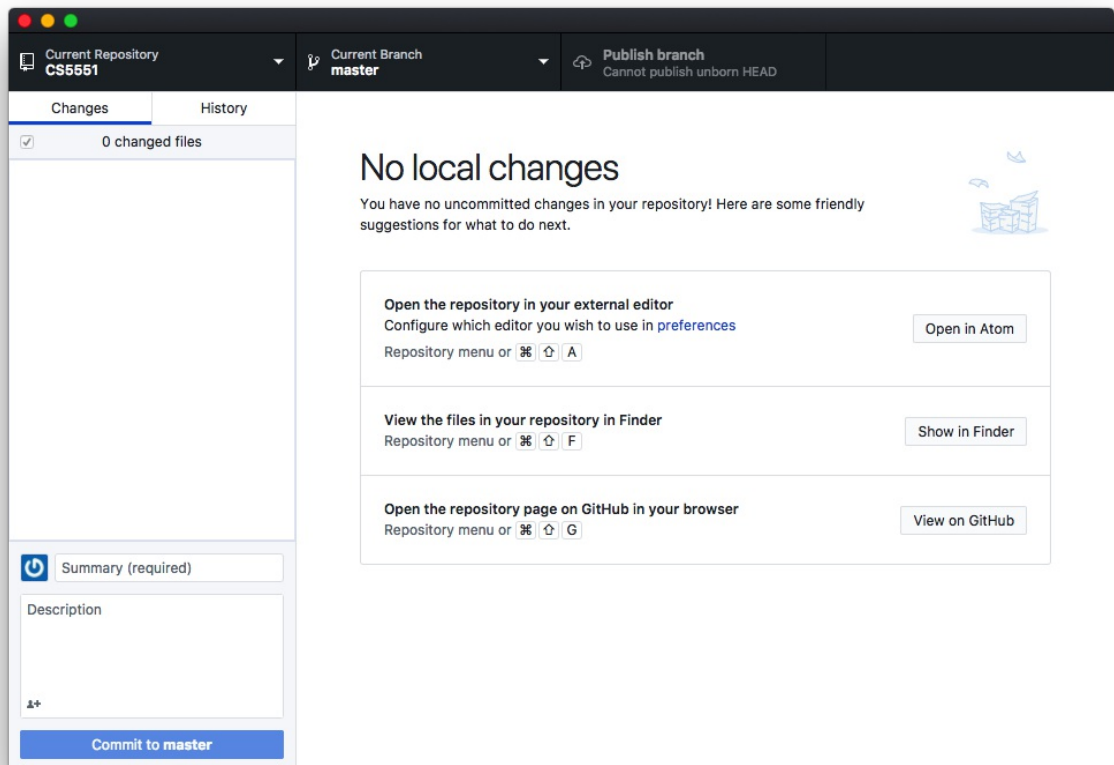


Screenshot

GitHub desktop

- Attribute commits with collaborators easily
- Checkout branches with pull requests and view CI statuses
- Syntax highlighted diffs



Gitshell

- Frictionless Context Switching. Create a branch to try out an idea, commit a few times, switch back to where you branched from, apply a patch, switch back to where you are experimenting, and merge it in.

