

GitBox – User Manual

Git repository hosting inside a Dropbox folder

Abstract

As a developer I've come to appreciate code versioning a lot, especially the distributed version control systems. My favorite is git (thus the project, doh...), but it's a matter of personal preference. I use git for most of my personal/hobby projects, theses and work; but there are some practical problems:

I use multiple machines and multiple operating systems, thus synchronizing my work between them and creating/managing backups is a pain without some third party source repository. For work, a company versioning system will do, and for open source projects github will do. But for personal research projects another solution was needed (private git hosting is too expensive, period).

The idea came (not mine, Google it) from the file synchronization service called Dropbox. It keeps a "special" folder on the file system in sync between different operating systems and machines (i.e. whatever you place in one will be available on all); as well as keeping a backup in their cloud service. It's private and up to 10GB (through referrals) are free.

Thus the idea, that if we place a git repository inside the Dropbox folder, it will be available virtually anywhere, as well as backed up in case of a HDD failure. Whenever a commit is pushed into the repo, Dropbox would automatically sync it with the others. Instant private git hosting.

Although managing git repositories inside Dropbox aren't too hard (it's the same as if you'd manage it anywhere else remotely), you still have to remember a "handful" of commands as well as long paths. GitBox was born out of the necessity to make this process more user-friendly and automatic: creating, cloning and importing repositories should be one-liners.

Hope you like it; I'm open for suggestions, requests, bug reports and any feedback whatsoever. ☺

P.S. Less is more... I prefer stable products over large feature sets.

Péter Szilágyi

Table of Contents

Abstract.....	1
1 Introduction	3
2 Installation	3
2.1 Linux and *nix	3
2.2 Windows	3
3 Verification.....	4
4 Commands	4
4.1 gitbox	4
4.2 gitbox list.....	4
4.3 gitbox create <repo>	4
4.4 gitbox clone <repo>	5
4.5 gitbox import <repo>	5
5 Releases	5
6 License.....	5

1 Introduction

GitBox is a cross-platform git repository manager for hosting private personal git repos inside Dropbox folders. Thanks to the nature of Dropbox, features include automatic synchronization between multiple machines and operating systems, as well as online backups.

Currently GitBox is capable of creating git repositories inside Dropbox folders, cloning repositories residing inside these folders, and importing existing git repos into Dropbox folders.

What GitBox is not, is a collaboration tool between multiple developers. Sharing a git repository through Dropbox will have serious consequences as there is no way to do atomic syncs.

2 Installation

As a prerequisite, GitBox requires you to have a valid Dropbox installation on your machine. If you haven't already done so, in order to install Dropbox, you'll need to first register an account for free at the Dropbox website. (Consider registering through [my referral link](#) as support for this project, you'll also get an additional 250MB of storage space). After registration you can [download](#) and install their desktop client for various platforms.

After you installed, configured and are happy with your Dropbox, the next step is to download the GitBox package, which is a simple zip file available from the project's [github download page](#). You need not worry about operating systems, since there is a single, cross-platform GitBox bundle. After acquiring the distribution zip package, GitBox should be extracted **into** your active Dropbox folder (the root itself is a good choice since GitBox is already in its own folder). Make sure you're happy with the location before proceeding.

2.1 Linux and *nix

To finish the installation in Linux and other flavors of *nix, the setup.sh shell script in the setup folder should be executed.

```
> cd <gitbox root>/setup
> sh setup.sh
GitBox was successfully configured.
```

2.2 Windows

In order to finish installation under Windows, the GitBox root folder containing the executable batch file should be added to the user's path environmental variable.

For users of Windows XP Service Pack 2 and newer, Windows Vista and Windows 7, a setup script was also included in the bundle (<install root>\GitBox\setup\setup.bat), which will configure the path automatically. Simply run this batch file and you're ready to go.

At the moment, users of previous versions of Windows need to add GitBox manually to their path variable through the Control Panel -> System -> Advanced -> Environment Variables. The GitBox root folder should be appended to the path. Please note that only newly started programs will use the modified path.

3 Verification

In order to verify that GitBox was correctly installed and configured, switch to a random folder on your computer and execute "gitbox" (under *nix, do this from a shell).

At this point GitBox will check whether there is a valid git installation on the local computer and offer to install one if needed. After which a git bash console should be started up.

Should the gitbox script not be found, please go back and double check your paths and environmental variables.

4 Commands

This section contains an exhaustive list of all the commands currently supported by GitBox.

4.1 gitbox

In Windows checks for git and opens up a new bash console prepared for executing git (and other bash) commands.

In Linux this command simply ensures that git is available and prints its version.

4.2 gitbox list

Lists all the git repositories within the GitBox collection.

4.3 gitbox create <repo>

Creates a new empty git repository called <repo> inside the GitBox repository collection.

This command is used when a new repository/project is needed and there aren't any existing repositories which can be imported into GitBox.

4.4 gitbox clone <repo>

Clones a git repository named <repo> from the GitBox collection into the current folder, configures a default remote called "gitbox" for committing code back into GitBox through git.

At this point a simple "git push" will commit the changes into the GitBox repo.

4.5 gitbox import <repo>

Imports a git repository from the current folder into GitBox under the name <repo>. It also configures a remote called "gitbox" for the local repo.

To push changes from the imported repository into GitBox, the "git push gitbox" command is required.

5 Releases

- Version 0.1.0 Unreleased yet
 - Support for repository listing, creation, cloning and importing
 - Automatic git installation in OpenSuSE and Windows
 - Automatic GitBox configuration in *nix and Windows XP SP2+, Vista and Windows 7

6 License

Copyright © 2010 Péter Szilágyi <peter@karalabe.com>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.