What the Git?

The Old World

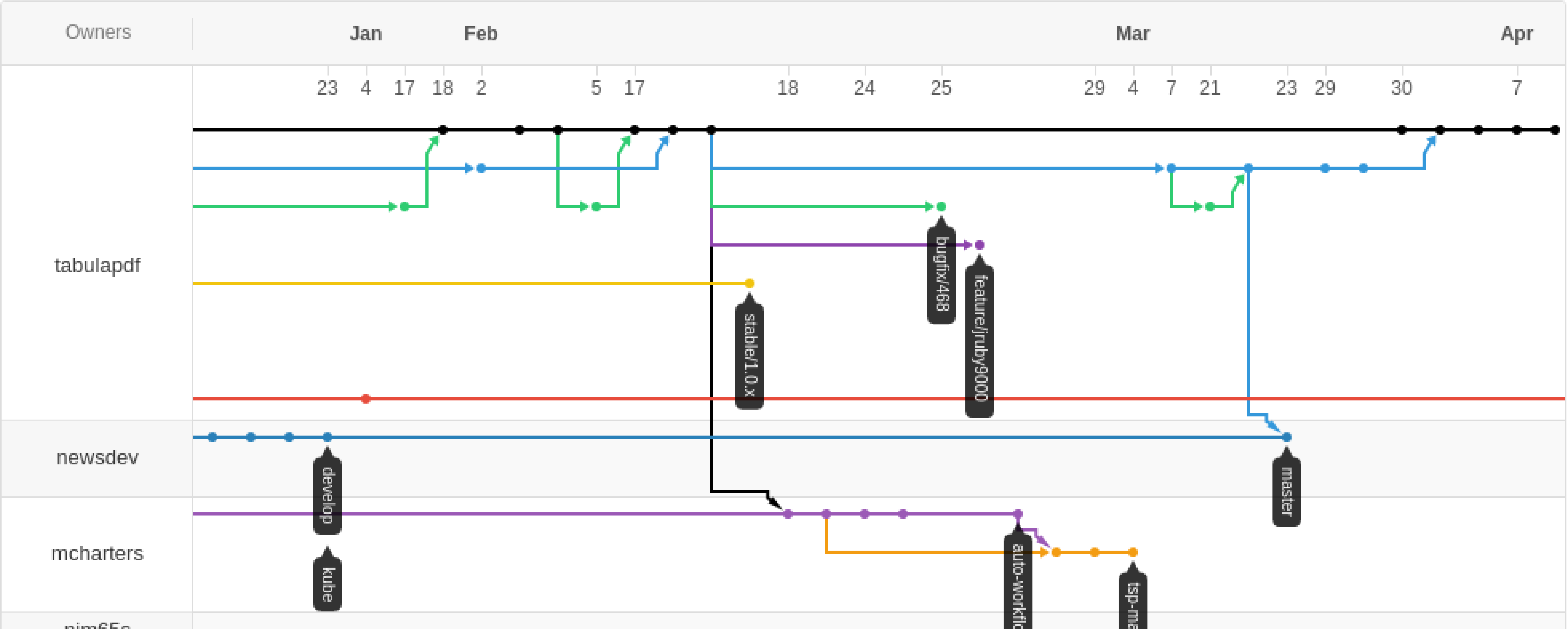
- Have you ever had to collaborate on a project by
  - Shuttling a USB drive back and forth
  - Using Dropbox
  - E-mailing your document around

- Have you ever accidentally deleted someone else's changes because you both saved the same file?

