8 августа 2011 в 12:20

Как начать работать с GitHub: быстрый старт

Git*



Распределенные системы контроля версий (DVCS) постепенно замещают собой централизованные. Если вы еще не используете одну из них — самое время попробовать.

В статье я постараюсь показать, как можно быстро начать экспериментировать с git, используя сайт github.com.

В статье не будут рассмотрены различия между разными DVCS. Также не будет детально рассматриваться работа с git, по этой теме есть множество хороших источников, которые я приведу в конце статьи.

Итак, сайт github.com позиционируется как веб-сервис хостинга проектов с использованием системы контроля версий git, а также как социальная сеть для разработчиков. Пользователи могут создавать неограниченное число репозиториев, для каждого из которых предоставляется wiki, система issue tracking-a, есть возможность проводить code review и многое другое. GitHub на данный момент является самым популярным сервисом такого рода, обогнав Sourceforge и Google Code.

Для open-souce проктов использование сайта бесплатно. При необходимости иметь приватные репозитории, есть возможность перейти на платный тарифный план:

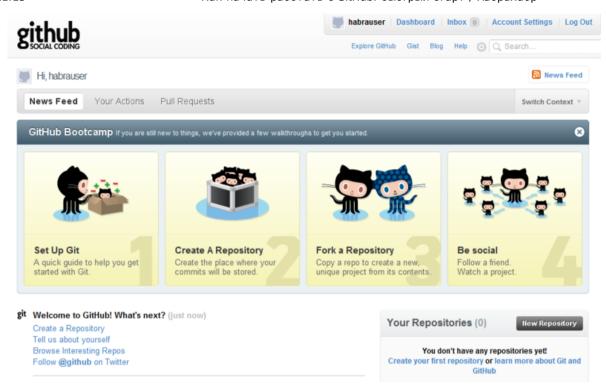


Business Plans



Начнем с регистрации. Идем по ссылке $\underline{github.com/signup/free}$ и вводим свои данные. После регистрации мы попадаем на Dashboard нашего аккаунта:

habrahabr.ru/post/125799/ 1/15



Сейчас у нас нет ни одного репозитория, и мы можем либо создать новый репозиторий, либо ответвиться (fork) от уже существующего чужого репозитория и вести собственную ветку разработки. Затем, при желании, свои изменения можно предложить автору исходного репозитория (Pull request).

Но для начала установим git и настроим его для работы с сайтом.

Если вы работаете в Windows, качаем и устанавливаем <u>msysgit</u>. Это консольная версия git для Windows (далее расказ будет вестись на примере этой OC).

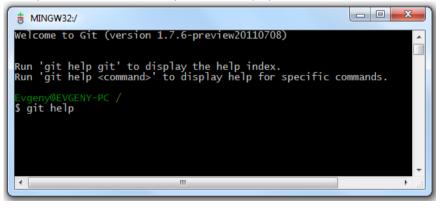
Инструкция для MacOS X (eng)

Инструкция для Linux (eng)

Проблем возникнуть не должно, просто везде жмем Next. После установки выбираем в контекстном меню Проводника Git Bash:

```
Git Add all files now
Git Commit Tool
Git History
Git Gui
Git Branch
```

или через Git Bash.Ink в папке с установленой программой:



Прописываем в консоли свои данные и настройки переносов строк:

```
git config --global user.name "ваше имя" git config --global user.email "ваша почта" git config --global core.autocrlf true git config --global core.safecrlf true
```

Кстати, рекомендую пройти неплохой <u>интерактивный курс</u> по использованию git из консоли. Курс проходится за несколько часов

habrahabr.ru/post/125799/ 2/15

и дает необходимые базовые навыки.

Для тех, кто предпочитает gui — для Windows существует несколько таких инструментов для работы с git. Два основных — это $\underline{SmartGit}$ (кроссплатформенный) и $\underline{TortoiseGit}$. Оба неплохие, и какой использовать — дело вкуса. Я опишу работу с $\underline{TortoiseGit}$. Для маков выбор giu тоже имеется.

