

Contents

1 Προαπαιτούμενα	1
2 Clone του αποθετηρίου στο μηχάνημά μας	2
3 Επεξεργασία	2
3.0.1 Τρόπος 1	2
3.0.2 Τρόπος 2	3
4 Εισαγωγή των αρχείων μας	3
5 Ανέβασμα των αρχείων	3
6 Netlify	4
7 Pull request	4
8 Παρατηρήσεις	4
9 Netlify-CLI	5
9.1 Windows	5
9.1.1 ruby + jekyll	5
9.1.2 Netlify cli setup	7
9.2 Arch Linux	8
10 Κατασκευή Βιβλίου	8
10.1 Προαπαιτούμενα	8
10.1.1 Εγκατάσταση fonts	8
10.2 Επεξεργασία	9
10.3 Μετατροπή από .tex σε pdf	9

DISCLAIMER: Διαβάστε ολόκληρες τις οδηγίες μέχρι το τέλος πριν ξεκινήσετε. Οι οδηγίες καλό είναι να μην να ακολουθούνται κατά γράμμα αλλά όχι δογματικά, μην τις εκτελείτε στα τυφλά, διαβάστε και αναρωτηθείτε τι κάνει η κάθε εντολή και γιατί την εκτελούμε. Μόνο έτσι θα καταλάβετε πραγματικά τη λογική των παραδοτέων και των εργαλείων που χρησιμοποιούμε.

1 Προαπαιτούμενα

1. Έχουμε εγκατεστημένη μία Linux διανομή στο σύστημά μας και κατά προτίμηση τα Arch.

2. Έχουμε κάνει fork στο προφίλ μας το **site** από τον οργανισμό μας.
3. Έχουμε κάνει fork στο προφίλ μας τα **submodules** από τον οργανισμό μας.

2 Clone του αποθετηρίου στο μηχάνημά μας

1. `git clone https://github.com/*your-username*/site`

3 Επεξεργασία

3.0.1 Τρόπος 1

1. `true > .gitmodules` (Αδειάζει το αρχείο)
2. `git add .`
3. `git rm --cached _gallery _bibliography _notes _quotes images`
4. `rm -R _gallery _bibliography _notes _quotes images`
5. `cd _includes`
6. `git rm --cached extras text`
7. `rm -R extras text`
8. `git add .`
9. `git commit -m "your message"`
10. `git push origin`
11. `cd ..`

3.0.1.1 Προσθήκη των submodules

1. `git submodule add https://github.com/*your-username*/_gallery`
2. `git submodule add https://github.com/*your-username*/_quotes`
3. `git submodule add https://github.com/*your-username*/images`
4. `git submodule add https://github.com/*your-username*/bibliography _bi`
5. `git submodule add https://github.com/*your-username*/notes _notes`
6. `cd _includes`

7. `git submodule add https://github.com/*your-username*/extras`
8. `git submodule add https://github.com/*your-username*/text`
9. `cd ..`
10. `git add .`
11. `git commit -m "your-message"`
12. `git push origin`

3.0.2 Τρόπος 2

1. αλλαγή στα link των submodules όπως στο βήμα **Προσθήκη των submodules** στο αρχείο `.gitmodules`
2. `git submodule update --remote --init`
3. `git submodule update --remote --merge`

Πριν κάνετε τις δικές σας αλλαγές πάτε στην ενότητα Netlify για να βεβαιωθείτε ότι το site λειτουργεί σωστά

4 Εισαγωγή των αρχείων μας

Τα αρχεία.