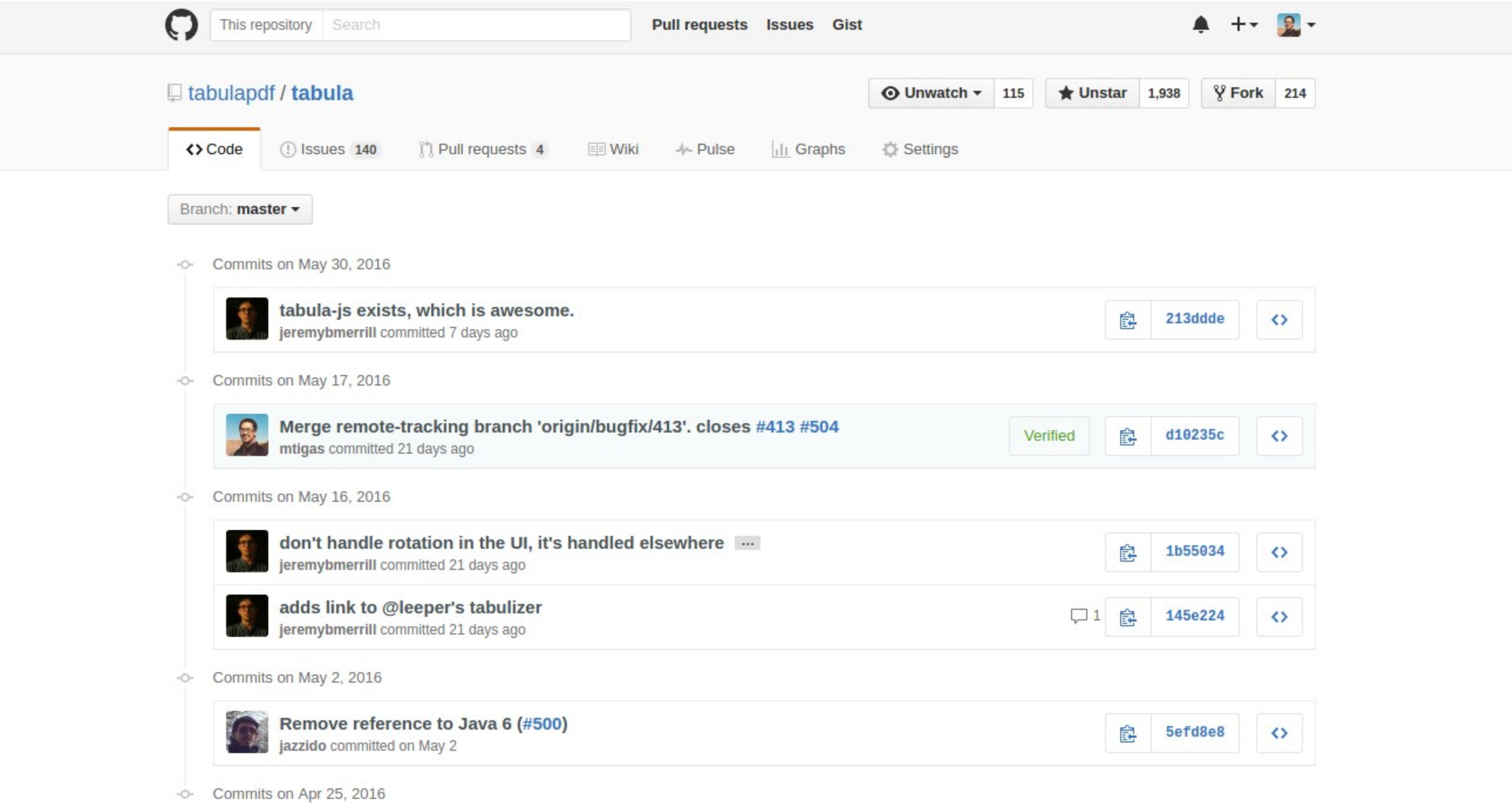
Git

- Saves your work
- Helps you back it up online
- Lets you go back to any saved version of any file
  - “Data diary, but for code”

- Helps you collaborate with other people – even if you're editing the same files in a project



https://github.com/tabulapdf/tabula/network



tabulapdf / tabula

Commits on May 30, 2016

tabula-js exists, which is awesome.
jeremybmerrill committed 7 days ago — 213ddde

Commits on May 17, 2016

Merge remote-tracking branch 'origin/bugfix/413'. closes #413 #504
mtigas committed 21 days ago — Verified — d10235c

Commits on May 16, 2016

don't handle rotation in the UI, it's handled elsewhere …
jeremybmerrill committed 21 days ago — 1b55034

