

Asciidoctor mal Groovy

Stephan Classen



Über mich

Stephan Classen

- » TDD enthusiast
- » Liebt Open Source
- » Hasst repetitive Aufgaben



Über mich

Stephan Classen

- » TDD enthusiast
- » Liebt Open Source
- » Hasst repetitive Aufgaben
- » Ein bisschen paranoid



Über mich

stephan.classen@canoo.com

 github.com/sklassen/sessions

Aktuelle Folien



[delivering end-user happiness]

canoo

Die Screenshot Extension

- » Idee und Proof of concept von François-Xavier Thoorens
- » Verwende GEB um einen Browser zu kontrollieren und Screenshots zu erstellen

Definitionen

» AsciiDoc

- A mature, plain-text writing format for authoring documentation, books and more.

» Asciidoctor

- A fast text processor and publishing toolchain for converting AsciiDoc files.

Advertising



A hand holding a yellow chalk marker is drawing a simple lightbulb on the right side of the word "Advertising". The board is covered with various terms related to advertising, written in different colors and sizes.

selling
customer
trademark
commercial
online
advertisement
internet
radio
mobile
media
ads
banner
concept
brand
marketing
newspapers
billboards

Wenig Formatierung viel Inhalt

== Parameters

==== screenshot

The _screenshot_ block macro has the syntax `screenshot::<url>[<parameters>]` and supports the following parameters (note: the URL is not needed when the browser was navigated to the page using a _geb_ block or other means)

name:: an optional unique file name (will be generated otherwise), the screenshot will be called `<name>.png` . This is the first positional parameter.

frame:: Try awesomeness with `iphone5` , `nexus5` or `browser` . This is the second positional parameter.

dimension:: size of the screenshot in the format `<width>x<height>` for instance 800x600. Also the values of 'frame' are supported.

selector:: the CSS-like dom selector to screenshot. Only this will be part of the image. For instance `div #login_window` .

Note: it is not allowed to set both 'frame' and 'dimension' or 'frame' and 'selector' for a screenshot.

Note: `screenshot::http://google.com[google]` and `screenshot::http://google.com[name=google]` are equivalent.

==== geb

The _geb_ block supports the following parameters:

dimension:: size of the screenshot in the format `<width>x<height>` for instance 800x600. Also the values of 'frame' are supported. This is the first positional parameter.

Wenig Formatierung viel Inhalt

== Parameters

==== screenshot

The `screenshot` block macro has the syntax `screenshot::<url>[<parameters>]` and supports the following parameters (note: the URL is not needed when the browser was navigated to the page using a `geb` block or other means)

`name`:: an optional unique file name (will be generated otherwise), the screenshot will be called `<name>.png`. This is the first positional parameter.

`frame`:: Try awesomeness with `iphone5`, `nexus5` or `browser`. This is the second positional parameter.

`dimension`:: size of the screenshot in the format `<width>x<height>` for instance 800x600. Also the values of 'frame' are supported.

`selector`:: the CSS-like dom selector to screenshot. Only this will be part of the image. For instance `div #login_window`.

Note: it is not allowed to set both 'frame' and 'dimension' or 'frame' and 'selector' for a screenshot.

Note: `screenshot::http://google.com[google]` and `screenshot::http://google.com[name=google]` are equivalent.

==== geb

The `geb` block supports the following parameters:

`dimension`:: size of the screenshot in the format `<width>x<height>` for instance 800x600. Also the values of 'frame' are supported. This is the first positional parameter.

Tabellen



[delivering end-user happiness]

canoo

Syntax Highlighting und Callouts

```
package com.mrhaki.adoc

class Sample {
    String username ①

    String toString() {
        "${username?.toUpperCase() ?: 'not-defined'}" ②
    }
}
```

- ① Simple property definition where Groovy will generate the `setUsername` and `getUsername` methods.
- ② Return username in upper case if set, otherwise return `not-defined`.

