The Stacks project framework

Pieter Belmans

August 2 2017

The Stacks project framework

The Stacks project framework

Python

- plasTEX
- Flask
- peewee

The Stacks project framework

Python

- plasTEX
- Flask
- peewee

responsive design in HTML, CSS and JS

Old framework

Unless someone is willing to work on outdated spaghetti code which will be replaced soon I won't discuss this.

Old framework

Unless someone is willing to work on outdated spaghetti code which will be replaced soon I won't discuss this.

Except for a mobile stylesheet, which doesn't require knowing the framework anyway.

New framework: Gerby

some modest goals:

1. better maintainable, more robust
2. make it mobile-friendly
3. easier to run locally (right now you need an Apache webserver with PHP and SQLite, and perform a gazillion configuration steps)
4. applicable to other projects: e.g. Jacob Lurie's books
5. ...

New framework: Gerby

some modest goals:

1. better maintainable, more robust
2. make it mobile-friendly
3. easier to run locally (right now you need an Apache webserver with PHP and SQLite, and perform a gazillion configuration steps)
4. applicable to other projects: e.g. Jacob Lurie's books
5. . . .

3 parts (all Python!)

TeX splitting up into tags, converting to HTML

New framework: Gerby

some modest goals:

1. better maintainable, more robust
2. make it mobile-friendly
3. easier to run locally (right now you need an Apache webserver with PHP and SQLite, and perform a gazillion configuration steps)
4. applicable to other projects: e.g. Jacob Lurie's books
5. . . .

3 parts (all Python!)

TeX splitting up into tags, converting to HTML

database updating tags, extracting extra information

New framework: Gerby

some modest goals:

1. better maintainable, more robust
2. make it mobile-friendly
3. easier to run locally (right now you need an Apache webserver with PHP and SQLite, and perform a gazillion configuration steps)
4. applicable to other projects: e.g. Jacob Lurie's books
5. . . .

3 parts (all Python!)

TeX splitting up into tags, converting to HTML

database updating tags, extracting extra information

website interface

part 1: TEX stuff

plasTEX: `https://github.com/tiarno/plastex`

1. DOM-based
2. convert LATEX to HTML, DocBook, text, . . .
3. can parse macros and packages

part 1: TEX stuff

plasTEX: `https://github.com/tiarno/plastex`

1. DOM-based
2. convert LATEX to HTML, DocBook, text, . . .
3. can parse macros and packages

new renderer for plasTEX: Gerby, see `https://github.com/pbelmans/plastex/tree/gerby`

1. whenever *something* (section, lemma, equation) has a label, look up its tag
2. convert *something* to HTML, and write this to a text file
3. filename contains some metadata: `lemma-10.19.1-00DV-algebra-lemma-NAK.tag`
4. proofs are separate entities (because `\begin{proof}...\end{proof}` is outside the environment)

part 1: what needs to happen?

1. fix various bugs (indicated by `TODO` or `FIXME` in the code)
2. implementation:
    - handling `\item`'s
    - use `pdf2svg` for `xypic` or `tikz-cd`?
    - use `mathjax-server`?
3. improve documentation
4. check output on Stacks project, Higher Topos Theory

part 1: what needs to happen?

1. fix various bugs (indicated by `TODO` or `FIXME` in the code)
2. implementation:
   - handling `\item`'s
   - use `pdf2svg` for `xypic` or `tikz-cd`?
   - use `mathjax-server`?
3. improve documentation
4. check output on Stacks project, Higher Topos Theory

   no programming knowledge needed!

part 2: database (and other tools)

need to populate a database (SQLite at the moment) with

1. the output from plasTEX
2. navigation data
3. bibliography data
4. dependency data
5. . . .

part 2: database (and other tools)

need to populate a database (SQLite at the moment) with

1. the output from plasTeX
2. navigation data
3. bibliography data
4. dependency data
5. . . .

but also

1. build pdf's
2. generate graphs
3. read off Git history

part 2: what needs to happen?

everything from the previous slide, except output from plasTeX

part 2: what needs to happen?

everything from the previous slide, except output from plasTEX

1. ~~the output from plasTEX~~
2. navigation data
3. bibliography data
4. dependency data
5. . . .

but also

1. build pdf's
2. generate graphs
3. read off Git history

part 3: website

based on Flask

> *Flask is a micro web framework written in Python and based on the Werkzeug toolkit and Jinja2 template engine. It is used by Pinterest, LinkedIn, . . .*

part 3: website

based on Flask

> *Flask is a micro web framework written in Python and based on the Werkzeug toolkit and Jinja2 template engine. It is used by Pinterest, LinkedIn, . . .*

1. it is *micro*, so no need for reading hundred of pages of documentation
2. super easy to run locally (if you don't have access to an internet connection)
3. Jinja2 is what is used in plasTEX too

part 3: what needs to happen?

in short: a lot

part 3: what needs to happen?

in short: a lot

also: focus on mobile devices (it's not 2012 anymore!)

part 3: what needs to happen?

in short: a lot

also: focus on mobile devices (it's not 2012 anymore!)

1. overall layout
2. navigation for tags
3. bibliography
4. static pages
5. search
6. . . .

part 3: what needs to happen?

in short: a lot

also: focus on mobile devices (it's not 2012 anymore!)

1. overall layout
2. navigation for tags
3. bibliography
4. static pages
5. search
6. . . .

different setup?

part 3: what needs to happen?

in short: a lot

also: focus on mobile devices (it's not 2012 anymore!)

1. overall layout
2. navigation for tags
3. bibliography
4. static pages
5. search
6. . . .

different setup? maybe use JavaScript to have a setup which works on mobile devices?

Possible new features?

we have a brainstorm session on Thursday