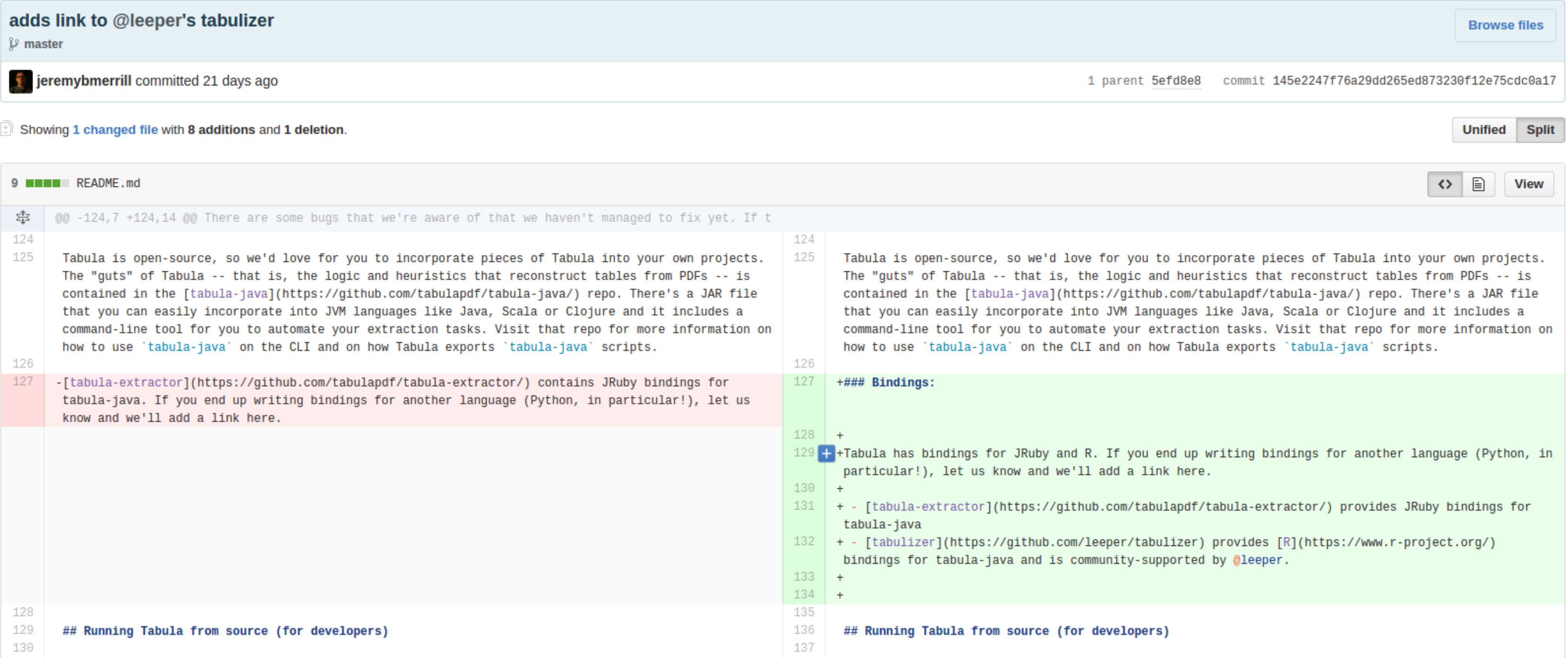
adds link to @leeper's tabulizer
jeremybmerrill committed 21 days ago — 1 — 145e224

Commits on May 2, 2016

Remove reference to Java 6 (#500)
jazzido committed on May 2 — 5efd8e8

Commits on Apr 25, 2016

https://github.com/tabulapdf/tabula/commits/master



adds link to @leeper's tabulizer

master

jeremybmerrill committed 21 days ago

1 parent 5efd8e8 commit 145e2247f76a29dd265ed873230f12e75cdc0a17

Showing **1 changed file** with **8 additions** and **1 deletion**.

9 README.md

```
@@ -124,7 +124,14 @@ There are some bugs that we're aware of that we haven't managed to fix yet. If t
```

```

 Tabula is open-source, so we'd love for you to incorporate pieces of Tabula into your own projects. The "guts" of Tabula -- that is, the logic and heuristics that reconstruct tables from PDFs -- is contained in the [tabula-java](https://github.com/tabulapdf/tabula-java/) repo. There's a JAR file that you can easily incorporate into JVM languages like Java, Scala or Clojure and it includes a command-line tool for you to automate your extraction tasks. Visit that repo for more information on how to use `tabula-java` on the CLI and on how Tabula exports `tabula-java` scripts.

-[tabula-extractor](https://github.com/tabulapdf/tabula-extractor/) contains JRuby bindings for tabula-java. If you end up writing bindings for another language (Python, in particular!), let us know and we'll add a link here.

 ## Running Tabula from source (for developers)

```

```

 Tabula is open-source, so we'd love for you to incorporate pieces of Tabula into your own projects. The "guts" of Tabula -- that is, the logic and heuristics that reconstruct tables from PDFs -- is contained in the [tabula-java](https://github.com/tabulapdf/tabula-java/) repo. There's a JAR file that you can easily incorporate into JVM languages like Java, Scala or Clojure and it includes a command-line tool for you to automate your extraction tasks. Visit that repo for more information on how to use `tabula-java` on the CLI and on how Tabula exports `tabula-java` scripts.

+### Bindings:
+
+Tabula has bindings for JRuby and R. If you end up writing bindings for another language (Python, in particular!), let us know and we'll add a link here.
+
+ - [tabula-extractor](https://github.com/tabulapdf/tabula-extractor/) provides JRuby bindings for tabula-java
+ - [tabulizer](https://github.com/leeper/tabulizer) provides [R](https://www.r-project.org/) bindings for tabula-java and is community-supported by @leeper.
+
+

 ## Running Tabula from source (for developers)

```

https://github.com/tabulapdf/tabula/commit/145e2247f76a29dd265ed873230f12e75cdc0a17

Jargon

- Repository (n)
- Commit (n, v)
- Branch (n)

- Clone (v) / Fork (n, v) / Remote (n)
- Push (v) / Pull (v)
- Check out (v)
- Merge (v)

Repository

- “repo” for short

- noun: A “project” in git.