1. Τοποθετούμε τα αρχεία `.md` στο φάκελο `_gallery`
2. Τοποθετούμε τις εικόνες στο φάκελο `images`

5 Ανέβασμα των αρχείων

1. Βλέπουμε τι αλλαγές έχουμε κάνει και σε ποιο φάκελο με την εντολή `git status`
2. Κάνουμε `cd` στους φακέλους που έχουμε κάνει τις αλλαγές και:
 - `git add .`
 - `git commit -m "your-message"`
 - `git push origin`
1. Κάνουμε το ίδιο και στο directory **site** και ελέγχουμε με το `git status` μέχρι να μας βγάλει το μήνυμα:

On branch master
Your branch is up to date with 'origin/master'.
nothing to commit, working tree clean

6 Netlify

1. Συνδεόμαστε στην πλατφόρμα του Netlify μέσω του λογαριασμού μας στο GitHub
2. Επιλέγουμε Add a new site → Import an existing project
3. Επιλέγουμε το GitHub και δίνουμε τις απαραίτητες άδειες
4. Πατάμε το Only select repositories και επιλέγουμε το site
5. Επιλέγουμε το μενού Show advanced → New variable και δίνουμε ως key το RUBY_VERSION και ως value το 2.6.2.47
6. Κάνουμε deploy το site

Θα πρέπει να μπορούμε να δούμε το περιεχόμενο που προσθέσαμε

7 Pull request

1. Πηγαίνουμε στα αποθετήρια μας που έχουμε κάνει αλλαγές και πατάμε το κουμπί contribute → create a pull request
2. Γράφουμε μια σύντομη περιγραφή των αλλαγών που έχουμε κάνει και δίνουμε τα links από την ιστοσελίδα μας που πιστοποιούν ότι οι αλλαγές μας δουλεύουν και περιμένουμε την έγκριση από τους διαχειριστές

8 Παρατηρήσεις

1. Για τα pull request επιλέγουμε μόνο τα submodules που έχουμε κάνει αλλαγές και όχι το site
2. **Tip:** Είναι καλύτερο να κάνουμε add και commit συγκεκριμένα files κάθε φορά και να δίνουμε ακριβή περιγραφή των αλλαγών που έχουμε κάνει. Με αυτό τον τρόπο είναι πιο εύκολο να ελεγχθούν τα αρχεία σε ένα pull request.

3. **Tip:** Σε περίπτωση που κάποιος άλλος από την ομάδα μας έχει κάνει επιτυχώς ένα pull request πριν από εμάς θα πρέπει να ακολουθήσουμε δύο επιπλέον βήματα:
- Βλέπουμε αν τα forked αποθετήρια μας είναι synced με το αποθετήριο του οργανισμού μας, αν όχι τότε πατάμε το κουμπί sync. Καλό είναι να κάνουμε sync πριν ανεβάσουμε εμείς τις δικές μας αλλαγές για να αποφεύγονται συγκρούσεις.
 - Στο terminal χρησιμοποιούμε την εντολή `git pull origin` αφού έχουμε δει ποιοι φάκελοι χρειάζονται αλλαγές με την εντολή `git status`

9 Netlify-CLI

9.1 Windows

Απαραίτητα: `node.js`, `ruby` + `jeekyll`, `Netlify cli` ### `node.js`
`install`

- Ακολουθούμε απλά το `install` του και για καλό και για κακό βάζουμε και την επιλογή να κάνει `install` και το `Chocolatey` (δε δοκιμάστηκε χωρίς αυτό)

9.1.1 `ruby` + `jeekyll`

- Ανοίγουμε το `installer`

Σε αυτή τη φάση κάνουμε και τις 3 επιλογές

- Μόλις τελειώσει με τις επιλογές μπορούμε άπλα να το κλείσουμε.
- Κοιτάζουμε αν το έχουμε εγκαταστήσει σωστά με την εντολή `jeekyll -v`
- Στη συνέχεια χρησιμοποιούμε την εντολή `bundler install` για να μας δημιουργήσει όλα τα απαραίτητα `gems` που χρειαζόμαστε.
- Μετά από αυτό κάλο είναι να γίνει ένα `reboot` (το συνιστά και το `installer`)

