

```
speccer -cmd text -sec SCENARIOS -at 3 -set temp.md
```

to set the text of the 3rd paragraph in the SCENARIOS section.

The comments and meta data of the 3rd paragraph are not modified by this operation. So you may want to change them too.

---

see, add and change comment to a paragraph of a section

Go into a folder that contains your spec. You will need to know the position of the paragraph you want to see. Therefor you might want to first print all positions of the section (here SCENARIOS) with

```
speccer -cmd positions -sec SCENARIOS
```

then you can run

```
speccer -cmd comment -sec SCENARIOS -at 3
```

to see the current comments for the 3rd paragraph in the SCENARIOS section.

Now if the author of your comment already made a comment to this paragraph and if you want to append to this old comment, you first have to save the old one.

To do this, run

```
speccer -cmd comment -sec SCENARIOS -at 3 -author Jim > temp.md
```

to save the comment of `Jim` to the file `temp.md`. Then you can edit this file. If you just want to remove the old comment or if there was no comment you may simply start with an empty file `temp.md`.

After you edited the file, run

```
speccer -cmd comment -sec SCENARIOS -at 3 -author Jim -set temp.md
```

to set the comment of `Jim` on the 3rd paragraph of the SCENARIOS section to the content of the file `temp.md`.

---

move a paragraph of a section

Go into a folder that contains your spec. You will need to know the position of the paragraph you want to move. Therefor you might want to first print all positions of the section (here SCENARIOS) with

```
speccer -cmd positions -sec SCENARIOS
```

then you can run

```
speccer -cmd move -sec SCENARIOS -at 3 -to 1
```

to move the 3rd paragraph in the SCENARIOS section to be the first.

---

delete a paragraph of a section

Go into a folder that contains your spec. You will need to know the position of the paragraph you want to delete. Therefor you might want to first print all positions of the section (here SCENARIOS) with

```
speccer -cmd positions -sec SCENARIOS
```

then you can run

```
speccer -cmd rm -sec SCENARIOS -at 3
```

to remove the 3rd paragraph in the SCENARIOS section.

---

see and change meta data of a paragraph of a section

Go into a folder that contains your spec. You will need to know the position of the paragraph you want to see the meta data. Therefor you might want to first print all positions of the section (here SCENARIOS) with

```
speccer -cmd positions -sec SCENARIOS
```

then you can run

```
speccer -cmd meta -sec SCENARIOS -at 3
```

to see all meta data of the 3rd paragraph in the SCENARIOS section.

If you want to change a datum you will have to pass the name:

```
speccer -cmd meta -sec SCENARIOS -at 3 -name RESPONSIBLE -set Jim
```