- Contains a folder and everything inside it*.
  - …And the entire history of changes to that folder.

    (you can make some exceptions, for things like super large data files)*

Commit

- noun: a particular version or “snapshot” of your repository.
  - “look at my most recent commit”

- verb: to create

Branch

- noun: a version of a repository that can be worked on in parallel to other branches

```
● (“live”)
 \ - - - ● - ● - ● (“master”)
          \ - - ● (“mike-experiment”)
```

Clone

- verb: to download a copy of a repository to your computer

Fork

- verb: to create your own copy of a repository
- This is like making a copy of a shared document in Google Docs so you can make your own edits

Remote

- noun: a copy of a repository that's somewhere other than your computer.
- A remote can be:
  - Your copy in GitHub (or another web service)
  - Someone else's fork of the same repository on GitHub (or etc.)
  - Other crazy things
    - Folders on your computer (you can stash your repository in Dropbox, etc)

Push

- verb: to upload your commits to the website

Pull

- verb: to get new commits from the website

Check out

- verb: to switch to a different branch

Merge

- It's what you do when you're making commits at the same time as somebody else and you're **pulling** their commits so you have a consistent history.

```
- ● - ● - ● - ● (“master”)
   \ - - ● - ● - ● - ● (“mike-experiment”)
```

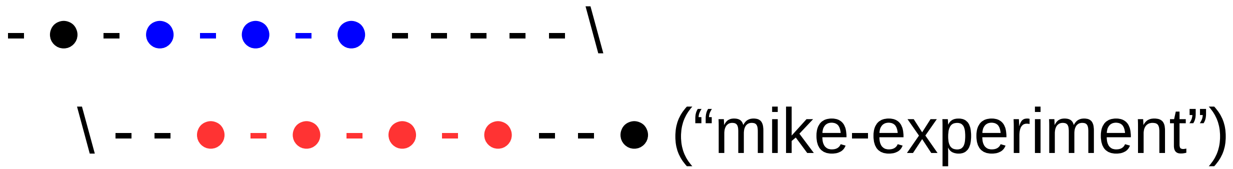
Merge

- It's what you do when you're making commits at the same time as somebody else and you're **pulling** their commits so you have a consistent history.

```
- ● - ● - ● - ● (“master”)
    \ - - ● - ● - ● - ● (“mike-experiment”)
```

Merge

- It's what you do when you're making commits at the same time as somebody else and you're **pulling** their commits so you have a consistent history.



Merge

- Work on the project at the same time as someone else? OK!
- Add new files that weren't there before? OK!
- Rename or delete files? Usually OK!
- Edit the same file as someone else? Usually OK!

Merge

- Work on the project at the same time as someone else? OK!
- Add new files that weren't there before? OK!
- Rename or delete files? Usually OK!
- Edit the same file as someone else? Usually OK!

- When merging commits, the computer knows how to combine the changes as long as they're not ambiguous – like two people editing the same line.

Merge Conflict

- This happens if:
  - You edited the <h1> tag at the top of an HTML file and committed it
  - But someone else already made a different change to the same tag of the same file and committed it and pushed it.
  - And you're trying to merge what they've done

- Don't worry! Git will warn you, and show you both versions before it continues merging.

Workflow

- Do some stuff
- Commit
- Do more stuff
- Commit

Workflow

- Do some stuff
- Commit
- Do more stuff
- Commit

- Wait: all your work is only stored on your computer!

Workflow

- The workflow is like packing things into boxes when moving.
- When you **commit** (packed & labeled a box), the contents haven't gone anywhere just yet.
  - But you can always unpack a specific box if you need something!
- **Push** sends the boxes elsewhere.
- **Pull** receives boxes from elsewhere.

Workflow

- Do some stuff
- Commit
- Pull / Merge (if necessary) / Push
- Do more stuff
- Commit
- Pull / Merge (if necessary) / Push

Tips: Always Be Committing

- As long as you make a commit, you will always be able to recover that state of the repository – even if files are later merged or deleted.

Tips: Branches & forks help keep things in order.

- We usually work directly on the main branch when doing little after-the-fact updates, but inside individual forks or branches when we're building things out and actively making lots of commits

- Branches:
  - “master” branch, “mike” branch, “lena” branch, “sisi” branch
- Forks:
  - "propublica/master", "mtigas/master", "ashaw/master"