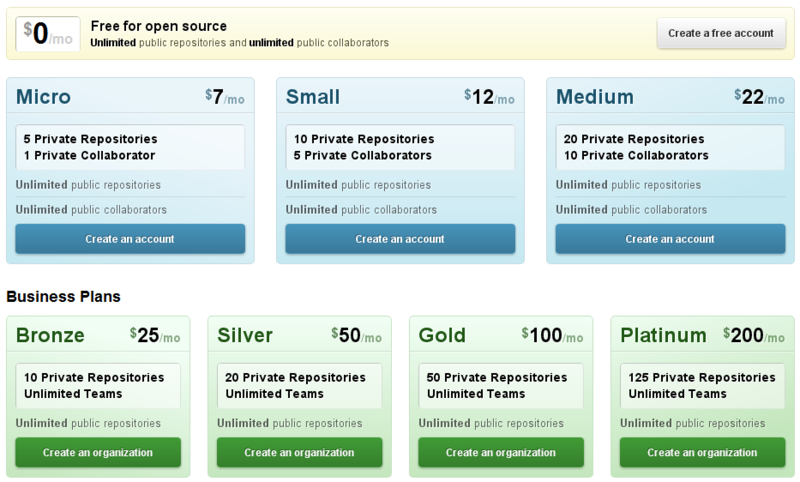
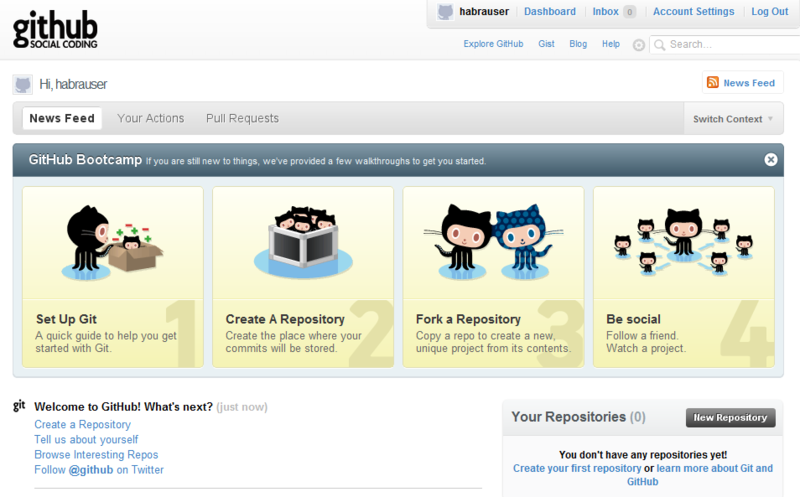
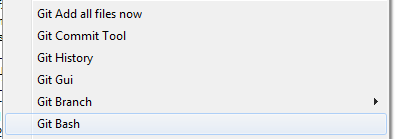
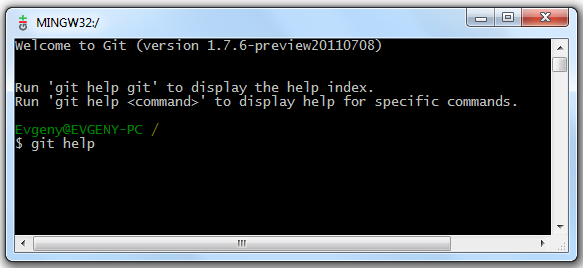
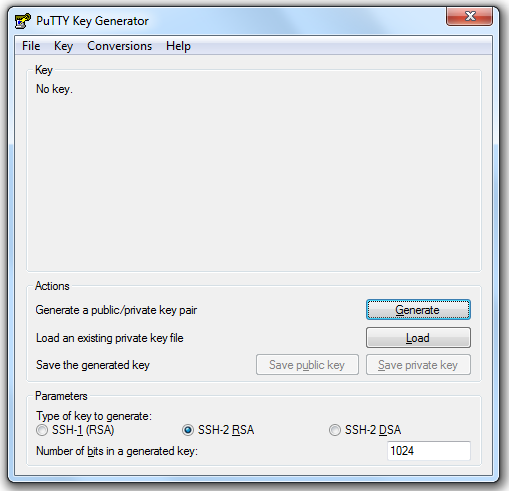
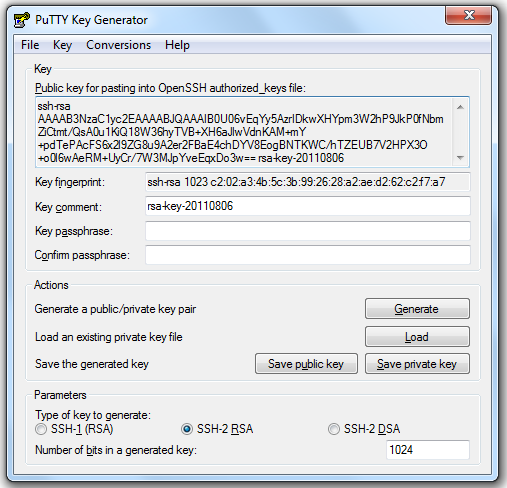
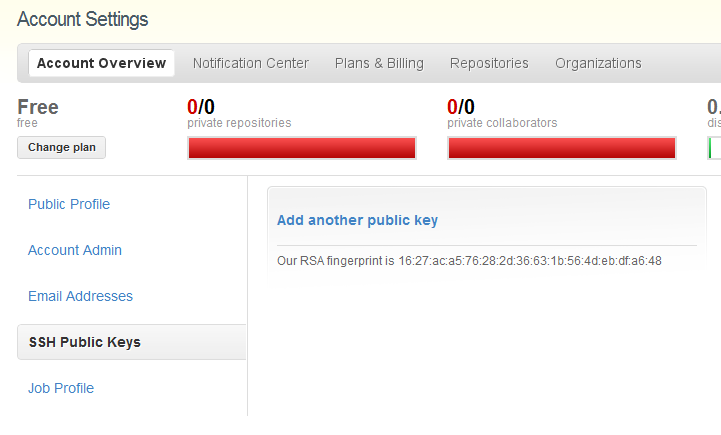
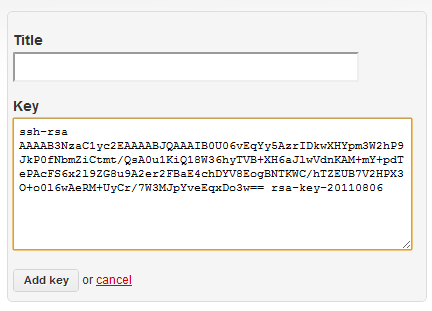