Includes



[delivering end-user happiness]

canoo

Attribute (Variablen)



Fenced Content



[delivering end-user happiness]

canoo

Conditionals



[delivering end-user happiness]

canoo

Kommentare

REALSOURCECOMMENTS

**/* WHEN I WROTE THIS, ONLY GOD AND
I UNDERSTOOD WHAT I WAS DOING
NOW, GOD ONLY KNOWS */**

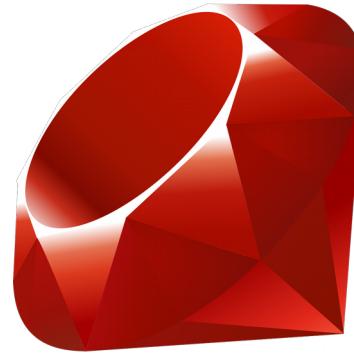
Ausgabe Formate (Backends)



LATEX



Runtimes



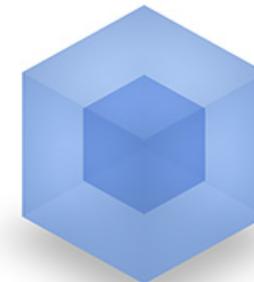
[delivering end-user happiness]

canoo

Build Tools



maven



webpack



[delivering end-user happiness]

canoo

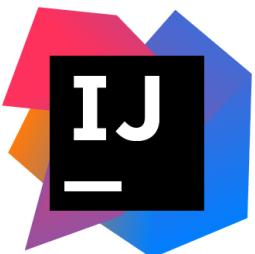
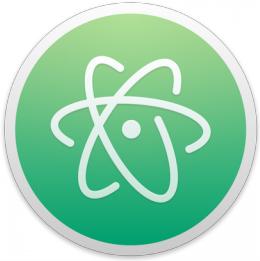
Erweiterungen



[delivering end-user happiness]

canoo

Instant Preview



README.adoc

```
1 = AsciidoctorJ Screenshot Extension
2
3 This AsciidoctorJ extension automates the documentation of your
4 No more hassles when you change simple settings like CSS or chan
5 Your documentation stays up to date!
6
7 This extension is based on the work of https://github.com/fix/as
8
9 == Quick reference
10
11 Basic usage is a block macro screenshot that points to a URL:
12 The title of the block macro is used as the caption of the image
13
14 ...
15 .Google Landing page
16 screenshot::http://google.com[google]
17
18
19 Use a block geb to control the browser (i.e. for filling in a
20 The content of the block is passed to http://www.gebish.org/[Geb]
21
22 ...
23 [geb]
24 go "http://google.com"
25 $("input", name: "q").value("asciidoctor")
26 waitFor(5){true}
27
28 == Parameters
29
30 === screenshot
31
32 The screenshot block macro has the syntax screenshot::<url>[<
33 (note: the URL is not needed when the browser was navigated to t
34
35 name:: an optional unique file name (will be generated otherwise
36 frame:: Try awesomeness with 'iphone5', 'nexus5' or 'browser'. T
37 dimension:: size of the screenshot in the format '<width>x<heigh
```

README.adoc Preview

This AsciidoctorJ extension automates the documentation of your webapp using screenshots. No more hassles when you change simple settings like CSS or change that button that was too big. Your documentation stays up to date!

This extension is based on the work of [François-Xavier Thorens](#)

Table of Contents

- 1. Quick reference
- 2. Parameters
 - 2.1. screenshot
 - 2.2. geb
- 3. Examples

1. Quick reference

Basic usage is a block macro `screenshot` that points to a URL: The title of the block macro is used as the caption of the image.

.Google Landing page

Github Diff Preview

asciidocorj-screenshot / README.adoc  or cancel

 Edit file  Preview changes Spaces 2 Soft wrap

AsciidoctorJ Screenshot Extension

This AsciidoctorJ is a awesome extension which automates the documentation of your webapp using screenshots. No more hassles when you change simple settings like CSS or change that button that was too big. Your documentation stays always up to date!

This extension is based on the work of [François-Xavier Thorens](#)

Quick reference

Basic usage is a block macro `screenshot` that points to a URL: The title of the block macro is used as the caption of the image.