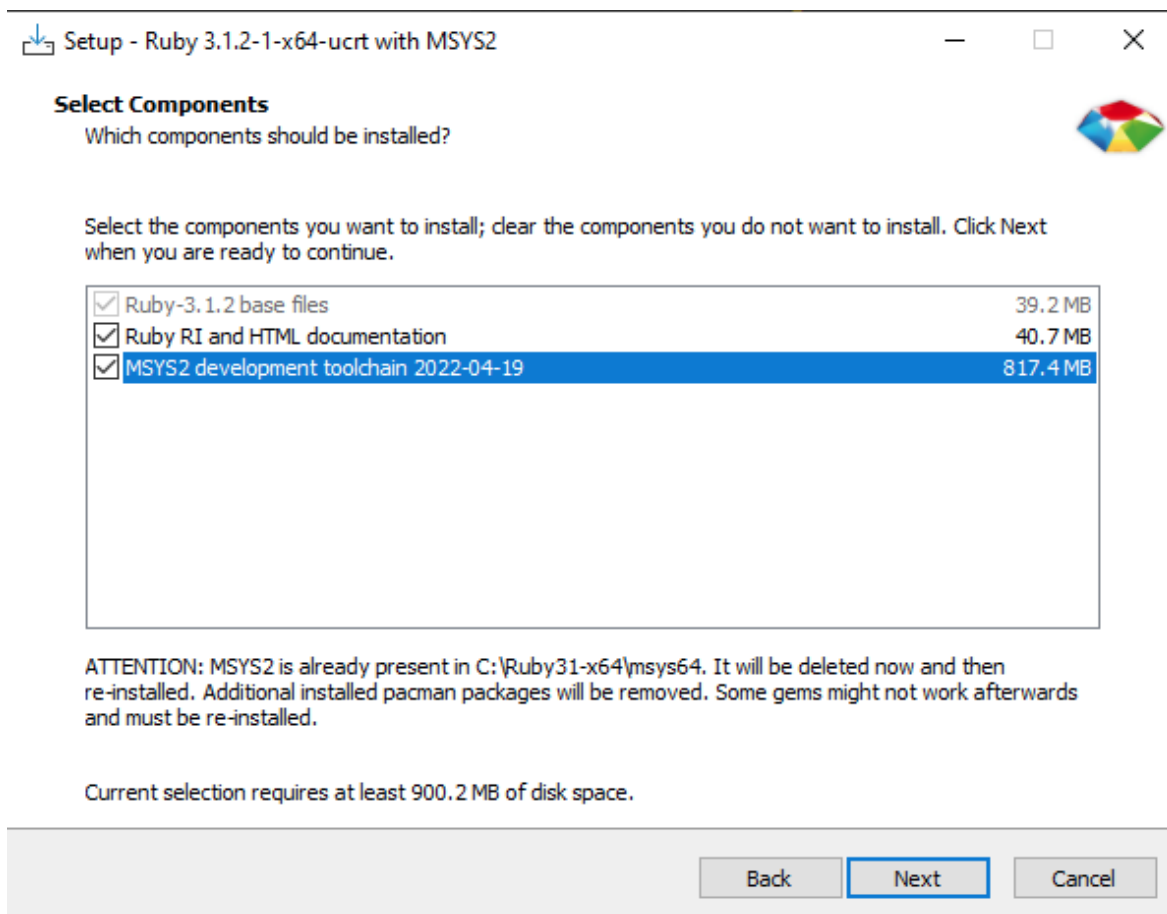


Figure 1: image

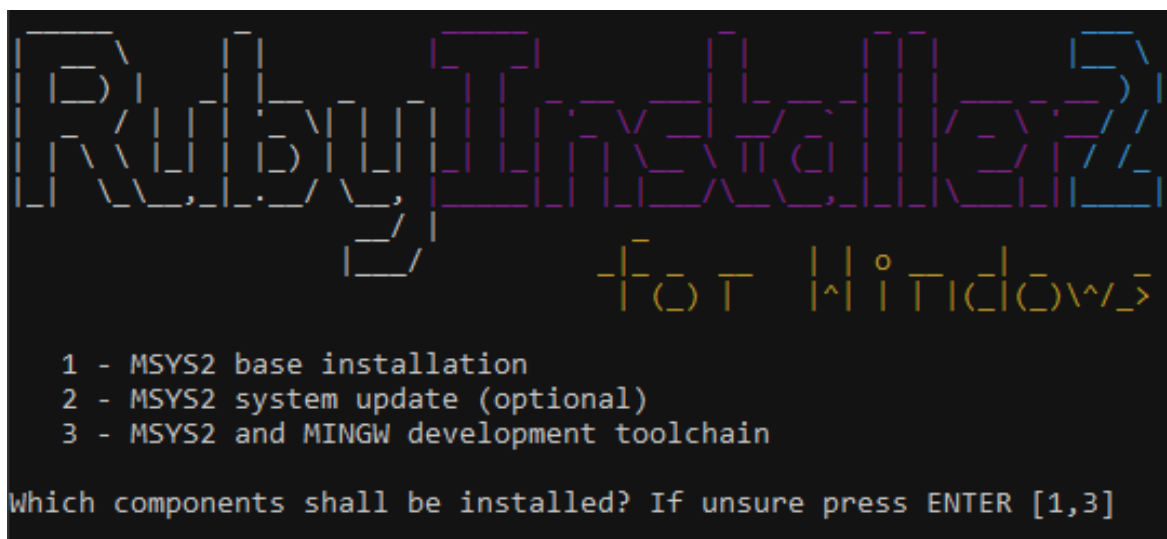


Figure 2: image

9.1.2 Netlify cli setup

- Αφού έχουμε τελειώσει με όλα τα install
- χρησιμοποιούμε την εντολή `npm install netlify-cli -g` για την εγκατάσταση του Netlify cli
- κάνουμε `netlify` για να δούμε ότι έχει γίνει η εγκατάσταση όπου θα μας εμφανίσει το παρακάτω πίνακα

```
netlify-cli/12.1.0 win32-x64 node-v18.12.1
USAGE
$ netlify [COMMAND]

COMMANDS
$ addons      (Beta) Manage Netlify Add-ons
$ api         Run any Netlify API method
$ build       (Beta) Build on your local machine
$ completion  (Beta) Generate shell completion script
$ deploy      Create a new deploy from the contents of a folder
$ dev         Local dev server
$ env         Control environment variables for the current site
$ functions   Manage netlify functions
$ recipes     (Beta) Create and modify files in a project using pre-defined recipes
$ graph       (Beta) Control the Netlify Graph functions for the current site
$ init        Configure continuous deployment for a new or existing site. To create a
              new site without continuous deployment, use `netlify sites:create`