- **Role-Based Codelines.** Have a branch that always contains only what goes to production, another that you merge work into for testing, and several smaller ones for day to day work.
- **Feature Based Workflow.** Create new branches for each new feature you're working on so you can seamlessly switch back and forth between them, then delete each branch when that feature gets merged into your main line.
- **Disposable Experimentation.** Create a branch to experiment in, realize it's not going to work, and just delete it - abandoning the work—with nobody else ever seeing it (even if you've pushed other branches in the meantime).

```

=>> git
Bash completion has been installed to:
  /usr/local/etc/bash_completion.d

zsh completions and functions have been installed to:
  /usr/local/share/zsh/site-functions

Emacs lisp files have been installed to:
  /usr/local/share/emacs/site-lisp/git
Hansons-Air:~ hanson$ git --version
git version 2.28.1
Hansons-Air:~ hanson$ git
[-bash: git: command not found
Hansons-Air:~ hanson$ git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
      [--exec-path<path>] [--html-path] [--man-path] [--info-path]
      [-p | --paginate] [-P | --no-pager] [--no-replace-objects] [--bare]
      [--git-dir<path>] [--work-tree<path>] [--namespace<name>]
      <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add        Add file contents to the index
  mv         Move or rename a file, a directory, or a symlink
  reset      Reset current HEAD to the specified state
  rm         Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
  bisect     Use binary search to find the commit that introduced a bug
  grep       Print lines matching a pattern
  log        Show commit logs
  show       Show various types of objects
  status     Show the working tree status

grow, mark and tweak your common history
  branch     List, create, or delete branches
  checkout   Switch branches or restore working tree files
  commit     Record changes to the repository
  diff       Show changes between commits, commit and working tree, etc
  merge      Join two or more development histories together
  rebase     Reapply commits on top of another base tip
  tag        Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
  fetch      Download objects and refs from another repository
  pull       Fetch from and integrate with another repository or a local branch
  push       Update remote refs along with associated objects

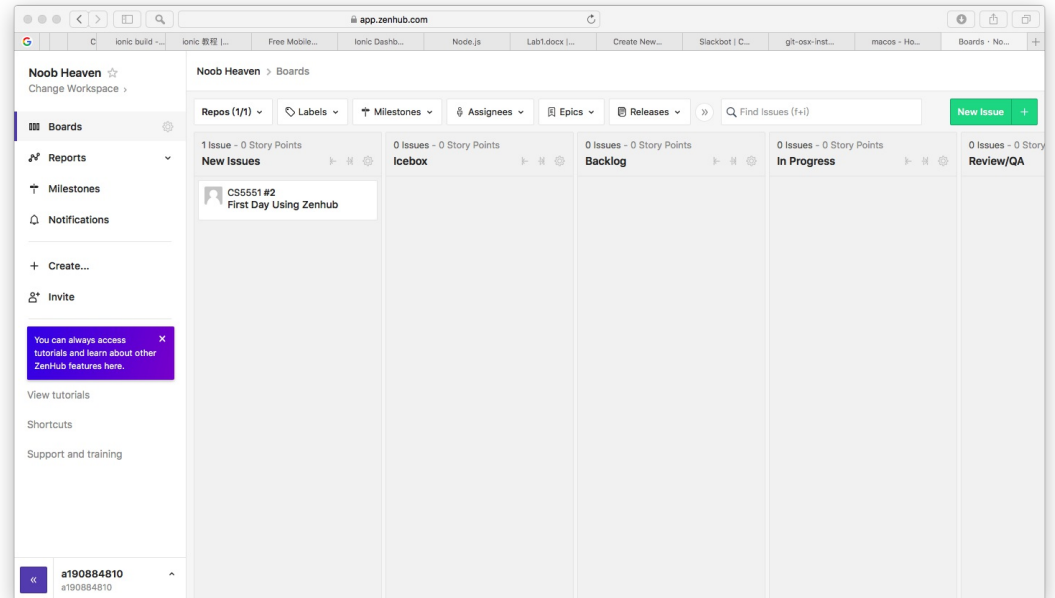
'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
Hansons-Air:~ hanson$

