











Неделя #1: Техническое задание

Работа с gulp. Автоматическая сборка проекта

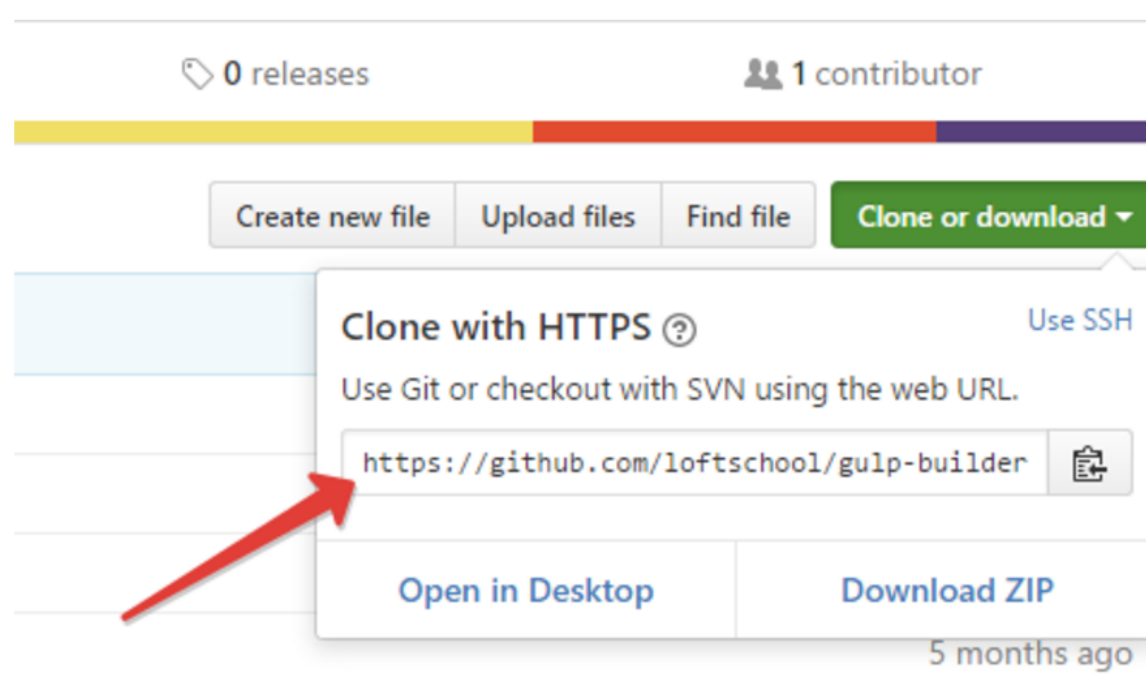
Подготовка к выполнению задания

-  Внимательно изучить все материалы первой недели обучения (первые 3 вебинара):
 -  Workflow #1 - terminal, git, npm
 -  Workflow #2 - gulp
 -  Workflow #3 - webpack, yarn
-  Знать основы git, github, javascript, terminal, npm, gulp
-  Быть готовым общаться с наставниками и коллегами. Не бояться “гуглить”.
-  Для работы над выпускными проектами вы будете использовать gulp сборку, доступную на github [по ссылке](#).
-  Работа на курсе с webpack согласуется отдельно с Ковальчуком Дмитрием и вашим наставником.

Пошаговая инструкция

-  1. Зарегистрируйтесь в github.
-  2. Перейдите в [репозиторий сборки](#)

Скопируйте ссылку



-  3. В терминале перейдите в папку с вашими сайтами и наберите команду

```
git clone https://github.com/loftschool/gulp-builder-dz1.git my-builder
cd my-builder
```

-  4. Установим зависимости проекта

```
npm i
```



5. Убедитесь, что сборка работает (у вас должны быть установлены: gulp 4 и gulp-cli)

```
gulp
```



6. Создайте две отдельных дополнительные Задачи (далее “таски”, от англ. “task”), которые будут выполнять следующие функции:



6.1 Первый таск - генерация спрайтов из графических файлов (png, gif).



6.2 Второй таск - копирование файлов из папки с исходниками в папку для продакшена (картинки, шрифты).



7. Подключите созданные таски к общей сборке проекта.



8. Отредактируйте README.md - добавьте в него свои ФИО и краткое описание того, что вы сделали.



9. Когда всё будет готово сделайте коммит.

```
git add .  
git commit
```



10. Создайте на github новый репозиторий без README файла (т.е. не нужно ставить галочку напротив “initialize this repository with a README”). Назовите его “my-builder”.



11. Скопируйте адрес нового репозитория.



12. Наберите в терминале

```
git remote set-url origin git@github.com:username/my-builder.git
```



13. Мы только что поменяли ссылку на репозиторий. Раньше это была ссылка на лофтскул, теперь, на наш личный проект. Значит теперь мы будем пушить файлы на гитхаб в свой новый личный репозиторий. Давайте убедимся, что мы сделали правильно.

```
git remote -v  
// origin git@github.com:username/my-builder.git (fetch)  
// origin git@github.com:username/my-builder.git (push)
```



14. Отправим файлы на гитхаб, в наш репозиторий.

```
git push -u origin master
```

ВАЖНО! У вас должен быть настроен доступ по SSH (см. вебинар по git)



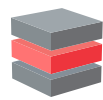
15. Проверяем результат

```
https://github.com/username/my-builder
```



Требования к вашим задачам

- 📌 Названия задач должны отражать их назначение. Например: scss-compile для компилирования scss файлов, ftp-upload - для загрузки файлов на сервер и.т.д.
- 📌 Возможность запуска задач по отдельности, путём вызова команды “gulp <task name>”



Требования к работе сборки

- 📌 Для запуска работы должна использоваться одна команда gulp - задача(task) default
- 📌 Разделение структуры проекта. В отдельных директориях: исходные файлы и сборка проекта для продакшена. Другими словами одни задачи выполняют компиляцию файлов из исходников, например для файлов стилей из scss в css, для html - конвертация из pug файлов. Другие задачи занимаются копированием и группированием файлов в отдельную папку продакшена. Не забываем, что разработка ведется в одной папке, а файл сборки собирает в другую, которая будет выложена на сервер.

Итогом работы будет считаться полностью функционирующая сборка, с двумя добавленными вами задачами, полноценно обеспечивающая автоматизацию при разработке проекта.