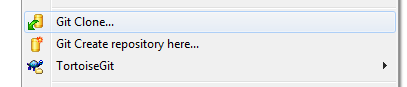
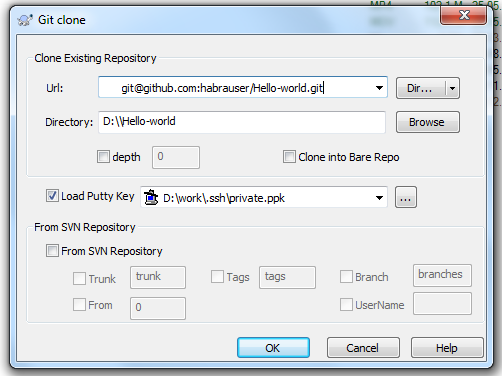
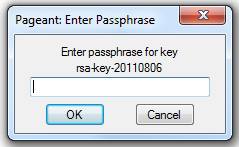
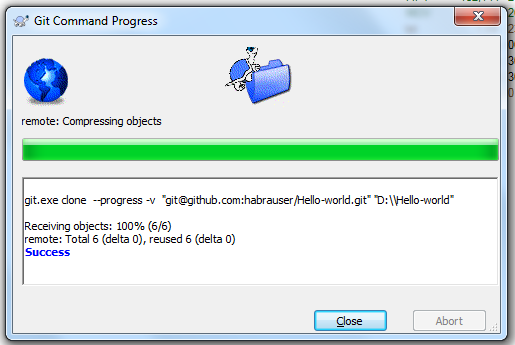
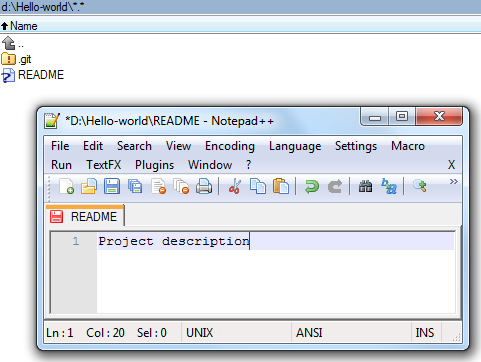
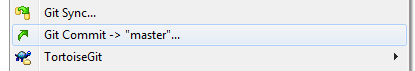
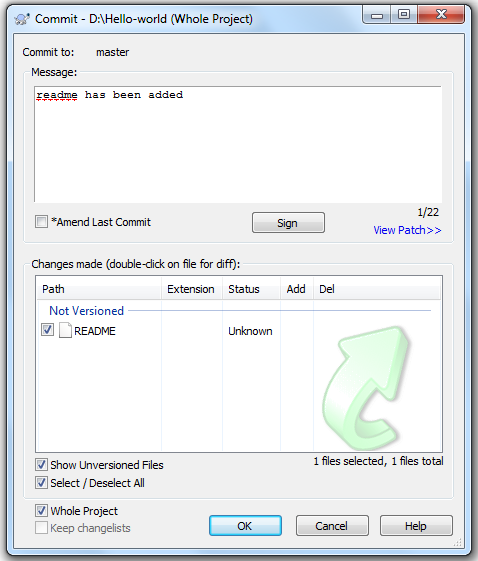