- <u>официальный клиент от GitHub</u> на мой взгляд пока достаточно сыроват.
- GitX лично мне не приглянулся
- GitBox наиболее следует mac-way, очень рекомендую попробовать именно его

Качаем по ссылке code.google.com/p/tortoiseqit/downloads/list. При установке везде жмем Next.

Теперь возвращаемся к github и создадим новый репозиторий. Находясь на Dashboard, жмем New Repository (https://github.com/repositories/new), вводим данные и жмем Create Repository.

GitHub позволяет работать с репозиториями тремя способами: SSH, HTTP и Git Read-Only, соответственно предоставляя ссылки трех видов для нашего репозитория:

- 1. git@github.com:habrauser/Hello-world.git
- 2. habrauser@github.com/habrauser/Hello-world.git
- 3. git://github.com/habrauser/Hello-world.git



Для того, чтобы просто забрать репозиторий на локальную машину, достаточно внутреннего протокола git (третья ссылка). Это наиболее быстрый и эффективный способ, который обеспечивает анонимный доступ только для чтения.

Если же мы захотим внести изменения в репозиторий на github, нужно пользоваться HTTP или SSH. Работа по http никаких трудностей не вызывает, в нужный момент просто используется пароль учетной записи на github.

Чтобы использовать SSH, нам нужно создать специальную пару ключей: публичный и приватный. Публичный будет размещен в настройках аккаунта на github, а приватный сохранен на локальной машине.

Для генерации ключей, можно воспользоваться инструментом ssh-keygen, который идет в комплекте с git (описание этого способа можно почитать $\underline{\text{тут}}$). Мы же будем использовать PuTTY (а точнее небольшую программку puttygen, входящую в его состав). PuTTY — это такой клиент для удаленного доступа, в том числе и с использованием SSH.

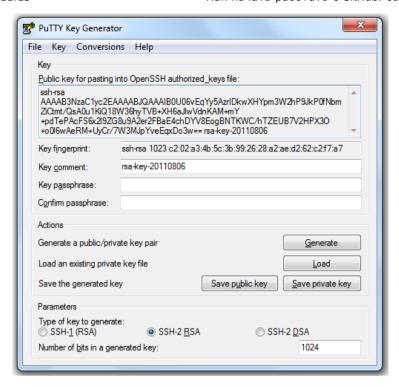
Качаем последнюю версию с официального сайта (http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html). Кстати, puttygen более старой версии (2007 год) идет в составе TortoiseGit.

После установки PuTTY, запускаем puttygen из папки с установленной программой:



Жмем Generate, двигаем некоторое время курсором мыши, для получения случайных данных, необходимых алгоритму

habrahabr.ru/post/125799/



Вводим пароль, защищающий наш приватный ключ в поле Key passphrase, вводим подтверждение, жмем Save private key, сохраняем.

Далее копируем публичный ключ в формате OpenSSH из текстовой области «Public key for pasting...» и идем в настройки нашего аккаунта на qithub (Account Settings) в раздел SSH Public Keys:

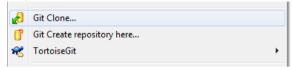


жмем Add another public Key, вставляем наш публичный ключ:

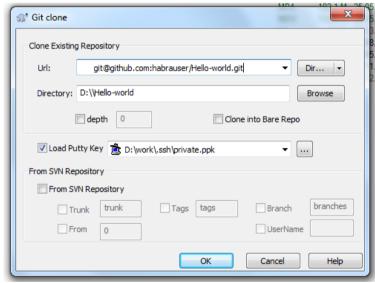
habrahabr.ru/post/125799/ 4/15



нажимаем Add key. Все, теперь мы готовы работать с github по ssh. Попробуем забрать наш пустой рерозиторий на локальную машину с использованием TortioшseGit. В контекстном меню проводника выбираем Git Clone...



В поле Url вставляем SSH-адрес нашего репозитория, в поле Load Putty Key указываем путь к нашему приватному ключу, жмем ОК.