$ link        Link a local repo or project folder to an existing site on Netlify
$ lm          Handle Netlify Large Media operations
$ login       Login to your Netlify account
$ open        Open settings for the site linked to the current folder
$ sites       Handle various site operations
$ status      Print status information
$ switch      Switch your active Netlify account
$ unlink      Unlink a local folder from a Netlify site
$ watch       Watch for site deploy to finish
```

Figure 3: image

ΔΕΝ ΞΕΧΝΑΜΕ ΝΑ ΒΑΛΟΥΜΕ ΤΗ ΣΩΣΤΗ VERSION ΤΗΣ RUBY `netlify env:set RUBY_VERSION 2.6.2.47`

1. Χρησιμοποιούμε την εντολή `netlify login` όπου θα μας ανοίξει τη σελίδα του Netlify για να κάνουμε authorize το τερματικό μας, στη συνέχεια πατάμε `netlify init`
2. πατάμε την εντολή που μας λέει να συνδεθούμε σε ήδη υπάρχον site και στη συνέχεια θα πατήσουμε για το repository που είναι το site μας.
3. `netlify build` για να χτιστεί η σελίδα μας και `netlify deploy` για να γίνει deploy η σελίδα σε ένα draft link στο οποίο μπορείτε να τεστάρετε τις αλλαγές σας χωρίς να χρειάζεστε

να κάνετε commit πολλά πέρα από αυτά που χρειάζεστε και να έχετε ένα πολύπλοκο history στο repository σας.

