



Размещение проектов на репозитории

Содержание

Мы рассмотрим такие темы про размещение проектов в резюме, как:

- Зачем размещать проекты?
- Какие правила размещения проектов?
- Создание и настройка проекта на Github
- Как выложить мой проект?

Зачем выкладывать проекты на репозитории?

Прежде чем предложить вам место в компании, работодатель проверит ваш уровень владения нужными навыками. Например, через тестовое задание. Как правило, оно содержит пример реальной ситуации, с которой сталкивались или могут встретиться аналитики в компании. Часто тестовое не покрывает все возможные рабочие ситуации, потому что работодатели изучают **проекты кандидата**.

Проекты, выполненные вами при прохождении курсов профессий Яндекс.Практикум, дают представление о ваших знаниях и умениях и их соответствии требованиям работодателя.

Прикладывать к резюме выполненные проекты необязательно. Но их наличие — большой плюс.

Какие правила размещения проектов?

1. Храните ваши проекты в репозиториях.

Мы рекомендуем размещать свои решения в облачных репозиториях.

Репозиторий — место, где хранятся и поддерживаются данные; чаще всего в виде файлов, доступных для дальнейшего распространения по сети. Популярные репозитории: Github, Gitlab, Bitbucket. Преимущество облачного репозитория — контроль доступа к нему. Вы можете сделать репозиторий как открытым для всех, так и закрытым. В закрытом репозитории ваши проекты будут доступны не всему интернету, а только тем, кому вы выдаёте доступ.

Чтобы работать в этих сервисах, понадобятся знания основ **Git** — распределённой системы управления версиями. Её применяют для хранения нескольких версий одного и того же документа, с возможностью возвращения к ранним версиям и определения того, кто и когда вносил изменения.

2. Не размещайте материалы курсов (тексты теории, задачи, квизы, датасеты) в репозиториях.

Содержание курсов — плод трудов многочисленного авторского коллектива, редакции и консультантов из ведущих компаний России. Многие данные были эксклюзивно предоставлены команде «Яндекс.Практикум» для обучения будущих специалистов.

3. В своих проектах оставьте только то, что вы сделали сами.

Основная цель репозитория — показать, на что вы способны. **Не размещайте в репозиториях полное описание проектов и предоставляемые наборы данных.** Как работодатель поймёт, о чём идет речь в проекте? Сделайте краткое описание к каждому проекту. Укажите источник данных и основные цели исследования. **Не забывайте удалять комментарии от ревьюера.** Комментарии от ревьюера нужны для образовательных целей, работодателю они ни к чему. Оставьте результаты обработки данных в результате исполнения ячеек: вызовы методов *info*, *head* и других подобных, построение графиков и диаграмм — это всё показывает те навыки, которые вы приобрели в рамках обучения в Яндекс.Практикум

4. Размещайте проекты с описаниями в отдельных папках. Назовите папки так, чтобы можно было отличить один проект от другого. Не забывайте давать тетрадам Jupyter Notebook значащие имена.

Если вы назовёте свои проекты ``super-duper-cool-project.ipynb``, ``yet-anoter-super-duper-cool-project.ipynb``, ``my-awesome-super-duper-cool-`

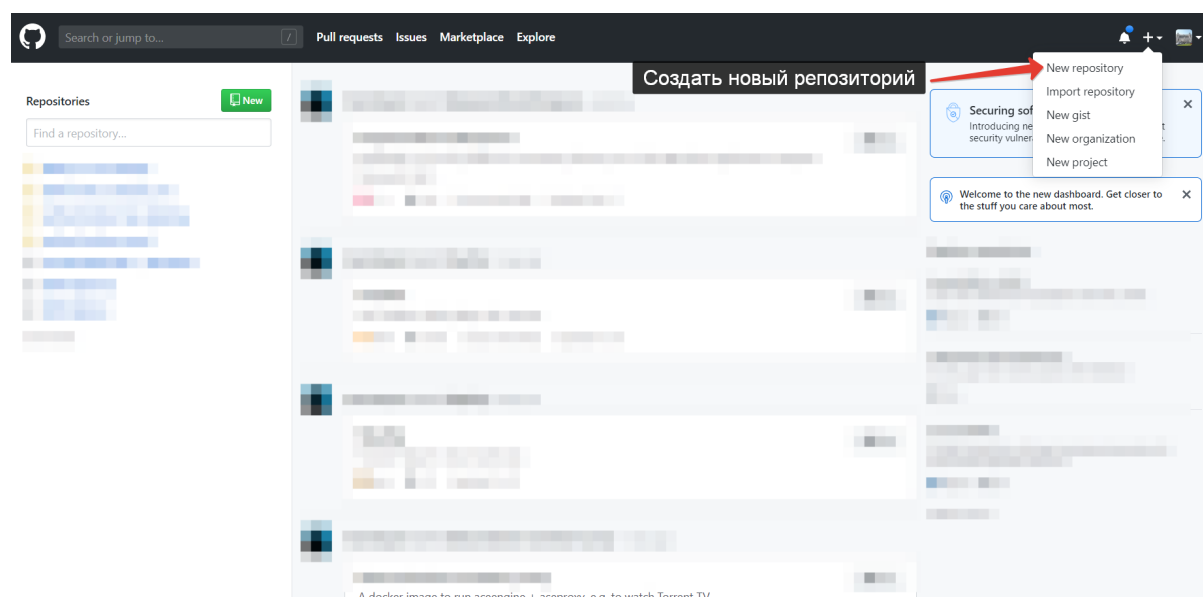
project.ipynb`, работодатель может запутаться в названиях и не понять, что вы делали в проекте.

