**Выкладываем сайт на сервер**

Фронтенд проекта Mesto готов. Но какой в этом толк, если ваш сайт никто не увидит?

Пора его «задеплоить». Deploy — это размещение проекта на сервере. После того, как вы задеплоите проект, им смогут пользоваться все — то есть сайт появится в интернете.

Процесс деплоя — это не всегда простое выкладывание файлов на сервер. Кроме этого деплой может включать: транспиляцию кода, подтягивание зависимостей и настройку серверных сценариев. Поэтому и придуман термин «деплой», а то говорили бы просто «залить на сервер».

Впрочем, в нашем проекте есть только несколько статичных файлов. Потому сегодня вся эта магия деплоя нам не понадобится. Но в курсе по бэкенду деплой будет устроен сложнее.

**Хостинг и Github Pages**

Чтобы сделать сайт доступным в интернете, нужен хостинг. Его предоставляет провайдер, на чей сервер вы выложите сайт. В реальных проектах нужно подбирать правильный хостинг для сайта: они отличаются по стоимости и возможностям.

Но для учебных проектов подойдёт Github Pages. Это сервис «Гитхаба», который позволяет выложить сайт в интернет прямо из Гитхаб-репозитория.

**Наcтройка Github Pages**

Чтобы добавить проект в сервис Github Pages нужно создать в репозитории ветку gh-pages и загрузить туда собранный проект. Но проще сделать это [специальным пакетом](https://www.npmjs.com/package/gh-pages) — gh-pages.

1. Установите его:

Скопировать кодBASH

npm install gh-pages --save-dev

В файле package.json в разделе devDependencies появится новая строка:

Скопировать кодJSON

devDependencies: {

...

"gh-pages": "^3.1.0"

...

}

1. Добавьте в package.json новый скрипт — deploy. Этот скрипт должен вызывать пакет gh-pages и передавать ему папку, в которой находится собранный проект. В нашем случае это папка dist:

Скопировать кодJSON

scripts: {

...

"deploy": "gh-pages -d dist"

}

1. Обычно перед деплоем нужно собрать проект командой npm run build. Чтобы не делать это каждый раз вручную, можно сказать npm, чтобы перед каждым вызовом скрипта deploy вызывался build. Для этого в раздел scripts добавьте скрипт predeploy:

Скопировать кодJSON

scripts: {

...

"predeploy": "npm run build",

"deploy": "gh-pages -d dist"

}

Такой скрипт будет вызываться при каждом вызове скрипта deploy.

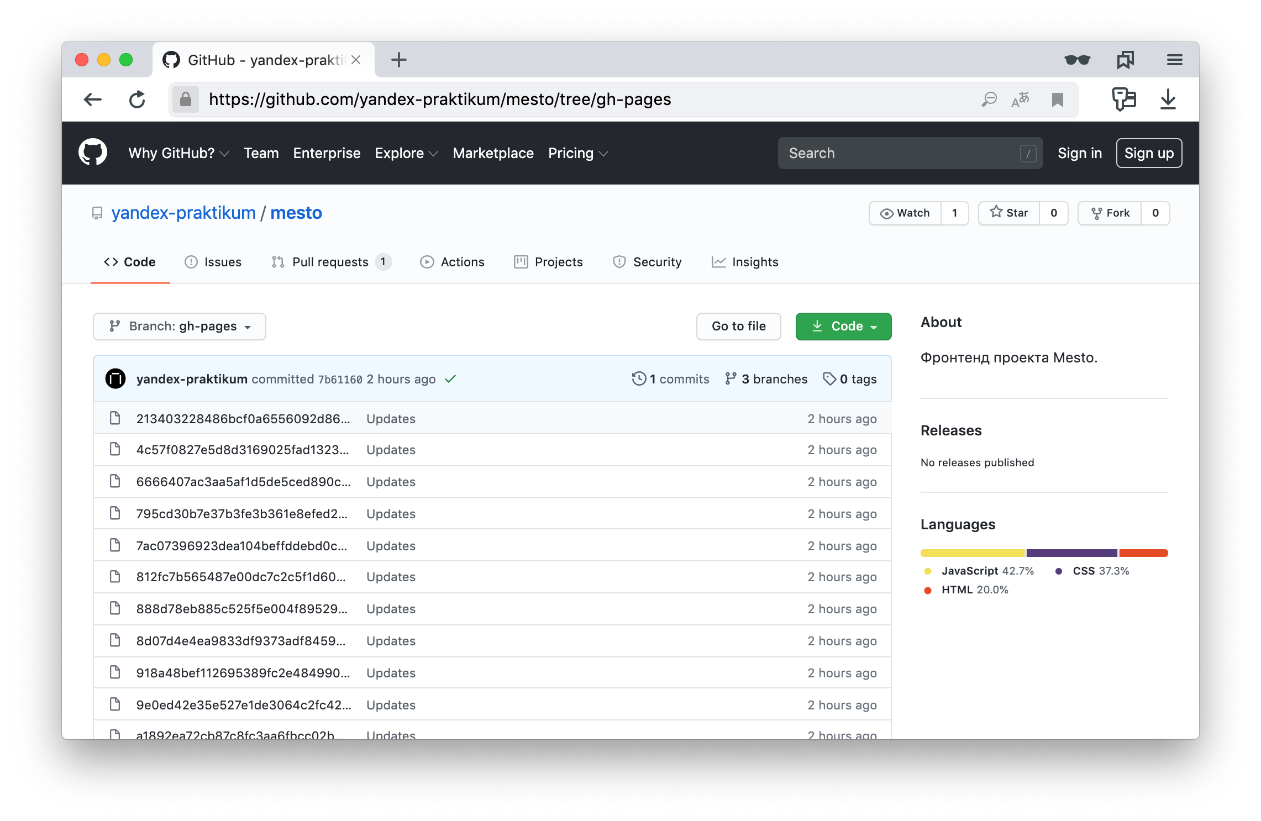
1. Все настройки готовы. Запустите команду npm run deploy. Проект соберётся и содержимое папки dist попадёт в удалённую ветку gh-pages. Если при этом возникла ошибка:

Скопировать код

Couldn't find remote ref refs/heads/gh-pages

Введите команду npx gh-pages-clean и начните пункт 4 заново.

1. Зайдите на «Гитхаб» и убедитесь что ветка gh-pages со сборкой появилась там:



1. Через несколько минут сайт будет доступен по адресу:

Скопировать код

https://stasbasov.github.io/mesto

Только замените stasbasov на своё имя пользователя, а mesto –– на название репозитория.

Теперь, после внесения изменений в проект, достаточно вызвать команду npm run deploy и изменения отобразятся на сайте.

Добавьте ссылку на сайт в описание репозитория на «Гитхабе». Пользователи, которые зайдут в него, смогут посмотреть не только код, но и сам сайт и быстро протестировать его. Это удобно!

https:// akatryukhin.github.io/mesto