Публикация на тему

Система контроля версий GIT для WEB

Использование системы контроля версий GIT в web-разработке

Автор
Михалькевич Александр Викторович

Публикация

Наименование Система контроля версий GIT для WEB
Автор А.В.Михалькевич
Специальность Использование системы контроля версий GIT в web-разработке,
Анотация
Anotation in English
Ключевые слова
Количество символов 196071

Содержание

Введение
1 Аннотация
2 Ключевые слова
3 Система контроля версий GIT для WEB
Заключение
Список использованных источников
Приложения

Введение

1 Аннотация

Система контроля версий — это система, сохраняющая изменения в одном или нескольких файлах так, чтобы потом можно было восстановить определённые старые версии.

2 Ключевые слова

Система контроля версий, GIT, github.com, web-приложение, репозиторий

3 Система контроля версий GIT для WEB

Среди прочих систем контроля версий (таких как SVN, Subversion, Mercurial и других) GIT выделяется наличием большого количества online-репозиториев для хранения проектов как с открытым исходным кодом (open source), так и с закрытым (private).

Системы контроля версий работают не просто с файлами системы, а с состояниями файлов.

В Git файлы могут находиться в одном из трёх состояний:

зафиксированном,
изменённом
подготовленном.

Зафиксированный значит, что файл уже сохранён в вашей локальной базе. К *изменённым* относятся файлы, которые поменялись, но ещё не были зафиксированы. *Подготовленные* файлы — это изменённые файлы, отмеченные для включения в следующий коммит.

Таким образом, в проекте с использованием Git есть три части: каталог Git (Git directory), рабочий каталог (working directory) и область подготовленных файлов (staging area).

Каталог Git — это место, где Git хранит метаданные и базу данных объектов вашего проекта. Это наиболее важная часть Git, и именно она копируется, когда вы клонируете репозиторий с другого компьютера.

Стандартный рабочий процесс с использованием Git выглядит примерно так:

1. Мы изменяем файлы в вашем рабочем каталоге.
2. Мы подготавливаем файлы, добавляя их слепки в область подготовленных файлов.
3. Мы делаем коммит. При этом слепки из области подготовленных файлов сохраняются в скрытый каталог Git.
4. Мы заливаем изменения на удаленный репозиторий с помощью команды push.

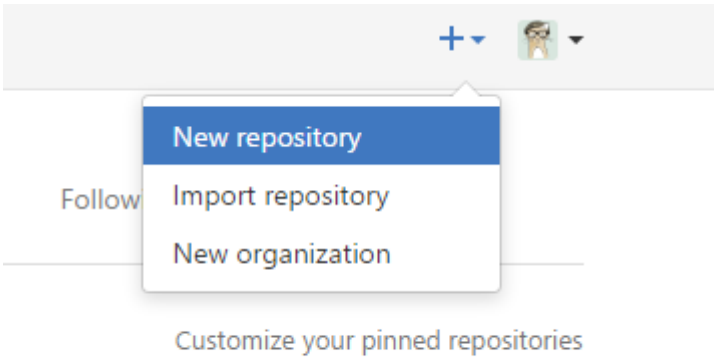
Если рабочая версия файла совпадает с версией в каталоге Git, файл считается зафиксированным.

Удаленный репозиторий

Для создания удаленного репозитория, нужна регистрация на на github.com (для openSource проектов). В учебных целях используется github.com, т.к. он бесплатен и работает с открытым исходным кодом.

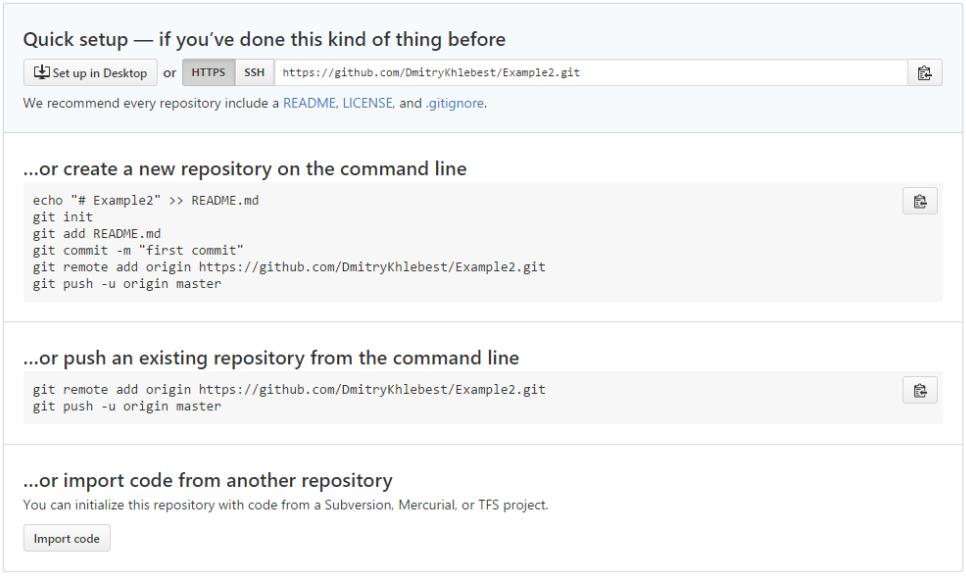
Зарегистрировавшись на github.com, заходим на свою страницу, нажимаем «+» (Create new…) и выбираем пункт «New repository».