Pageant запросит у нас пароль для приватного ключа (потом этого делать не потребуется)

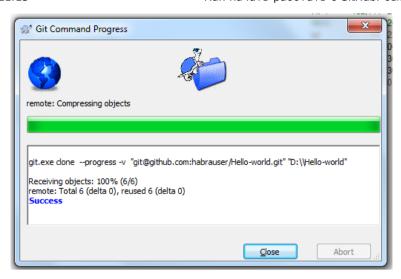


Pageant — это агент SSH-аутентификации в составе PuTTY, он позволяет управлять закрытыми ключами. В трее висит его значек:



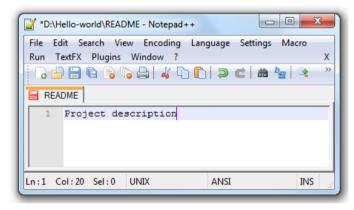
Репозиторий успешно склонирован на локальную машину

habrahabr.ru/post/125799/ 5/15

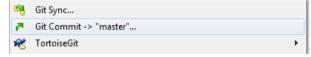


Теперь попробуем изменить локальный репозиторий и отправить изменения на github. Добавим в локальный репозиторий файл README (файл с именем README обрабатывается github специальным образом — его содержимое будет отображаться в качестве описания репозитория на соответствующей странице)

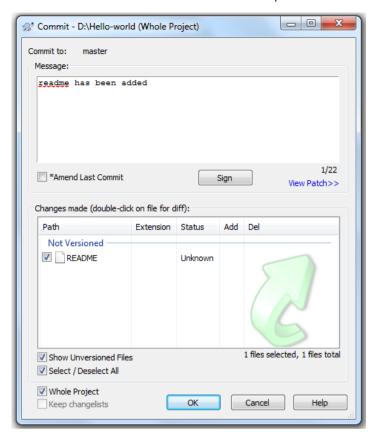




Закоммитим изменения в локальный репозиторий



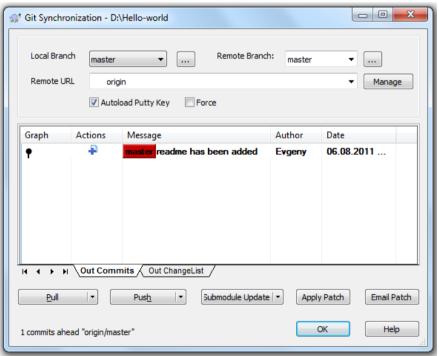
habrahabr.ru/post/125799/



и синхронизируем его с репозиторием на github:

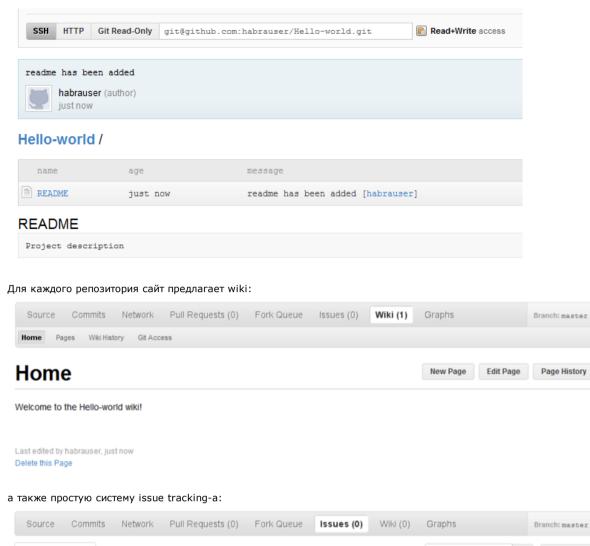


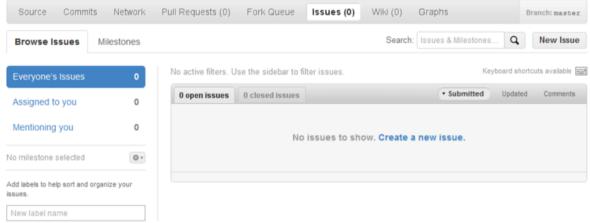
нажимаем Push



Теперь зайдя на страницу нашего репозитория мы увидим следующее:

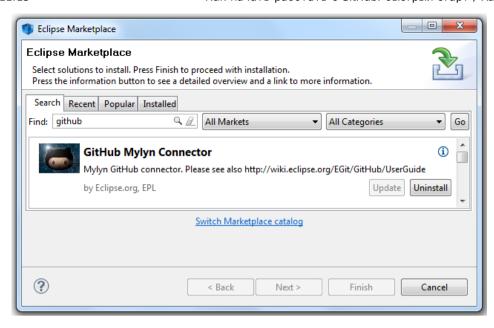
habrahabr.ru/post/125799/ 7/15



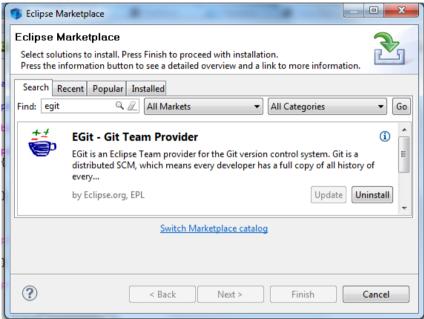


кстати, для тех, кто использует в работе Eclipce — есть соответствующий mylyn-коннектор для github:

habrahabr.ru/post/125799/



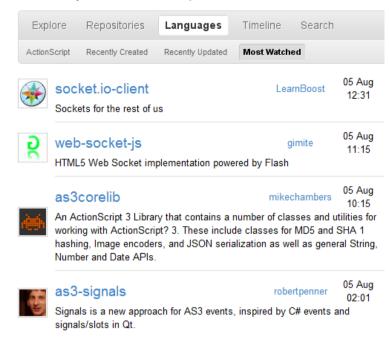
и плагин EGit:



По ссылке Explore GitHub открывается каталог репозиториев, в котором можно искать по множеству других критериев, в том числе по языкам программирования, популярности и т.п.

habrahabr.ru/post/125799/ 9/15

ActionScript most watched repositories



Резюмируя хочется сказать, что если вы начинающий разработчик, планирующий начать пользоваться системами контроля версий, или же более опытный и присматривающийся к распределенным VCS, но не знающий как начать, то есть смысл попробовать git, используя такой замечательный инструмент как github.com.

Полезные ссылки

Для работы c git:

code.google.com/p/msysgit/ git для windows www.syntevo.com/smartgit/index.html SmartGit code.google.com/p/tortoisegit/ TortoiseGit http://www.chiark.greenend.org.uk/~sqtatham/putty/ PuTTY

Про git на русском:

 $\frac{habrahabr.ru/blogs/Git/106912/}{sypauhas}$ «Удачная модель ветвления для git» — перевод хорошей англоязычной статьи $\frac{githowto.com/}{sypeuhas}$ интерактивный курс по работе с git из консоли

habrahabr.ru/blogs/Git/106912/ «Почему git» + обсуждение

habrahabr.ru/blogs/development/68341/ «Git для переходящих с SVN» + обсуждение

habrahabr.ru/blogs/Git/75990/ «Командная работа в git» + обсуждение

progit.org/book/ru/ русский перевод книги «Pro Git» (переведено не до конца)

habrahabr.ru/blogs/Git/123111/ инструкция-шпаргалка для начинающих

los-t.livejournal.com/tag/qit%20guts цикл постов «внутренности git»

