



Community Experience Distilled

GitHub Essentials

Unleash the power of collaborative workflow development using GitHub, one step at a time

Achilleas Pipinellis

[PACKT]
PUBLISHING

GitHub Essentials

Unleash the power of collaborative workflow development using GitHub, one step at a time

Achilleas Pipinellis



BIRMINGHAM - MUMBAI

GitHub Essentials

Copyright © 2015 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: September 2015

Production reference: 1280915

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.

ISBN 978-1-78355-371-6

www.packtpub.com

Credits

Author

Achilleas Pipinellis

Copy Editor

Trishya Hajare

Reviewer

Umesh Ram Sharma

Project Coordinator

Shweta H Birwatkar

Commissioning Editor

Dipika Gaonkar

Proofreader

Safis Editng

Acquisition Editor

Nikhil Karkal

Indexer

Hemangini Bari

Content Development Editor

Sumeet Sawant

Production Coordinator

Nitesh Thakur

Technical Editor

Saurabh Malhotra

Cover Work

Nitesh Thakur

About the Author

Achilleas Pipinellis is an open source enthusiast and tries to get involved in as many projects as possible. He was introduced to Linux almost 10 years ago and hasn't looked back ever since. His distribution of choice is Arch Linux, a lightweight and flexible system that adheres to the KISS philosophy.

He is currently working as a system administrator and likes to try new technologies, especially those that require some special deployment. He also enjoys writing technical guides and articles that help people learn about new technologies. He strongly believes that comprehensive documentation is essential to a project's growth and recognition.

In his free time he practices Aikido and enjoys going to conferences that promote the open source movement.

About the Reviewer

Umesh Ram Sharma has more than 6 years of experience in the architecture, design, and development of scalable and distributed cloud-based applications. He has a master's degree in computer science and information technology and is also an expert in the practical and technical implementation of various offerings of J2EE stack, Hibernate, and Spring Stack.

He is currently working as a senior software engineer, with a growing interest in the DevOps area. He handles product infrastructure on AWS cloud and develops expertise around automated deployments. He has demonstrated great value by implementing deployment, configuration management with puppet and various technologies, such as AWS Cloud, J2EE, MySQL, MongoDB, memcache, Apache Tomcat, and Hazelcast.

www.PacktPub.com

Support files, eBooks, discount offers, and more

For support files and downloads related to your book, please visit www.PacktPub.com.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.PacktPub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at service@packtpub.com for more details.

At www.PacktPub.com, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



<https://www2.packtpub.com/books/subscription/packtlib>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can search, access, and read Packt's entire library of books.

Why subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print, and bookmark content
- On demand and accessible via a web browser

Free access for Packt account holders

If you have an account with Packt at www.PacktPub.com, you can use this to access PacktLib today and view 9 entirely free books. Simply use your login credentials for immediate access.

Table of Contents

Preface	vii
Chapter 1: Brief Repository Overview and Usage of the Issue Tracker	1
Exploring the repository's main page	1
Creating a new repository	2
The commits page and a comparison with the git log command	4
The branches page and a comparison with the git branch command	7
The Raw, Blame, and History buttons	9
The Watch, Star, and Fork buttons	11
Changing the description and URL	12
Learning how to use the powerful benefits of the issue tracker	13
Creating a new issue	14
Assigning issues to users	18
Labels	20
Why labels are a great asset to UX	20
Creating new label names and setting different colors	20
Using labels to group issues	24
Milestones	24
Why milestones are a great help when working with code versioning	24
Creating a new milestone	25
Adding issues to milestones	26
Using milestones to see which issues are resolved or are yet to be resolved	27
Tips and tricks	27
Learning about the README file	27
Navigating easily with keyboard shortcuts	28
Summary	28

Chapter 2: Using the Wiki and Managing Code Versioning	29
Using the wiki	29
Why wikis are a nice place to document your project	30
Create a new wiki page	30
Deleting a page	32
A Markdown-powered wiki – an introduction to Markdown	33
How to add a sidebar and a footer to your wiki	35
Watching a wiki page's commit history and reverting to a previous state if needed	38
Managing code versioning	41
Creating a release	41
Editing a release	45
Pushing a tag from the command line	46
Marking as prerelease	46
Making a draft of a release	48
Uploading your own files	48
Tips and tricks	49
Subscribing to new releases via atom feed	50
Editing the wiki locally	50
Installing gollum	50
Cloning the wiki and see the preview in your browser	50
Making changes locally and pushing to GitHub	51
Summary	52
Chapter 3: Managing Organizations and Teams	53
The difference between users and organizations	53
Organization roles and repository permission levels	54
Creating an organization	55
Global member privileges	58
Repositories	60
Teams – a great way to grant selective access to your organization projects	62
Creating a team	62
Inviting people	65
Accepting an invitation	68
Team members permissions	69
Request to join a team	71
Step 1 – as a user	71
Step 2 – as a user	71
Step 3 – as an owner or team maintainer	72
Adding repositories to a team	72