```

Zen Hub Plugin

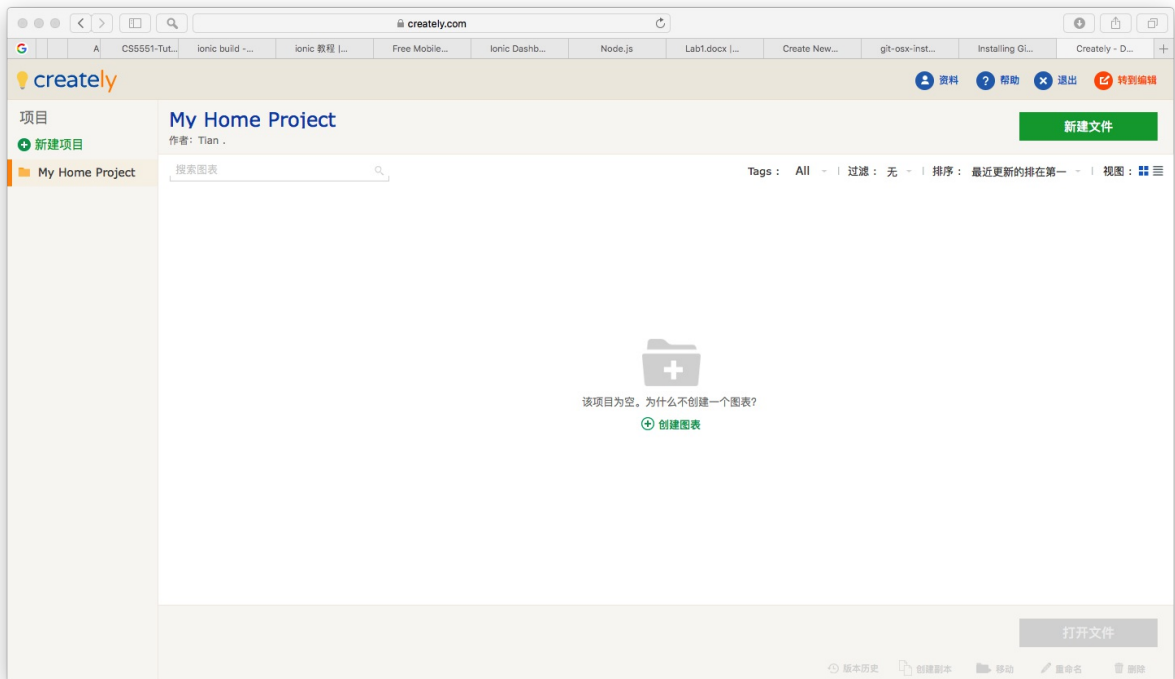
- **Data-driven project management**
- **ZenHub runs on GitHub data**, so your Burndown charts, Velocity tracking, and Release reports are always accurate.

- Plan, track, release
- - ZenHub makes it easy to break down complex projects into small, manageable tasks to align your team—and ship faster.