Wer benutzt Asciidoctor

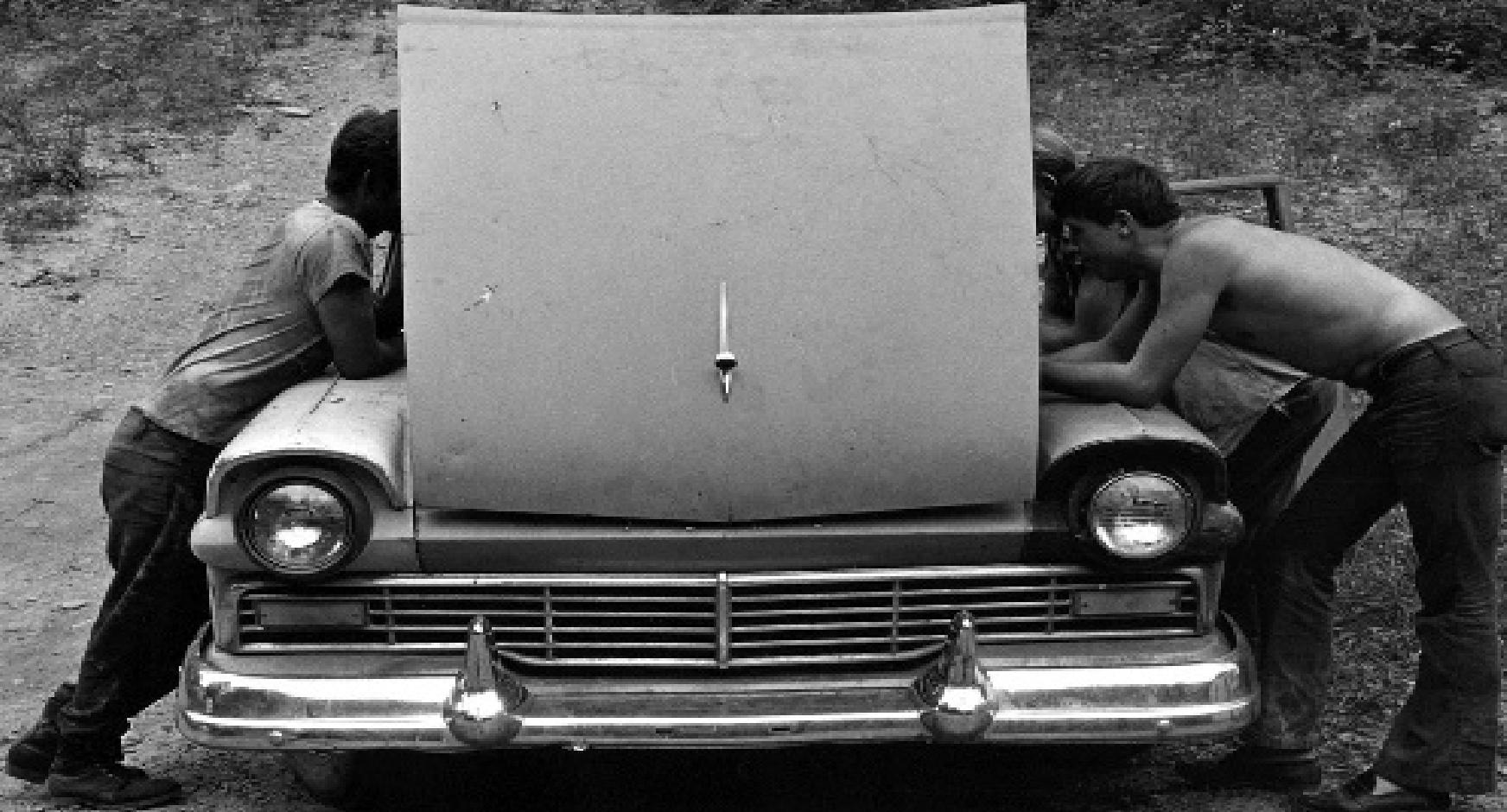


O'REILLY®