Рисунок 1. Создание репозитория, github.com



После заполнения формы с единственным обязательным полем – названием репозитория, попадаем на такую страницу:

Рисунок 2. Пустой репозиторий github.com



Далее, используя командную строку, переносим файлы с локального репозитория на удаленный.

cd project – переход в папку с проектом

git init – инициализация пустого репозитория

git add * - добавление в новых файлов в репозиторий

git commit –m "text commit" – фиксация изменений

git status – проверка статуса

git remote add site http://github.com/user/project.git – создание переменной site хранящей путь к удаленному репозиторию.

git push site master – заливка файлов текущего репозитория на удаленный.

git pull site master – загрузка файлов удаленного репозитория на текущий.

Использование графической оболочки TortoiseGit для GIT

Тем, кто не привык пользоваться консолью, git предлагает множество, как бесплатных, так и платных графических оболочек. Среди множества оболочек остановим свой выбор на одной. TortoiseGit, или черепашка для git. Качаем по ссылке

code.google.com/p/tortoisegit/downloads/list. При установке везде жмем Next.

Клонирование репозитория

Клонирование репозитория можно осуществить как с помощью консольных команд, так и с помощью графической оболочки TortoiseGit

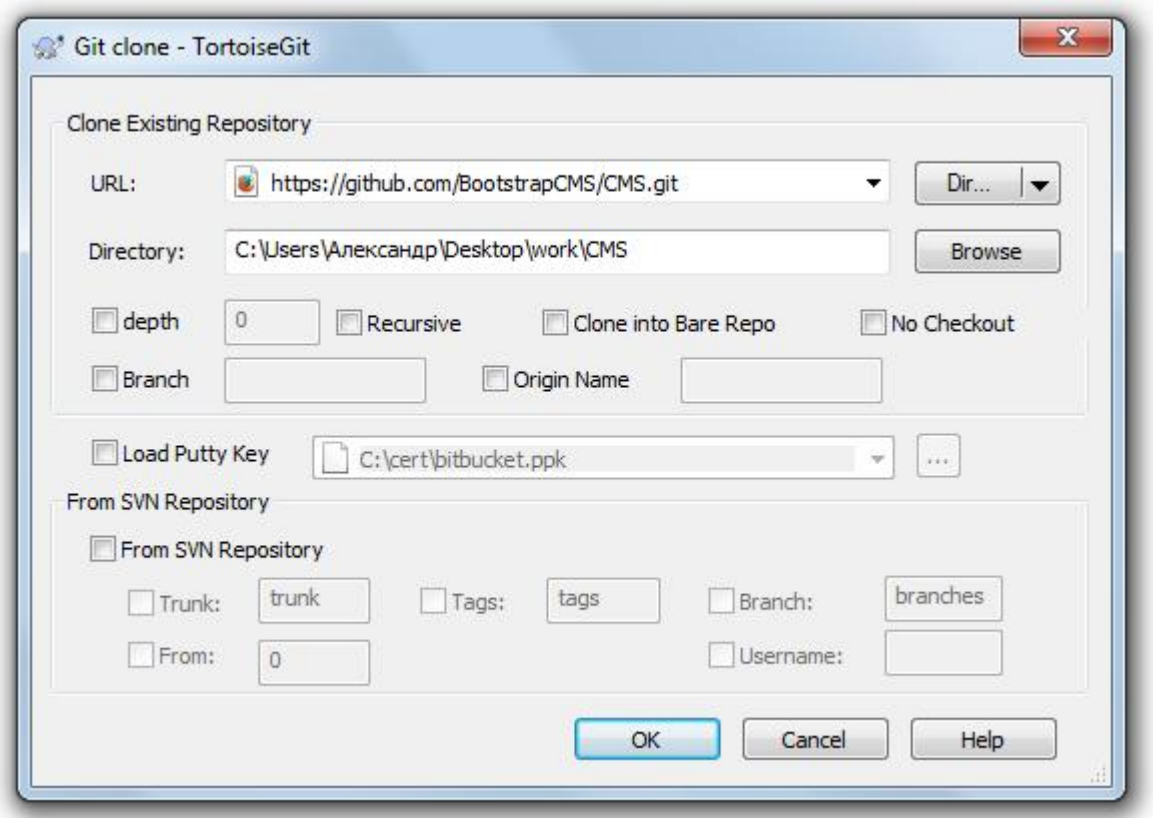
git clone http://github.com/user/project.git – клонирование удаленного репозитория с помощью консоли.