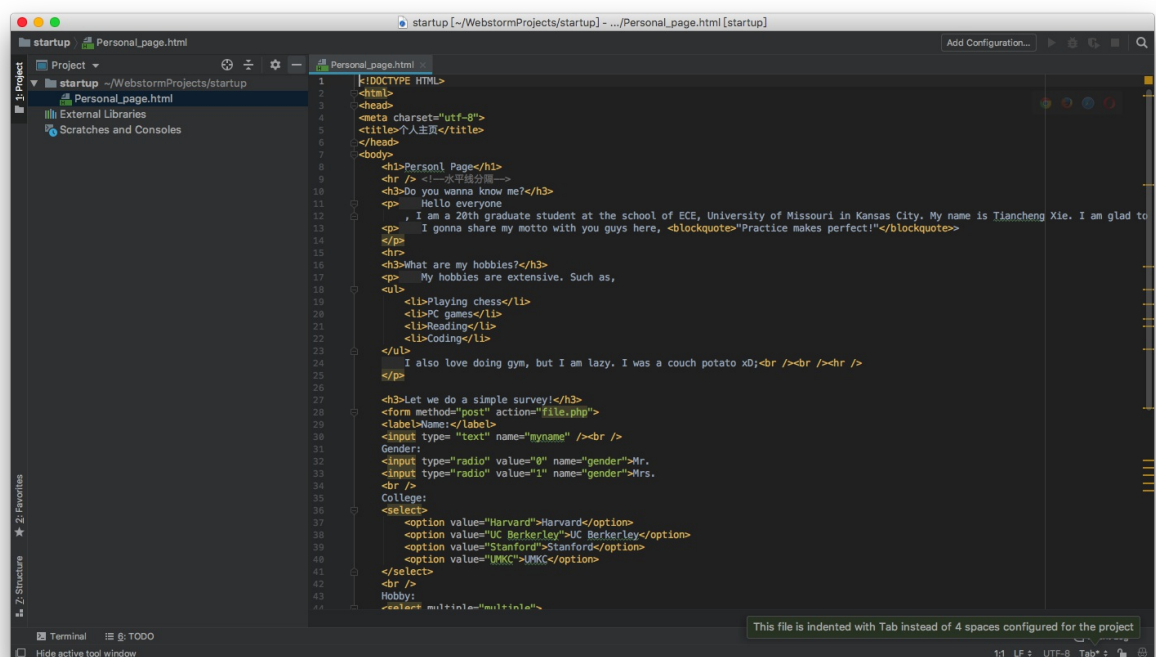
Creatly

- 1-click create
- Drag drop shapes
- Easy, preset connectors
- Aligned, sized and grouped
- Themes and color palette