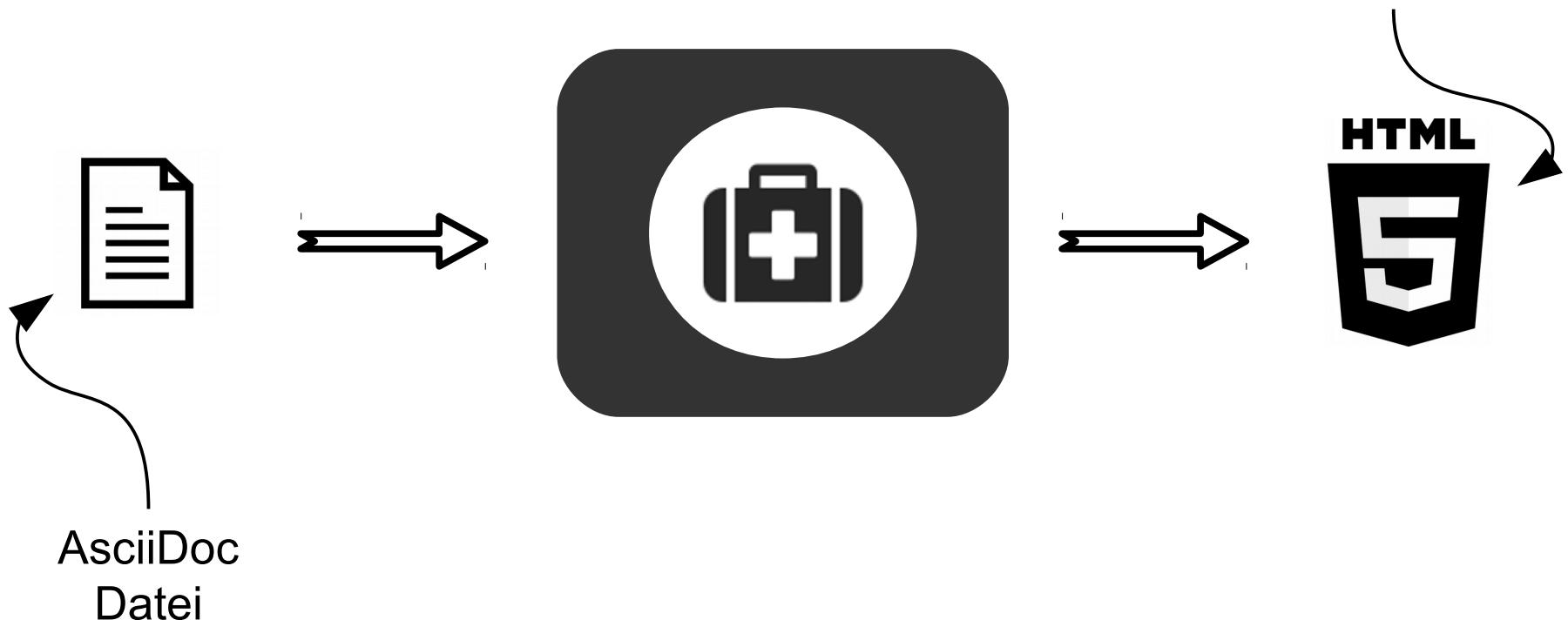


[delivering end-user happiness]

