

Лабораторная работа 3

Работа с GitHub

Основы GitHub	3
Регистрация в GitHub.....	4
Доступ по SSH	4
Внесение собственного вклада в публичные проекты.....	5
Создание ответвлений (fork)	5
Рабочий процесс с использованием GitHub	5
Задание	7
Задание 1. Отправка запроса на слияние	7

Основы GitHub

GitHub — это крупнейшее хранилище Git репозиториях, а так же центр сотрудничества для миллионов разработчиков и проектов. Огромный процент всех репозиториях хранится на GitHub, а многие проекты с открытым исходным кодом используют его ради Git хостинга, баг-трекера, рецензирования кода и других вещей. Так что, пока всё это не часть открытого Git проекта, наверняка вы захотите, или вам придётся взаимодействовать с GitHub при профессиональном использовании Git.

Регистрация в GitHub

Первым делом нужно создать бесплатную учётную запись. Просто зайдите на <https://github.com>, выберите имя которое ещё не занято, укажите адрес электронной почты и пароль, а затем нажмите большую зелёную кнопку “Sign up for GitHub”.

GitHub вышлет письмо для проверки вашего электронного адреса, необходимо выполнить подтверждение. GitHub использует ваш почтовый адрес для привязки ваших Git коммитов к вашей учётной записи. Если вы используете несколько почтовых адресов в своих коммитах и хотите, чтобы GitHub работал с ними корректно, то вам нужно будет добавить все используемые почтовые адреса в секцию под названием "Почтовые адреса" («Emails») в настройках профиля.

Доступ по SSH

На данный момент вы можете подключаться к репозиториям Git используя протокол авторизуясь при помощи только что созданного логина и Personal access token вместо пароля, токен можно создать в секции Developer Settings. Однако для того чтобы просто клонировать публично доступный проект, вам необязательно авторизоваться на сайте, но тем не менее, только что созданный аккаунт понадобится в то время, когда вы захотите загрузить (**push**) сделанные вами изменения.

Если же вы хотите использовать SSH доступ, в таком случае вам понадобится добавить публичный SSH ключ. Для генерации ключевой пары вы можете воспользоваться утилитой **ssh-keygen**, которая входит в большинство дистрибутивов linux.

Когда ключевая пара будет готова, откройте настройки вашей учётной записи нажав на иконку вашего профиля, расположенной в верхнем правом углу окна: и выберите Settings.

Выберите секцию слева под названием «SSH and GPG keys».

Затем нажмите «New SSH key» и добавьте публичную часть вашего ключа (файл ***.pub**, например **~/.ssh/id_rsa.pub** или другой в зависимости от типа шифрования) или вставьте содержимое файла в поле.

Внесение собственного вклада в публичные проекты

Создание ответвлений (fork)

Если вы хотите вносить свой вклад в уже существующие проекты, в которых у нас нет прав на внесения изменений путём отправки **push** изменений, вы можете создать своё собственное ответвление **fork** проекта. Это означает, что GitHub создаст вашу собственную копию проекта, данная копия будет находиться в вашем пространстве имён и вы сможете легко делать изменения путём отправки **push** изменений.

Исторически так сложилось, что англоязычный термин “fork” (создание ветвления проекта) имел негативный контекстный смысл, данный термин означал, что кто-то повёл или ведёт проект с открытым исходным кодом в другом, отличном от оригинала, направлении, иногда данный термин так же означал создание конкурирующего проекта с отдельными авторами. В контексте GitHub, “fork” (создание ветвления проекта) просто означает создание ветвления проекта в собственном пространстве имён, что позволяет вносить публичные изменения и делать свой собственный вклад в более открытом виде.

Таким образом, проекты не обеспокоены тем, чтобы пользователи, которые хотели бы выступать в роли соавторов, имели право на внесение изменений путём их отправки (push). Люди просто могут создавать свои собственные ветвления (fork), вносить туда изменения, а затем отправлять свои внесённые изменения в оригинальный репозиторий проекта путём создания запроса на принятие изменений (Pull Request). Запрос на принятие изменений (Pull Request) откроет новую ветвь с обсуждением отправляемого кода, и автор оригинального проекта, а так же другие его участники, могут принимать участие в обсуждении предлагаемых изменений до тех пор, пока автор проекта не будет ими доволен, после чего автор проекта может добавить предлагаемые изменения в проект.

Для того, чтобы создать ответвление проекта (fork), зайдите на страницу проекта и нажмите кнопку “Создать ответвление” (“Fork”), которая расположена в правом верхнем углу.

Через несколько секунд вы будете перенаправлены на собственную новую проектную страницу, содержащую вашу копию, в которой у вас есть права на запись.

Рабочий процесс с использованием GitHub

GitHub разработан с прицелом на определённый рабочий процесс с использованием запросов на слияния (Pull Request). Этот рабочий процесс хорошо подходит всем: и маленьким, сплочённым вокруг одного репозитория, командам; и крупным распределённым компаниям, и группам незнакомцев, сотрудничающих над проектом с сотней копий. Рабочий процесс GitHub основан на Тематических ветках.

Вот как это обычно работает:

1. Вы создаёте форк проекта:

```
$ git clone <repo_link>  
$ cd <repo_folder>
```

2. Создаёте тематическую ветку (например **hotfix**) на основании ветки **master**:

```
$ git checkout -b hotfix
```

3. Производите один или несколько коммитов с изменениями, улучшающих проект.

```
$ git commit -a -m «Fixed and improved»
```

4. Отправляете эту ветку в ваш проект на GitHub.

```
$ git push origin hotfix
```

5. На портале GitHub открываете запрос на слияние.

6. Обсуждаете его, вносите изменения, если нужно.

7. Владелец проекта принимает решение о принятии изменений, либо об их отклонении.

8. Получаете обновлённую ветку **master** и можете отправить её в свой форк.

Задание

Задание 1. Отправка запроса на слияние

Откройте страницу GitHub вашего преподавателя <https://github.com/gulyaeev>

Выберите любой публичный проект, сделайте его форк, внесите изменения в тематическую ветку и направьте запрос на слияние (Pull Request).

| *Если ваш запрос был принят или отклонён - значит я его посмотрел.*