The People tab	74
Managing access levels	76
Difference between Members and Outside collaborators	79
Demoting to an outside collaborator	81
Invite members	82
Organization settings	83
Profile	84
Team privacy	85
The third-party access	86
Audit log	86
Tips and tricks	88
How to transfer a project to an organization's namespace	88
How to convert a user account into an organization	91
Mention teams	94
Organization feed only in dashboard	95
Summary	96
Chapter 4: Collaboration Using the GitHub Workflow	97
Learn about pull requests	97
Why pull requests are a powerful asset to work with	97
The connection between branches and pull requests	98
Create branches directly in a project – the shared repository model	98
Create branches in your fork – the fork and pull model	98
How to create and submit a pull request	99
Use the Compare & pull request button	99
Use the compare function directly	102
Use the GitHub web editor	103
Submit a pull request	107
Peer review and inline comments	108
The layout of a pull request	109
Inline comments	113
Pull requests overview	116
Correct mistakes and re-push to branch	117
Merge the pull request	118
Remove/restore a branch after the pull request is merged	118
Revert a pull request	119
Tips and tricks	119
Close issues via commit messages	119
Task lists in pull requests	121
Downloading the diff of pull requests	123
A global list of your open pull requests	123

Adding a LICENSE file using the web editor	123
Creating new directories using the web editor	124
Summary	124
Chapter 5: GitHub Pages and Web Analytics	125
GitHub Pages	125
User, organization, and project pages	126
Creating a user or an organization page	127
Creating a project page manually	127
Creating a project page with GitHub page generator	128
Updating a project page with GitHub page generator	132
Using a custom domain	132
How to customize your page using Jekyll	133
Installing Jekyll	133
Introduction to Jekyll	135
Read more about Jekyll	138
Web analytics	139
Graphs	139
Contributors – additions/deletions	140
See a repository's traffic – visitors, clones, and popular content	142
Commits over time	143
Frequency of updates	144
Network	144
Members	146
Pulse	146
Tips and tricks	147
Making use of pages metadata with Jekyll	148
Summary	150
Chapter 6: Exploring the User and Repository Settings	151
User settings	151
Profile	152
Setting up multiple e-mails	153
Managing your SSH keys	154
Setting up two-factor authentication	156
Repository settings	157
Changing the default branch that appears in repository's main page	158
Enabling/disabling the wiki	159
Enabling/disabling the issue tracker	159

Adding collaborators	160
Transferring ownership – user to organization	161
Deleting a repository	162
Tips and tricks	162
Finding the size of your repositories	162
Fine-tuning e-mail notifications	163
Summary	163
Index	165

Preface

GitHub is the leading code-hosting platform with literally millions of open source projects having their code hosted on it. In conjunction with Git, it provides the means for a productive development workflow and is the preferred tool among developers. Starting with the basics of creating a repository, you will then learn how to manage the issue tracker, where your project can be discussed. Continuing our journey, we will explore how to use the wiki and write rich documentation that will accompany your project. Organization and team management will be the next stop and then onto the pull requests feature that made GitHub so well known. Next, we will focus on creating simple web pages hosted on GitHub and lastly we explore the settings that are configurable for a user and a repository.

What this book covers

Chapter 1, Brief Repository Overview and Usage of the Issue Tracker, explains some of the main features GitHub provides and what you can make out of them. The issue tracker is the heart of communication between a project's developers and/or users. Consider it as a notepad dedicated to each repository where you track bugs, reports, feature requests, and anything else that can be written down. GitHub has implemented many other features that sit on top of the issue tracker, such as labels and milestones, which provide the ability to better visualize and categorize all the issues.

Chapter 2, Using the Wiki and Managing Code Versioning, helps you learn how to create, edit, and maintain a wiki by providing a home for your documentation that will complement your project. You will also learn how to create a new release out of an existing branch or tag, accompanied with optional release notes. In this way, the end user can understand the changes from any previous versions.

Chapter 3, Managing Organizations and Teams, helps you learn how to create and manage the organizations that you are the owner of. You will also learn how to create teams, add users to them, and assign different access levels according to your needs.

Chapter 4, Collaboration Using the GitHub Workflow, helps you learn how to work with branches and pull requests, the most powerful features of GitHub.

Chapter 5, GitHub Pages and Web Analytics, helps you learn how to build web pages around your project, hosted exclusively on GitHub. You have the ability to make static web pages using HTML, CSS, and JavaScript.