 $\underline{\text{lib.custis.ru}/\%\text{D}0\%9B\%\text{D}0\%B8\%\text{D}0\%BB\%\text{D}1\%83\%\text{D}1\%81} \quad \%\text{D}0\%\text{A}2\%\text{D}0\%\text{B}E\%\text{D}1\%80\%\text{D}0\%\text{B}2\%\text{D}0\%\text{B}B\%\text{D}1\%8C\%\text{D}0\%\text{B}E\%\text{D}1\%80\%\text{D}0\%\text{B}B\%\text{D}1\%8C\%\text{D}0\%\text{B}B\%\text{D}1\%8C\%\text{D}0\%\text{B}E\%\text{D}1\%80\%\text{D}0\%\text{B}B\%\text{D}1\%8C\%\text{D}0\%\text{B}E\%\text{D}1\%80\%\text{D}0\%\text{B}B\%\text{D}1\%8C\%\text{D}0\%\text{B}E\%\text{D}1\%80\%\text{D}0\%\text{B}B\%\text{D}1\%80\%\text{D}0\%\text{B}B\%\text{D}1\%80\%\text{D}0\%\text{B}B\%\text{D}1\%80\%\text{D}0\%\text{B}B\%\text{D}1\%80\%\text{D}0\%\text{B}B\%\text{D}1\%80\%\text{D}0\%\text{B}B\%\text{D}1\%80\%\text{D}0\%\text{B}B\%\text{D}1\%80\%\text{D}0\%\text{B}B\%\text{D}1\%80\%\text{D}0\%\text{B}B\%\text{D}1\%80\%\text{D}0\%\text{B}B\%\text{D}1\%80\%\text{D}0\%\text{B}B\%\text{D}1\%80\%\text{D}0\%\text{B}B\%\text{D}1\%80\%\text{D}0\%\text{B}B\%\text{D}1\%80\%\text{D}0\%\text{B}B\%\text{D}1\%80\%\text{D}0\%\text{B}B\%\text{D}1\%80\%\text{D}0\%\text{B}B\%\text{D}1\%\text{B}0\%\text{D}1\%\text{D}1\%\text{B}0\%\text{D}1\%\text{B}0\%\text{D}1\%\text{B}0\%\text{D}1\%\text{B}0\%\text{D}1\%\text{B}0\%\text{D}1\%\text{D}1\%\text{D}1\%\text{B}0\%\text{D}1\%\text{D}$

Линус Торвальдс o git

habrahabr.ru/blogs/Git/80909/ книга «Волшебство git»

Про git на английском:

книги

- progit.org/book/ книга «Pro Git»
- <u>rutracker.org/forum/viewtopic.php?t=2808582</u> книга «Version Control with Git», 2009, O'Reilly
- <u>book.git-scm.com/</u> книга «Git Community Book»
- rutracker.org/forum/viewtopic.php?t=2808843 книга «Pragmatic Version Control Using Git», 2008, T. Swicegood
- rutracker.org/forum/viewtopic.php?t=3239579 книга «Pragmatic Guide to Git», 2010, Т. Swicegood. Описываемая версия git:
 1.7.2.1. Книга в формате двустраничных разворотов проблема/решение
- rutracker.org/forum/viewtopic.php?t=900767 книга «Git Internals»
- http://www-cs-students.stanford.edu/~blynn/gitmagic/index.html книга «Git Magic»

видео

- rutracker.org/forum/viewtopic.php?t=3520513 видеокурс, посвященный git. Довольно нескучный и с юмором.
- www.youtube.com/watch?v=8dhZ9BXQgc4 лекция о git от Randal Schwartz
- <u>excess.org/article/2008/07/ogre-git-tutorial/</u> видеотуториал Git The Basics Tutorial

другое

- gitref.org/ референс для изучения наиболее используемых команд
- <u>ftp.newartisans.com/pub/qit.from.bottom.up.pdf</u> инструкция по принципу «снизу вверх», т.е. от низкоуровнего устройства git к верхнеуровневым командам. Автор утверждает, что так информация лучше воспринимается
- <u>ndpsoftware.com/qit-cheatsheet.html</u> хороший cheatsheet
- whygitisbetterthanx.com/ преимущества git в сравнении с другими VCS
- devcheatsheet.com/tag/git/ подборка cheatsheets

git, github, TortoiuseGit, SmartGit, SSH, DVCS, VCS, контроль версий, Sourceforge, Google Code, octocat, PuTTY

+148 1212 FlashBurn ^{30,0} G+

КОММЕНТАРИИ (48) отслеживать новые:

в почте в трекере

Artouiros, 8 abrycta 2011 b 12:40 #

Под эклипсом использую EGit, очень удобно.

Mallen, 8 aBrycta 2011 в 13:16 # ↑ +1

А он симлинки уже умеет от файлов отличать?