9.2 Arch Linux

1. Εγκαθιστούμε τα nodejs, ruby
 - `sudo pacman -S nodejs`
 - `sudo pacman -S ruby base-devel`
1. Προσθέτουμε το ruby στο PATH (το ίδιο ισχύει για όλα τα shells)
 - `echo # Install Ruby Gems to ~/gems >> ~/.bashrc *`
 - `echo export GEM_HOME="$HOME/gems" >> ~/.bashrc`
 - `echo export PATH="$HOME/gems/bin:$PATH" >> ~/.bashrc`
 - `source ~/.bashrc`
1. Εγκαθιστούμε το Jekyll και το Bundler
 - `gem install jekyll bundler`
 - `bundler install`
1. Εγκαθιστούμε το Netlify
 - `npm install netlify-cli -g`

Ακολουθούμε τις οδηγίες που αναφέρονται για τα Windows

10 Κατασκευή Βιβλίου

10.1 Προαπαιτούμενα

1. `sudo pacman -S pandoc`
2. `sudo pacman -S texlive-most (2GB install)`
3. `sudo pip install pandoc-fignos`

10.1.1 Εγκατάσταση fonts

1. Βρισκόμαστε στο ~/ directory
2. `ls -la`

3. Βλέπουμε αν υπάρχει ο φάκελος `.fonts` αν δεν υπάρχει τότε `mkdir .fonts`
 4. κατεβάζουμε σε zip το font που θέλουμε από την ιστοσελίδα του
 5. `unzip font-που-κατεβάσαμε.zip -d ~/.fonts`
- optional: `yay community/ttf-meslo-nerd-font-powerlevel10k`

10.2 Επεξεργασία

- `git clone https://github.com/~your-username~/kallipos.git`
- `git submodule update --remote --init`
- `chmod +x make-latex.sh`
- `mkdir latex`

Από το `make-latex.sh` βάζουμε σε σχόλιο το `sed -i s+figure+Εικόνα+g` (αυτή η εντολή λειτουργεί μόνο σε Mac συστήματα)

Αλλάζουμε στο αρχείο `figure.lua` τη γραμμή `12 src = ".." .. src` : `arrow_right: src = "." .. src`. Ο λόγος που το κάνουμε αυτό είναι επειδή θέλουμε το αρχείο να ψάχνει στο `current directory` και όχι σε ένα `directory` πίσω.

10.3 Μετατροπή από `.tex` σε `pdf`

1. Κάνουμε πρώτα ένα merge των αρχείων `tex` σε ένα αρχείο `book.tex` (δεν το σβήνουμε καθώς θα χρειαστεί σε επόμενο παραδοτέο)
 - `pandoc -s latex/*.tex -o book.tex`
1. Μετατροπή του `book.tex` σε `book.pdf`
 - `pandoc -N -variable "geometry=margin=1.2in" -variable mainfont="το δικο σας font" -variable sansfont="το δικο σας font" -variable monofont="το δικο σας font" -variable fontsize=12pt -variable version=2.0 tex/book.tex -pdf-engine=xelatex -toc -o book.pdf`

Η παραπάνω εντολή θα βγάλει ένα warning ότι χρησιμοποιείτε λάθος μετατροπέα (`xelatex`) παρόλο που χρησιμοποιείτε τον σωστό. Δεν υπάρχει κανέναν πρόβλημα απλά αγνοήστε το.

Για τα φίλτρα της lua πρέπει να τα δηλώσουμε μέσα στα .txt αρχεία που θέλουμε να πειράξουμε και να τα βάλουμε και στο make-latex.sh για να γίνουν οι εισαγωγές μας στο pdf. -
pandoc --lua-filter=~your-filter~.lua

Βοηθητικό documentation:

- <https://pandoc.org/MANUAL.html#>
- <https://pandoc.org/lua-filters.html>
- https://garrettgman.github.io/rmarkdown/authoring_pandoc_markdown.htm

Αν σας βοήθησε το guide κάντε ένα upvote εδώ για να το βρουν και άλλοι.

Made with :heart: by Second Time Is The Charm