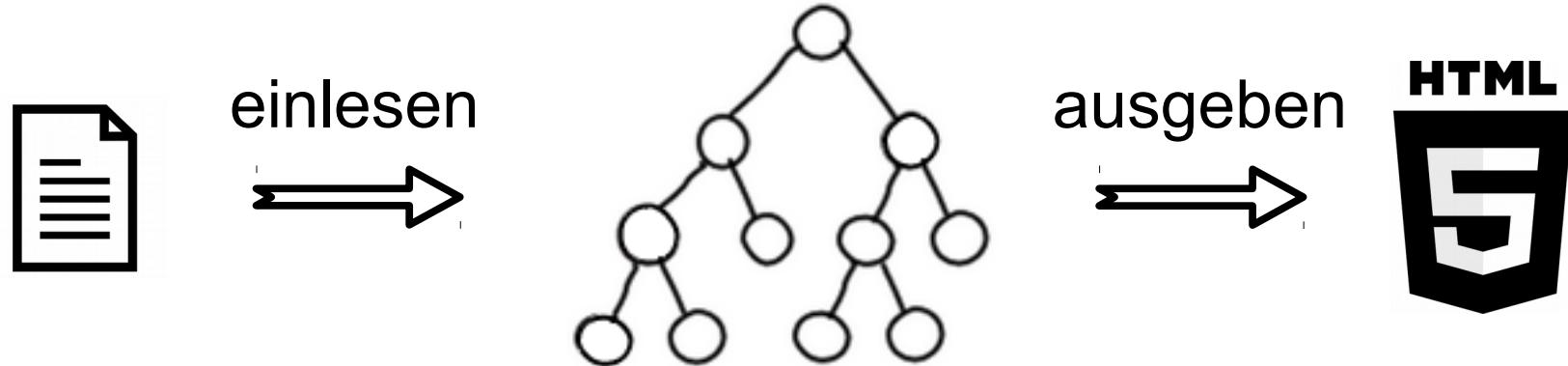
canoo



Asciidoctor Konvertierungsprozess



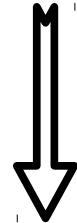
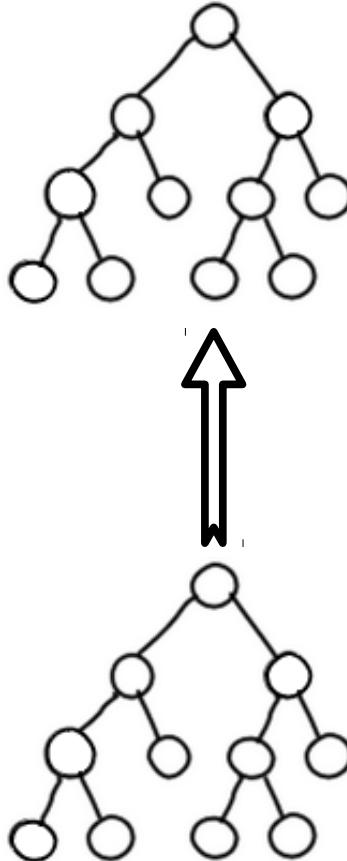
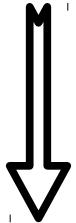
Asciidoctor Konvertierungsprozess



[delivering end-user happiness]

canoo

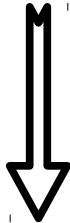
Details des Konvertierungsprozess



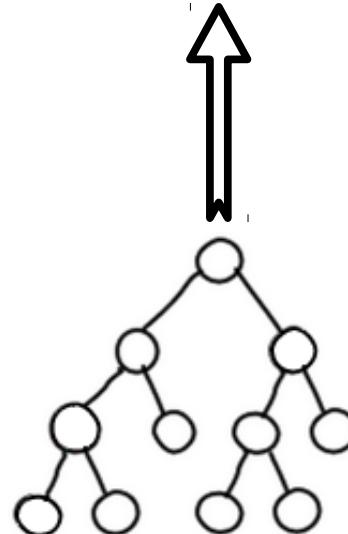
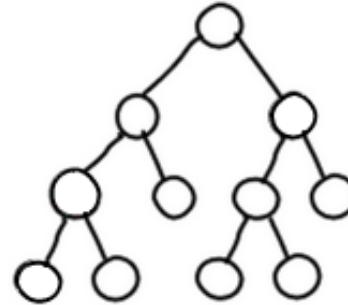
[delivering end-user happiness]

canoo

Details des Konvertierungsprozess



Pre

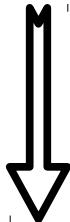


[delivering end-user happiness]