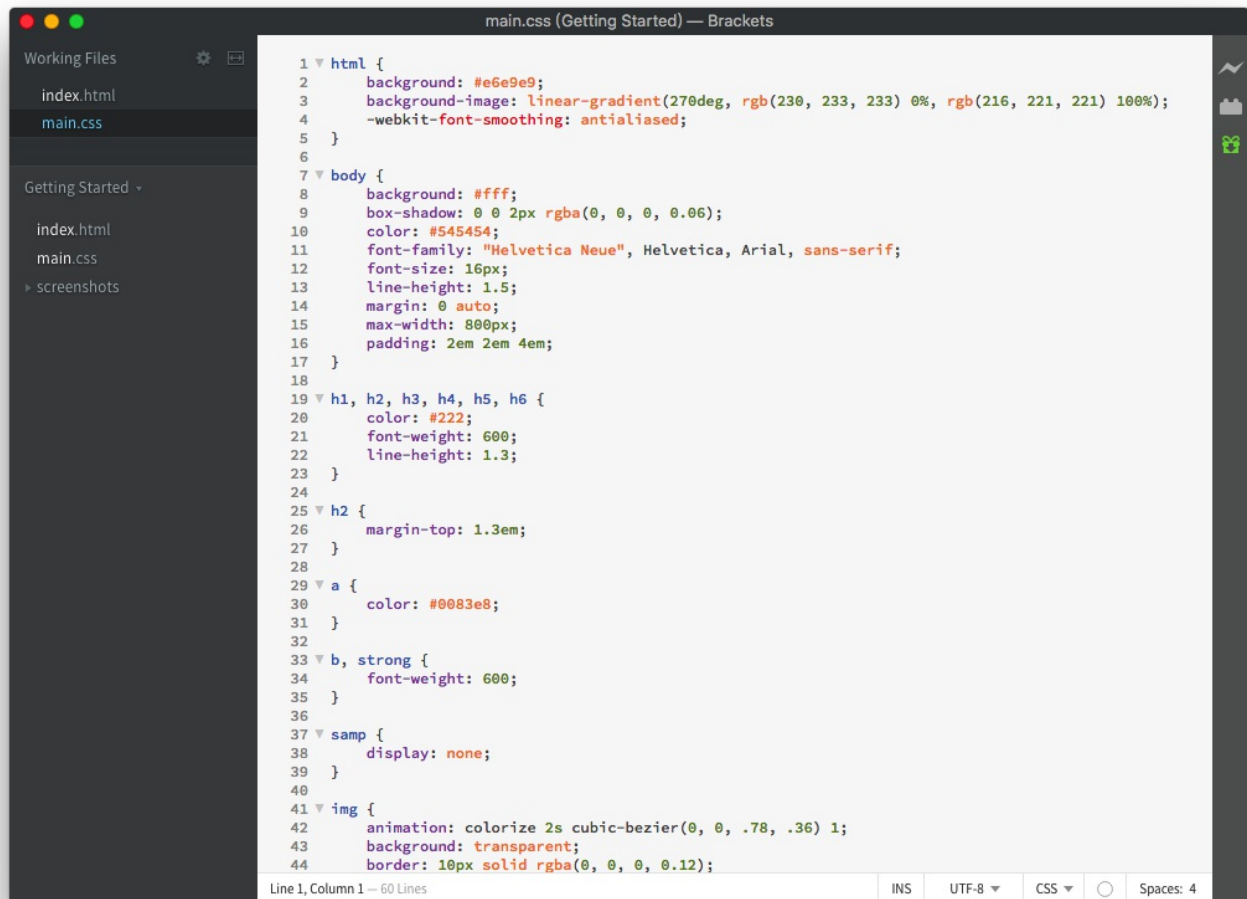
Web storm

- Intelligent coding assistance
- Modern frameworks
- Smart editor
- Navigation & Search



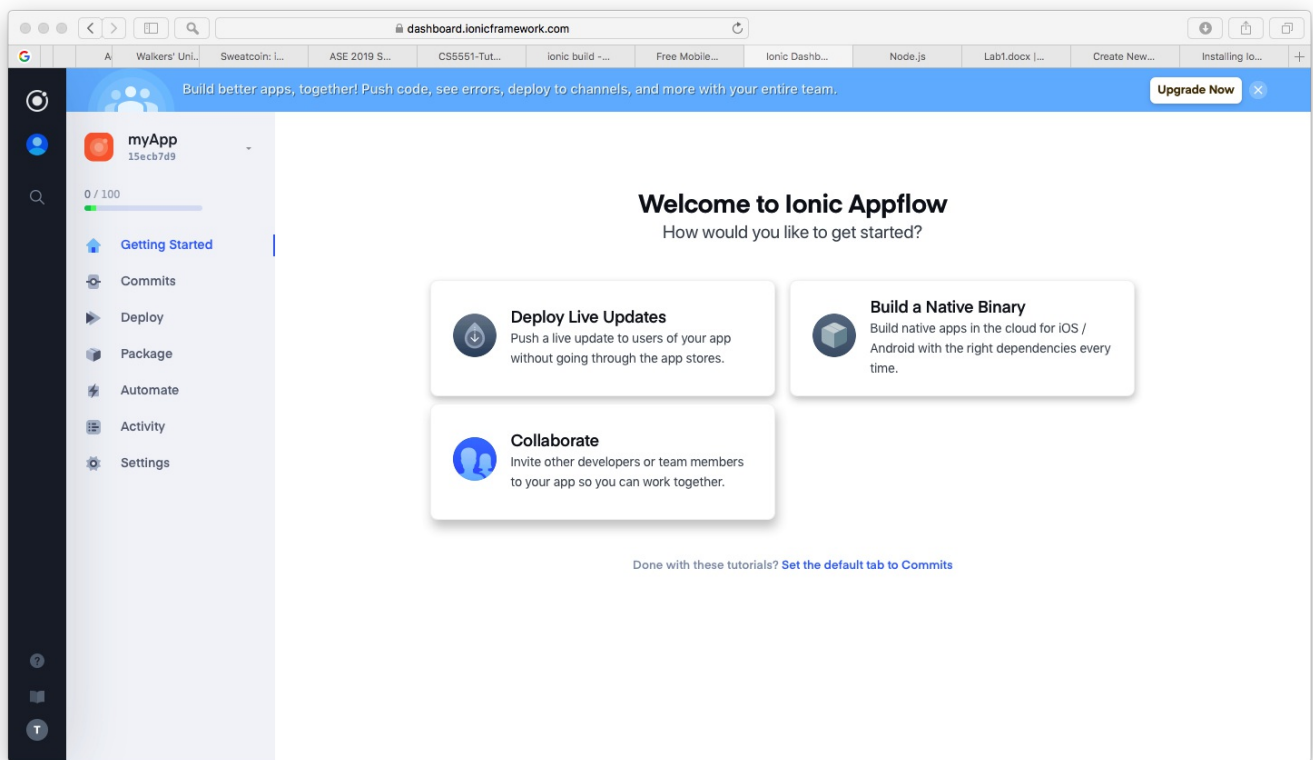
Brackets

Brackets is a lightweight, yet powerful, modern text editor. We blend visual tools into the editor so you get the right amount of help when you want it without getting in the way of your creative process. You'll enjoy writing code in Brackets.