Для клонирования проекта с помощью черепашки, нажимаем правой кнопкой по пустой папке с именем проекта.

Рисунок 3. Клонирование проекта с помощью черепашки.

И выбраем команду Git Clone. В поле URL, в появившемся окне, вставляем url проекта.

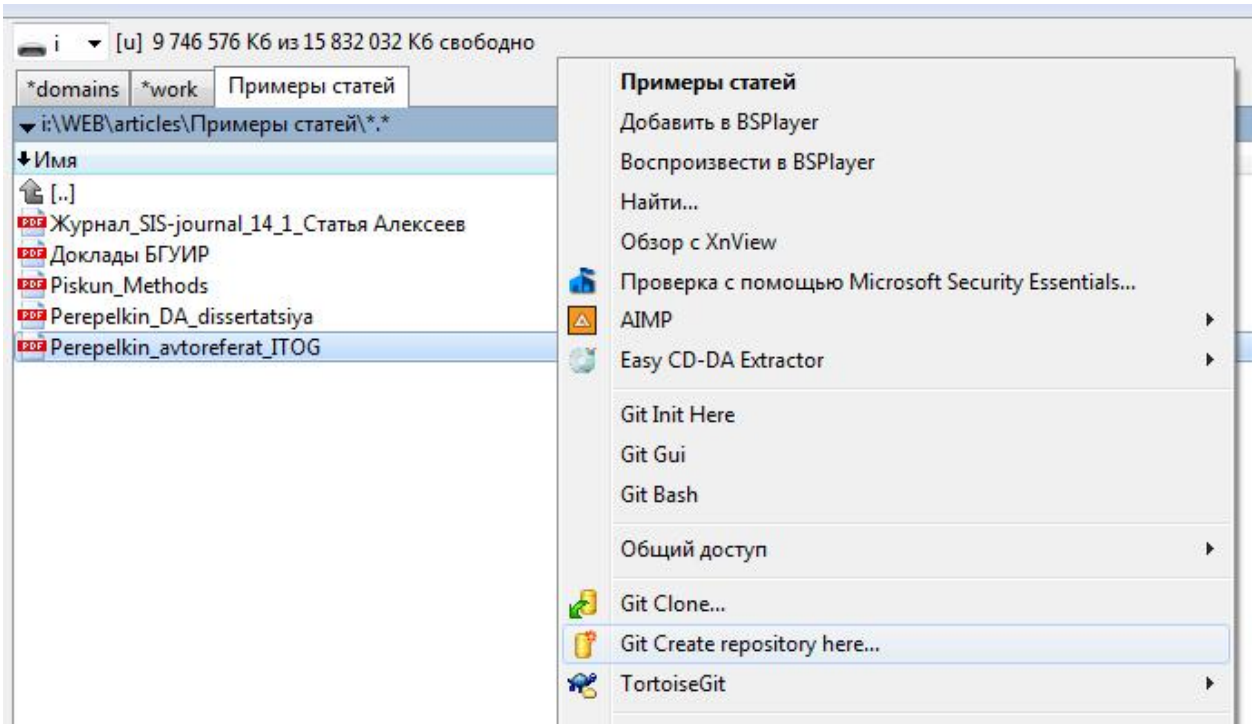
Рисунок 4. Поле для ввода url при клонировании проекта



Создание репозитория

Для создания репозитория в выбираем Git create repository here…

Рисунок 5. Создание репозитория в TortoiseGit



Пустой репозиторий создан. Его осталось наполнить файлами проекта.

Заключение

Список использованных источников

Приложения