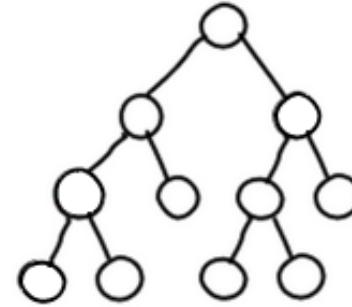


canoo

Details des Konvertierungsprozess



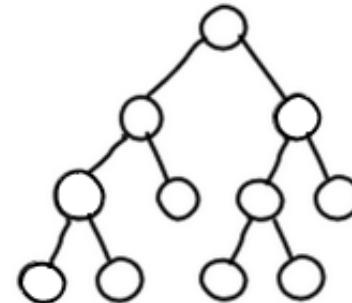
Pre



Block



Inline



[delivering end-user happiness]

canoo

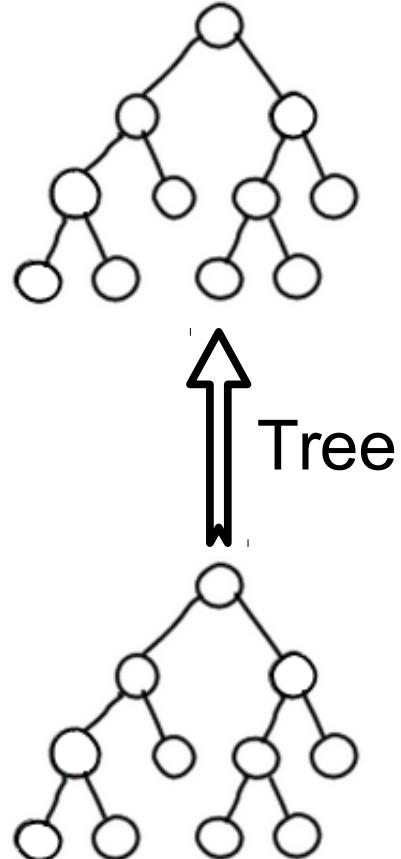
Details des Konvertierungsprozess



↓
Pre
↓

A double-headed vertical arrow indicating a bidirectional conversion process between the source file and the resulting structure.

Block
→
Inline

Two arrows pointing from the document icon to the tree diagram, indicating the conversion of a block-level element into an inline element.

↓
↓

A double-headed vertical arrow indicating a bidirectional conversion process between the resulting structure and the final output.

[delivering end-user happiness]

canoo

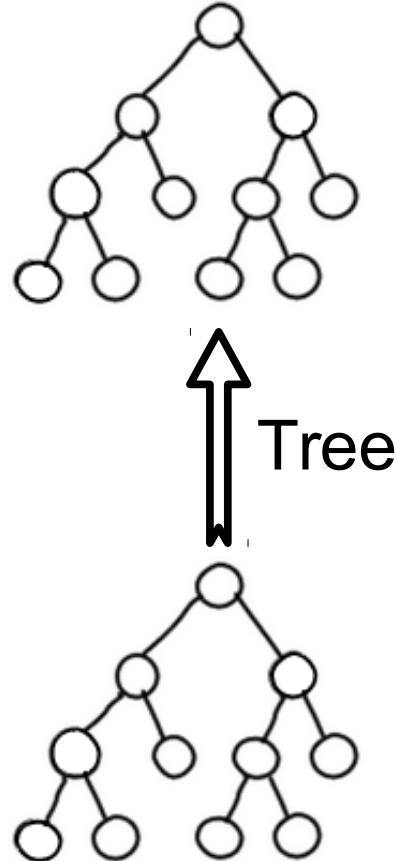
Details des Konvertierungsprozess



Pre
↓↑



Block
→
Inline



Backend
→



↓↑



[delivering end-user happiness]