Chapter 6, Exploring the User and Repository Settings, explores the most common and essential settings of a user and a repository. As a user, there is a lot of information you can set up in your user settings page, such as associating more than one e-mail to your account, adding multiple SSH keys, or setting up two-factor authentication. Similarly, some functionalities of a repository can be set up via its settings page. For example, you can enable or disable the wiki pages and grant write access to the public or completely disable the issue tracker.

What you need for this book

For this book, you'll just need Git; any version will do.

Who this book is for

This book is intended for experienced or novice developers with a basic knowledge of Git. If you ever wanted to learn how big projects such as Twitter, Google, or even GitHub, collaborate on code, then this book is for you.

Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: "Let's first create a README file and push it to GitHub in order to explore the commits page."


A block of code is set as follows:

```
echo "# github-essentials" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin git@github.com:<username>/<repository>.git
git push -u origin master
```

Any command-line input or output is written as follows:

```
sudo gem install bundler
```

New terms and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "The **Network** graph shows the branch history of the main repository as well as its forks."

[ Warnings or important notes appear in a box like this.]

[ Tips and tricks appear like this.]

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book – what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to feedback@packtpub.com, and mention the book title via the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on www.packtpub.com/authors.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Downloading the example code

You can download the example code files for all Packt books you have purchased from your account at <http://www.packtpub.com>. If you purchased this book elsewhere, you can visit <http://www.packtpub.com/support> and register to have the files e-mailed directly to you.

Downloading the color images of this book

We also provide you a PDF file that has color images of the screenshots/diagrams used in this book. The color images will help you better understand the changes in the output. You can download this file from: http://www.packtpub.com/sites/default/files/downloads/37160T_ColorImages.pdf.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/submit-errata>, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded on our website, or added to any list of existing errata, under the Errata section of that title. Any existing errata can be viewed by selecting your title from <http://www.packtpub.com/support>.

Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at copyright@packtpub.com with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

Questions

You can contact us at questions@packtpub.com if you are having a problem with any aspect of the book, and we will do our best to address it.

1

Brief Repository Overview and Usage of the Issue Tracker

The landing page of a project on GitHub depicts the contents of a person's local Git repository. Apart from the tree-like structure of the files, GitHub also provides some additional features that bring the most well-known and frequently used Git commands to your browser. Among others, these include the branches, commits, and tags of your repository.

In addition to the features mentioned above, GitHub also provides an issue tracker for each repository. This is where the discussions take place, bugs are tracked and reported, features are requested, and pretty much anything else that is relevant to the project is discussed.

GitHub has also implemented many other features that sit on top of the issue tracker, such as labels and milestones that provide the ability for better visualization and categorization of all the issues. We will explore all the features extensively, so don't worry if you aren't familiar with these terms yet.



Project and repository, although not the same thing, will be considered to have equal meaning and will be used interchangeably throughout this book.

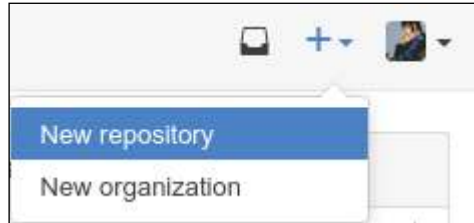
Exploring the repository's main page

The main page of a repository is the place where people spend most of their time when visiting a project. In this section, you will learn how to create a repository and then we will explore the vast features of GitHub that bring Git's command line to your browser.

Creating a new repository

Assuming you have already signed up in GitHub through <https://github.com/join>, we will see how to create a new repository that will host your code and explore the main repository's page.

Navigate to the top-right, click on the little cross beside your username, and choose **New repository**, as shown in the following screenshot:



Fill in a name under **Repository name**, which will ultimately form the URL under which your repository will be. This is the minimal action you need to perform in order to create a repository.

[ All the repositories on GitHub have the following URL scheme:
`https://github.com/<username>/<repository_name>`]

You can also provide a description so that people can tell with a glance what this is all about. Next option is whether your repository will be public or private. Generally, you go with public, unless you do not want your files to be seen by everybody. The private repos come with a price, though.

The very next thing GitHub provides is the ability to create the repository with a `README` file. Readme files usually include comprehensive information about the project you are hosting under your repository, such as installation guides, build and usage instructions, as well as guidelines on how one can contribute. You can always add a `README` file later, so leave this option unchecked for the time being.

Another nice feature is the ability to choose and include a `gitignore` file upon creation. You can choose from a collection of the useful `.gitignore` templates taken from <https://github.com/github/gitignore>.

Ultimately, the code that you will host on GitHub will be able to be forked and reused by third parties. If you are freshly starting a new repository, you can choose a license to include upon creation. Again, this is optional and you can always manually add a license file later.