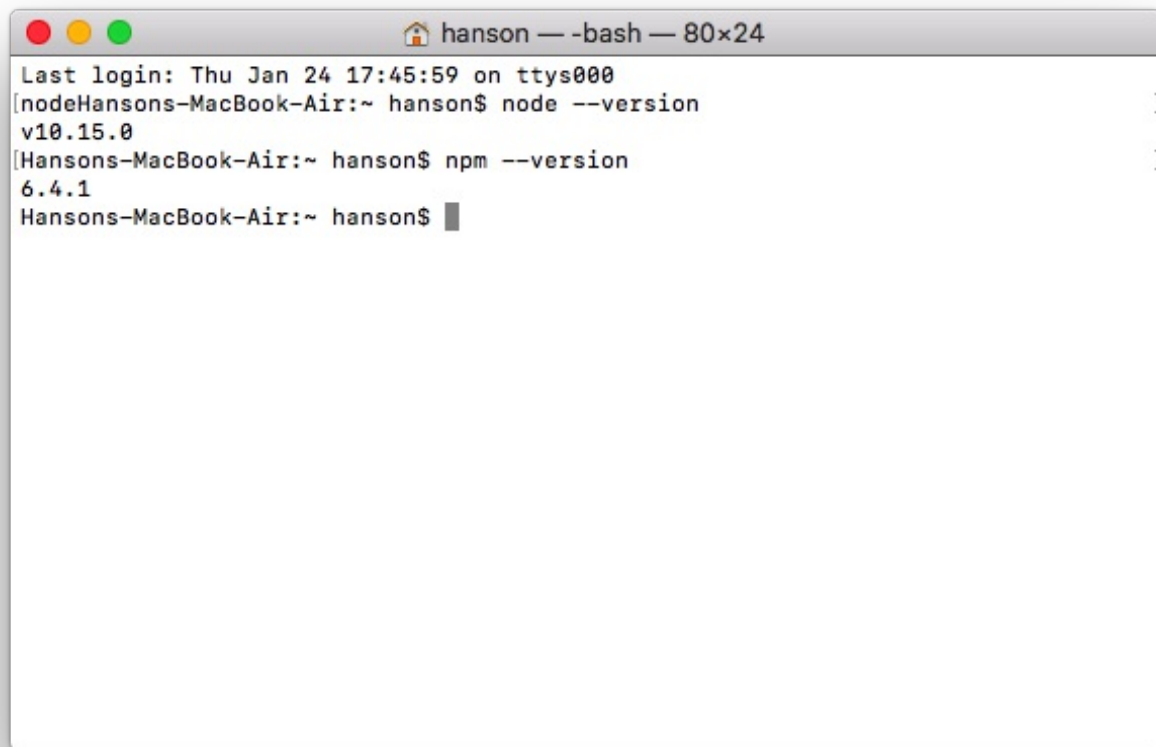
Ionic CLI

The Ionic command-line interface (CLI) is your go-to tool for developing Ionic apps.



Node and NPM

As an asynchronous event driven JavaScript runtime, Node is designed to build scalable network applications. In the following “hello world” example, many connections can be handled concurrently. Upon each connection the callback is fired, but if there is no work to be done, Node will sleep.



A macOS terminal window titled "hanson — -bash — 80x24". The window contains the following text:

```
Last login: Thu Jan 24 17:45:59 on ttys000
[nodeHansons-MacBook-Air:~ hanson$ node --version
v10.15.0
]
[Hansons-MacBook-Air:~ hanson$ npm --version
6.4.1
]
Hansons-MacBook-Air:~ hanson$ █
```