canoo

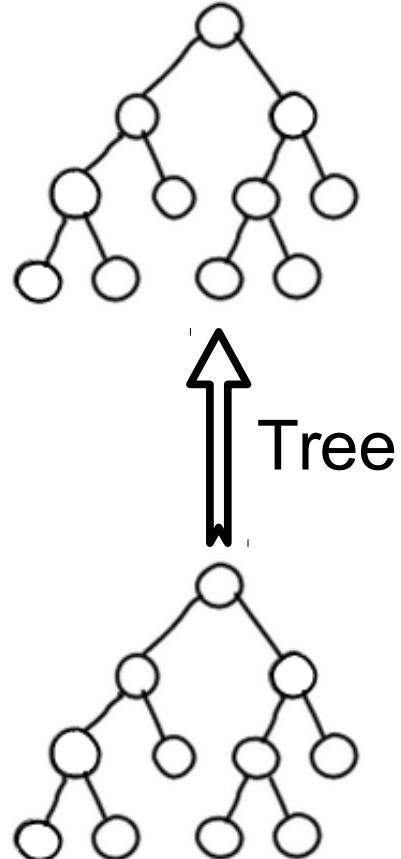
Details des Konvertierungsprozess



Pre
↓↑



Block
→
Inline



Backend
→



↓↑ Post



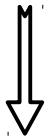
[delivering end-user happiness]

canoo

PreProcessor



» Bearbeitet AsciiDoc Dateien

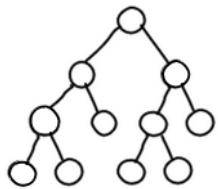
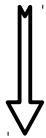
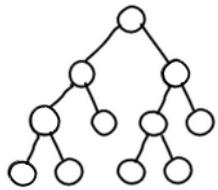


» Kann alles verändern/manipulieren



» Resultat geht zum Parsen

TreeProcessor

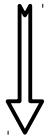


- » Bearbeitet den ganzen AST
- » Kann jeden Knoten im AST verändern
- » Resultat geht zum Backend

PostProcessor



» Bearbeitet das Resultat des Backend



» Kann alles verändern

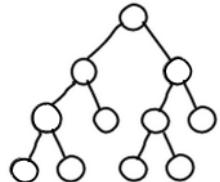
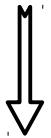


» Resultat ist der Finale Output

BlockProcessor



» Bearbeitet einen einzelnen Block



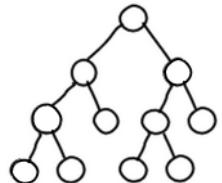
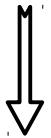
» Erhält den Kopf und Inhalt des Blocks

» Erzeugt einen Knoten im AST

BlockMacroProcessor



» Bearbeitet einen leeren Block



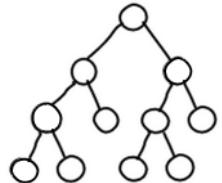
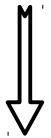
» Erhält den Kopf des Blocks

» Erzeugt einen Knoten im AST

InlineMacroProcessor

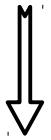
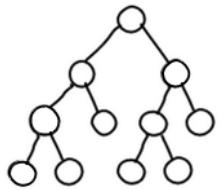


» Bearbeitet einen Teil eines Blockinhaltes



» Erzeugt einen Ersatz für den Teil

Backend



» Konvertiert den AST in das Output Format



Das Ziel

- » BlockMacroProcessor welcher Screenshots erstellt
- » BlockProcessor um den Browser zu kontrollieren

Vereinfachte Screenshot Extension

```
-- Capture Test

[geb]
....
go "file://example.html"
$("input").value("JavaLand 2017")
....

screenshot::"file://example.html"[ ]
```

3 Schritte zum Ziel

- » Macro welches ein statische Bild einfügt
- » Macro welches Screenshots erstellt
- » Block um den Browser zu kontrollieren

AsciidoctorJ 1.6.x

- » Annotationen für Extensions
- » Enums für Attribute und Konfigurationen
- » Helpermethoden

Extensions in Gradle

build.gradle

```
asciidoctor {  
    extensions {  
        inlinemacro(name: "issue") {  
            parent, issueNo, attributes ->  
                options = [  
                    "type": ":link",  
                    "target": "${issueTracker}/${issueNo}".toString()  
                ]  
                createInline(parent, "anchor", target, attributes, options).render()  
        }  
    }  
}
```

Schreibt eure Dokumentation mit Asciidoctor

- » <http://asciidoctor.org/>
- » <http://discuss.asciidoctor.org/>
- » <http://mrhaki.blogspot.ch/>