Let's hit the **Create repository** button and finish the repository creation. Here's what it looks like so far:

The screenshot shows the GitHub repository creation interface for the repository 'axilcas / github-essentials'. It provides several options for how to set up the repository:

- Quick setup — If you've done this kind of thing before:**
 - Option 1: `or HTTPS SSH git@github.com:axilcas/github-essentials.git` (with a copy icon)
 - Text: "We recommend every repository include a README, LICENSE, and .gitignore"
 - Option 2: "... or create a new repository on the command line"
 - Code block:

```
mkdir "github-essentials" && cd $_
git init
git add README.md
git commit -m "First commit"
git remote add origin git@github.com:axilcas/github-essentials.git
git push -u origin master
```
 - Option 3: "... or push an existing repository from the command line"
 - Code block:

```
git remote add origin git@github.com:axilcas/github-essentials.git
git push -u origin master
```
 - Option 4: "... or import code from another repository"
 - Text: "You can import an existing repository from a Subversion, Mercurial, or CVS page"
 - Button: `import code`

You can see that GitHub provides useful information on what to do next. If you already have an existing Git repository locally on your computer, you can push its code to GitHub or start fresh by following the instructions.



Since we will be working from the command line later, it is highly recommended to generate an SSH key to use with your GitHub account. Follow the guide at <https://help.github.com/articles/generating-ssh-keys/>. Also, make sure to properly configure your Git username and email settings; for more information, see <https://help.github.com/articles/setting-your-username-in-git/> and <https://help.github.com/articles/setting-your-email-in-git/>.

Congratulations on creating your first repository!

The next goal is to explore the repository's main page. This is the page you see when you navigate to `https://github.com/<username>/<repository>`, where you see the following:

- **<username>**: This is the username you registered with (found at the top-right corner)
- **<repository>**: This is the **Repository name** you filled in the previous steps.

The commits page and a comparison with the git log command

GitHub has a nice web UI that many common `git` commands can be presented to.

Let's first create a `README` file and push it to GitHub in order to explore the commits page:

1. Create the directory that will hold your code and `cd` into it:

```
mkdir -p ~/github-essentials
cd $_
```

2. Then, follow GitHub's instructions on a new project creation:

```
echo "# github-essentials" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin git@github.com:<username>/<repository>.git
git push -u origin master
```

Note that I use the Git protocol (`git@github.com`) that uses SSH underneath so I don't have to type my username and password each time. See the previous note on how to achieve this.



The directory name (in our example `github-essentials`) could be totally different from the repository name you entered upon creation. It is the remote URL you set with `git remote add` that must match with the repository URL GitHub provides.

Every time you add more commits, their total number will also appear on the project's main page. In the preceding steps, we did our first commit, so the count is set to one and hence the **1 commit** option is shown in the following screenshot:



Click on the highlighted link as shown in the preceding screenshot to enter the commits page.

From here, you can browse the list of commits (so far we got only one) and visualize the output of `git log`. Let's compare those two commits:

Type `git log` in your local repository; the output should be similar to this:

```
commit 304b13bd7100e213c4e5e291e190b0d0e3b06602
Author: Achilleas Pipinellis <achilleas@achilleas.com>
Date:   Thu Apr 7 23:26:40 2016 -0300
```

First commit.

Now, head over the commits page on GitHub. Here, you can see the same information depicted in a nice interface:



sample content of GitHub Essentials

- [read online Kant's Dog: On Borges, Philosophy, and the Time of Translation \(SUNY Series in Latin American and Iberian Thought and Culture\)](#)
- [Atlas of Cone Beam Imaging for Dental Applications pdf, azw \(kindle\), epub, doc, mobi](#)
- [download The Essential Reader's Companion \(Star Wars\)](#)
- [Reflections on Violence online](#)
- [**click Buffalo Cake and Indian Pudding \(Penguin Great Food\)**](#)
- [Theodor Adorno \(Routledge Critical Thinkers\) book](#)

- <http://conexdx.com/library/Kant-s-Dog--On-Borges--Philosophy--and-the-Time-of-Translation--SUNY-Series-in-Latin-American-and-Iberian-Thoug>
- <http://www.celebritychat.in/?ebooks/Chosen-Prey--Lucas-Davenport--Book-12-.pdf>
- <http://fortune-touko.com/library/The-Essential-Reader-s-Companion--Star-Wars-.pdf>
- <http://korplast.gr/lib/Frommer-s-Stockholm-Day-By-Day--Frommer-s-Day-by-Day---Pocket-.pdf>
- <http://paulczajak.com/?library/Buffalo-Cake-and-Indian-Pudding--Penguin-Great-Food-.pdf>
- <http://jaythebody.com/freebooks/Professional-Microsoft-SQL-Server-2012-Analysis-Services-with-MDX-and-DAX--Wrox-Programmer-to-Programmer-.pdf>