<u>m1kola</u>, 8 августа 2011 в 19:12 # ↑

Да, удобно... Но вот только git submodules не поддерживает. Проекты с субмодулями приходится из консоли коммитить :(

FlashBurn, 8 августа 2011 в 12:52 #

кстати да, упомяну пожалуй. Спасибо

masterrr, 8 августа 2011 в 12:52 # +3

А еще школьникам, студентам, учителям могут бесплатно выделить micro план на год-два. (Так он стоит 7\$)

kost, 9 августа 2011 в 11:50 # ↑

Ссылку забыли: github.com/edu

Goder, 8 августа 2011 в 12:53 # +2

Кроме Эклипса, плагины для VCS есть и у Visual Studio (2008/2010), сам пользуюсь Mercurial и Git напрямую из студии, — очень удобно.

<u>acerv</u>, 8 августа 2011 в 13:00 #

кто нибудь знает, как заставить MINGW по-русски ввод понимать?

MashaVokh, 8 abrycta 2011 b 14:42 # ↑

Нужен именно Mingw (то есть линуксовое окружение) или просто возможность работать с Git из консоли?

Если второе, то может помощь <u>utf8-git-on-windows</u>. Там набор из msysgit+tortoisegit, приведенных к единому utf8-знаменателю, вместо обычного разброда и шатания в виде cp1251 в именах файлов, cp866 в консоли и ansi во внутренних скриптах msysgit.

tulskiy, 8 августа 2011 в 13:06 #

А зачем держать какую-то прогу если через ssh-keygen все само делается?

И еще, напишите кто-нибудь, пожалуйста, статью как правильно работать с форкнутыми проектами и пулл реквестами. Например, как правильно забирать изменения из апстрима и не включать в пулл реквест кучу мерж коммитов, или что делать если в апстриме поменялись файлы, которые в пулл реквесте — ребэйс с форсом или отдавать мерж коммит, или это ложится на плечи самого мэйнтейнера?

git push github brunch_name

git pull github

перед этим сделать git remote add github <u>ilia-maslakov@github.com/ilia-maslakov/mc.git</u>

вместо <u>ilia-maslakov</u>@github.com/ilia-maslakov/mc.git подставить урл доступа к репу.

habrahabr.ru/post/125799/ 11/15

и да, присоединяюсь к предложению добавить эту инфу к топику, чтобы проще было новичку форкнуть готовый проект с полной историей

Envek, 9 августа 2011 в 18:00 #

+6

Самый основной метод:

- 1. Форкаете репозиторий github.com/user1/project, получаете github.com/you/project
- 2. Уже форкнутый репозиторий сливаете себе на комп:

git clone git@github.com:/you/project project

3. Создаёте отсылку на «родительский» репозитарий, например как на upstream

git remote add git@github.com:/userl/project upstream, чтобы потом сливать изменения с «родителя» себе git pull upstream master

4. Создаёте новую ветку

git checkout -b feature

5. Работаете, делаете коммиты, в случае необходимости отслеживания изменений в «родителе», сливаете изменения с него и вливаете в свою ветку таким образом:

git checkout master

git pull upstream master

git checkout feature

git merge master

6. Когда работу сделали, заливаете изменения в свой github-репозиторий в свою ветку:

git push origin feature

- 7. Теперь идёте на гитхаб, в свой репозиторий и жмёте вверху кнопочку «Pull request»
- 8. Слева выбираете в какую ветку будут вливаться изменения в родительском репозитории, справа какие изменения будут браться с вашего репозитория. По примеру: справа project/master, слева project/feature.

ВЫЖНО: Договоритесь с владельцем «родительского» репозитория, в какую ветку будете вливать изменения (он может написать это в README)

9. Заполняете название и описание (название потом попадёт в описание мёрдж-коммита и станет достоянием общественности, учтите это).

10. Нажимаете Send Pull Request

Вуаля, вы его отправили. Владелец рассмотрит ваши изменения и, возможно, их примет и вольёт к себе.