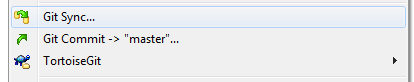
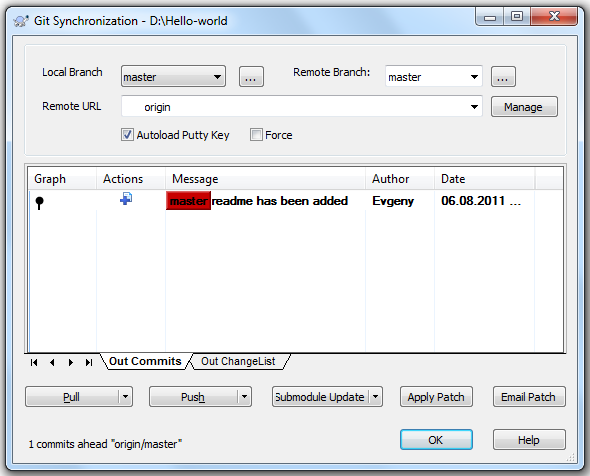
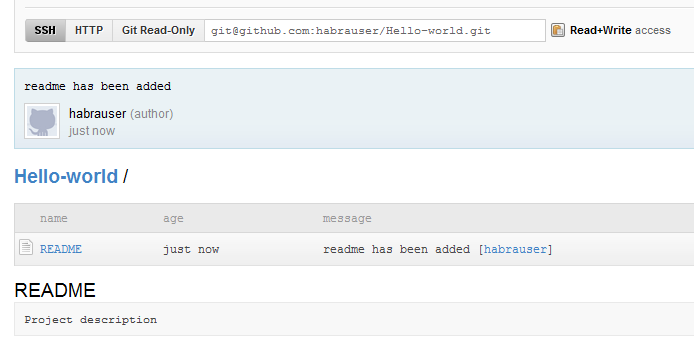
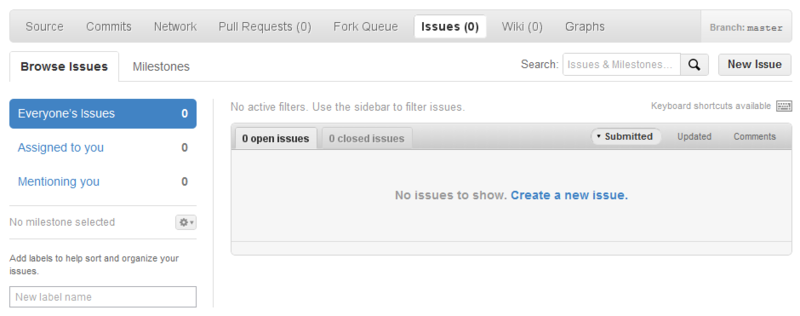
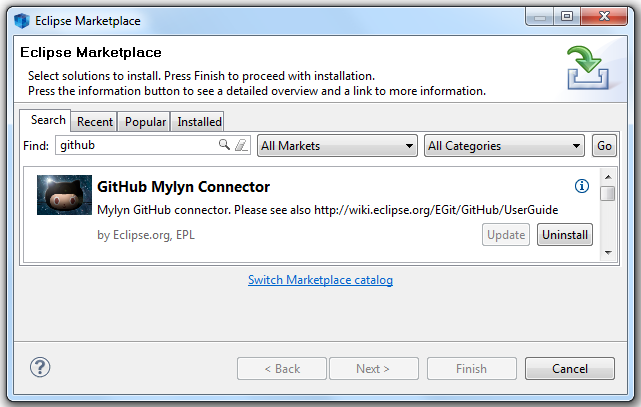
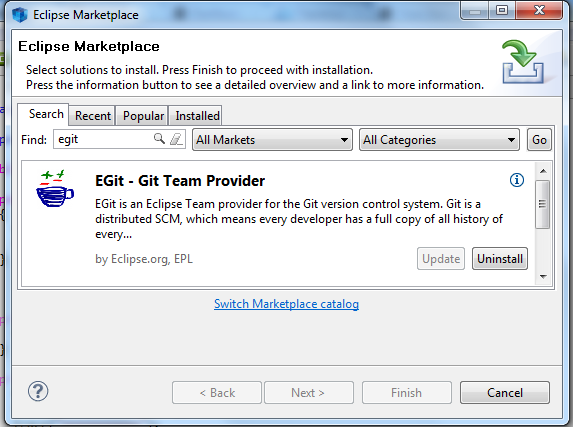
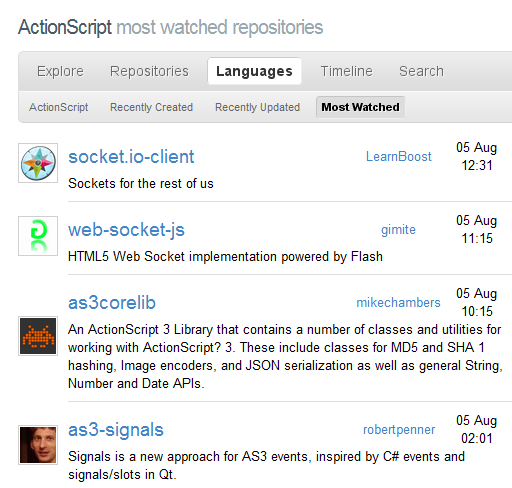
Как начать работать с GitHub: быстрый старт