На практике, лучше перед посылкой пулл-реквестов, вручную синхронизироваться с веткой, в которую будете посылать изменения, чтобы у владельца merge прошёл гладко (больше шансов, что пулл примут ;-))

He забудьте потом сделать git pull upstream master, чтобы увидеть изменения у себя.

Надеюсь, это мини-руководство поможет вам сделать мир чуточку лучше)))



tulskiy, 9 августа 2011 в 18:14 #

0

git merge master

Вот тут, если я не ошибаюсь создастся мерж-коммит, так? А вдруг мэйнтейнер не захочет чтобы у него в истории были невнятные merge from upstream/master?



Envek, 10 августа 2011 в 05:18 #

0

Тогда стоит посмотреть в сторону git rebase

git rebase master

Правда, стоит учесть, что git rebase меняет id коммитов, и, следовательно, его нужно делать ТОЛЬКО тогда, когда ваши изменения ещё лежат локально (т.е. ещё не за-push-ены на удалённый репозитарий)



FZambia, 26 сентября 2012 в 09:35 #

0

в 3-ем пункте, чтобы сделать отсылку на оригинальный репозиторий — нужно адрес репы и upstream поменять местами:

git remote add upstream https://github.com/user1/project

источник



<u>aknizubr</u>, 8 августа 2011 в 13:06 #

+6

Использую phpStorm, Pycharm, поддержка github в виде плагина установленного по умолчанию.

Оценка 5 из 5 как продуктов вообщем так и работы плагина.



AYShestakov, 13 июня 2012 в 14:10 #

0

почему то в phpStorm 4.0.2 push зависает.

Не сталкивались?

может где видели описание по настройке?



Небольшое дополнение для пользователей Windows.

+3

Консольный клиент (Git Bash) ищет приватный ключ ssh по пути ~/.ssh. Узнать, где находится этот каталог можно выполнив: pwd

Nikiit, 8 августа 2011 в 15:22 #

0

И еще помоему можно так: SET home=ВАШПУТЬ

<u>netslow</u>, 8 августа 2011 в 13:09 #

0

Я никак не могу нигде найти рецепт работы с Git+Eclipse. Как выкачать себе проект, чтобы он был нормальным eclipse проектом, и при этом можно было коммитить и делать push. С svn алгоритм такой: новый проект из svn -> создать проект используя визард, и у нас есть проект, привязанный к репозиторию. Как быть с git'oм?

netslow, 8 августа 2011 в 13:12 # 0

... при этом не очень бы хотелось хранить файлы .project и .classpath в репозитории

<u>VolCh</u>, 8 августа 2011 в 13:45 #

.gitignore, нет?

netslow, 8 августа 2011 в 14:37 # n

Я хотел сказать, что хранение этих файлов в qit решило бы проблему, можно выкачать код и импортировать проект. Но не хочется хранить IDE-specific файлы

FlashBurn, 8 августа 2011 в 13:51 # 0

нужно установить плагин EGit (Help -> Eclipse Marketplace...)

Затем правой кнопкой по проекту в Eclipse, выбрать Team -> Share Project. После установки egit там появится мастер для git

<u>netslow</u>, 8 августа 2011 в 14:39 # 0

Вы меня недооцениваете) Egit конечно стоит. Вопрос как в пустом воркспейсе получить привязанный проект

FlashBurn, 8 августа 2011 в 14:42 # 0

тогда File -> Import -> Git -> Projects from Git

netslow, 8 августа 2011 в 15:54 #

Ну почти то, что нужно. Только еще надо будет отдельно сказать share project, чтобы привязать проект к репозиторию. Но вообще у меня почему-то не работает Import(Eclipse 3.7, Egit 1.0)

НЛО прилетело и опубликовало эту надпись здесь

SergeyGrigorev, 8 августа 2011 в 13:45 # +6

В дополнение к этой статье, сделайте пожалуйста полноценное описание, как на github делать Fork, отправлять патчи, подтягивать изменения из основной ветки и т.д.

smind, 8 августа 2011 в 14:53 # +1

habrahabr.ru/blogs/Git/125799/#comment 4143817

LaggyLuke, 8 августа 2011 в 16:51 # 0

Для тех, кто знает английский, на гитхабе есть отличная документация, например: Fork A Repo

<u>AreD</u>, 8 августа 2011 в 14:23 # +1

Никто не подкажет, можно ли к NetBeans 7 прикрутить Git?