* [Git](https://habrahabr.ru/hub/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-а, есть возможность проводить code review и многое другое. GitHub на данный момент [является](http://habrahabr.ru/blogs/Git/120535/) самым популярным сервисом такого рода, обогнав Sourceforge и Google Code.  
  
Для open-souce проектов использование сайта бесплатно. При необходимости иметь приватные репозитории, есть возможность перейти на платный тарифный план:  
  
  
Начнем с регистрации. Идем по ссылке [github.com/signup/free](https://github.com/signup/free) и вводим свои данные.  
После регистрации мы попадаем на Dashboard нашего аккаунта:  
  
  
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 [msysgit](http://code.google.com/p/msysgit/downloads/list). Это консольная версия git для Windows (далее расказ будет вестись на примере этой ОС).  
[Инструкция для MacOS X (eng)](http://help.github.com/mac-set-up-git/)  
[Инструкция для Linux (eng)](http://help.github.com/linux-set-up-git/)  
Проблем возникнуть не должно, просто везде жмем Next. После установки выбираем в контекстном меню Проводника Git Bash:  
  
  
или через Git Bash.lnk 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 [интерактивный курс](http://githowto.com/) по использованию git из консоли. Курс проходится за несколько часов и дает необходимые базовые навыки.  
  
Для тех, кто предпочитает gui — для Windows существует несколько таких инструментов для работы с git. Два основных — это [SmartGit](http://www.syntevo.com/smartgit/index.html) (кроссплатформенный) и [TortoiseGit](http://code.google.com/p/tortoisegit/). Оба неплохие, и какой использовать — дело вкуса. Я опишу работу с TortoiseGit.   
Для маков выбор giu тоже имеется. 

* [официальный клиент от GitHub](http://mac.github.com/) — на мой взгляд пока достаточно сыроват.
* [GitX](http://gitx.frim.nl/) — лично мне не приглянулся
* [GitBox](http://gitboxapp.com/) — наиболее следует mac-way, очень рекомендую попробовать именно его

Качаем по ссылке [code.google.com/p/tortoisegit/downloads/list](http://code.google.com/p/tortoisegit/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  
https://habrastorage.org/storage2/277/c64/88a/277c6488aee581bfb4daf75fdb7a4d59.png  
  
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 [тут](http://help.github.com/win-set-up-git/)). 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  
  
  
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 github (Account Settings) в раздел SSH Public Keys:  
  
  
жмем Add another public Key, вставляем наш публичный ключ:  
  
  
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 OK.  
  
  
Pageant запросит у наc 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:  
https://habrastorage.org/storage2/ca4/f97/adb/ca4f97adb3589cedd1edd0bf539e57ca.png  
  
Репозиторий успешно склонирован на локальную машину  
  
  
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  
  
  
  
  
и синхронизируем его с репозиторием на github:  
  
  
нажимаем Push  
  
  
Теперь зайдя на страницу нашего репозитория мы увидим следующее:  
  
  
Для каждого репозитория сайт предлагает wiki:  
  
  
а также простую систему issue tracking-a:  
  
  
кстати, для тех, кто использует в работе Eclipсe — есть соответствующий mylyn-коннектор для github:  
  
  
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.  
  
  
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.