Nc Soft, 8 августа 2011 в 14:36 # +1

Почему-то в семерке новой его нет в доступных модулях, хотя раньше был.

Итак, качаем зип архив <u>plugins.netbeans.org/download/plugin/39785</u> и распаковываем.

habrahabr.ru/post/125799/ 13/15 Потом сервис->подключаемые модули->загружено там батон «добавить подключаемые подули», через обзор подключаем скачанные гитлиб и

гит. **AreD**, 9 августа 2011 в 00:01 # 0 Только что обновил каталог плагинов в NetBeans и Git плагин появился, датирован 2 августа сего года. Но все равно спасибо за ваш FlashBurn, 8 августа 2011 в 14:38 # +1 www.adam-bien.com/roller/abien/entry/netbeans 7 git a smoke **AreD**, 9 августа 2011 в 00:03 # 0 Спасибо за ваш оттвет, но вот незадача до сегодняшнего дня поиск по плагинам не находил этот плагин. Из этого и родился мой вопрос) <u>аааwk</u>, 8 августа 2011 в 15:54 # 0 Реально ли заставить Git работать с Gtihub из-под прокси? edelweard, 8 августа 2011 в 17:14 # 0 Можно использовать HTTPS вместо SSH для доступа к репозиторию на ГитХабе. **smind**, 8 августа 2011 в 17:58 # +1 однозначно можно, так и работаю с qithub с работы moadib, 8 августа 2011 в 20:27 # 0 Подтверждаю:) Mithgol, 8 августа 2011 в 16:31 # +3 Вам удалось сочинить ещё одну неплохую «домашнюю страницу» по Git, то есть отправную точку самостоятельных поисков информации. <u>Vitamon</u>, 8 августа 2011 в 23:45 # 0 Robert Penner +1 sphinks, 9 августа 2011 в 00:16 # Хорошая подборка ссылок. а вот сама статья не уверен, что для хабра. gothica, 12 сентября 2011 в 08:54 # n Никто не подскажет, как можно настроить Apatana Studio с GIT? **Finom**, 6 ноября 2011 в 23:02 # 0 Поправьте, пожалуйста, картинки в статье. Claud, 30 августа 2012 в 00:25 # Странно как-то: ключи сгенерированые через puttygen отказываются работать, а вот те что сделаны ssh-keygen заработали без проблем. <u>vladsharikov</u>, 26 октября 2013 в 15:08 # Гм. До переустановки винды все работало. Сейчас поставил Tortoise Git, пытаюсь скачать репозиторий с git в какую-нибудь папку, а компьютер Жму правой кнопкой в папке, выбираю «Git Clone», меню закрывается и ничего не происходит.

никак не реагирует на то. что я ему велю :)

От чего это может быть?

OKComputer, 27 октября 2013 в 20:17 # 0

Добавлю: если вы хотите сразу после прохождения обучающего курса залить результаты на гитхаб и продолжить работу в системе не устанавливая GUI, вводите команды в консоли

cd ~/.ssh ssh-keygen -t rsa -C «yourmail@yourmail.org»

habrahabr.ru/post/125799/ 14/15 Далее смотрите где создался файл id_rsa.pub и вводите его содержимое как SSH-RSA ключ в настройках вашего логина в github

Далее создаете нужное репо на гитхабе и заливаете в него результаты обучения. Команды на заливку гитхаб подскажет, но в общем должно быть что-то вроде

git remote add origin git@github.com:myusername/myreponame.git git push origin master

Удобный интерфейс Хабра? #2

Элон Маск скоро представит проект новой системы пассажирских перевозок Hyperloop

Открытое письмо для РОИ и ФИД по петиции об отмене 187-Ф3 (#ЗаконПротивИнтернета)

Все мозги в одном месте

Заказы для фрилансеров

Вакансии для айтишников

Уютная и дружелюбная