Напомним, чтобы скачать тетради *Jupyter Notebook* из тренажёра, перейдите на страницу проекта в меню *File*, выберите пункт *Download as* и выберите формат *notebook*.

Как работать с проектом на Github с помощью Git

Создание и настройка проекта на Github

Зарегистрируйтесь на github.com. Чтобы создать свой репозиторий, войдите на github, в правом верхнем углу нажмите на знак **+** и выберите пункт меню **New repository**.



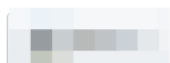
Выберите имя репозитория — оно должно содержать строчные латинские буквы, цифры и тире. В поле **Description** можете написать краткое описание репозитория. Вы можете выбрать два варианта типа доступа для репозитория: публичный (public) и приватный (private). Если вы выберете приватный доступ, то репозиторий смогут просматривать только те, кому вы выдадите доступ. Дополнительно вы можете поставить галочку в пункте **Добавить в репозиторий файл README при инициализации**. Так, при создании репозитория markdown-файл readme будет доступен.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner

Repository name *

 /

Great repository names are short and memorable. Need inspiration? How about **solid-barnacle**?

Description (optional)



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☒ Initialize this repository with a README

This will let you immediately clone the repository to your computer.

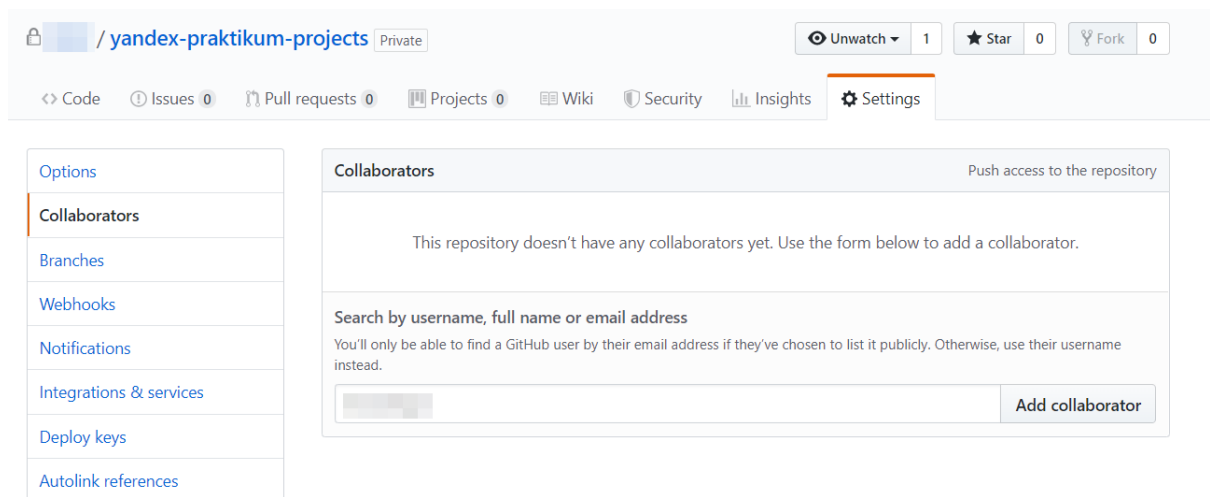
Add .gitignore: None ▾

Add a license: None ▾



Create repository

Чтобы посторонний человек — специалист по подбору персонала или ваш друг — мог посмотреть ваш репозиторий, нужно выдать доступ. Перейдите в меню **Settings → Collaborators** и в поле **Search by username, full name or email address** введите логин или *email* пользователя, которому хотите выдать доступ. Нажмите на **Add collaborator**:



После создания репозитория на github, нужно продолжить работу с ним локально. Например, через клиентов с графическим интерфейсом: GitHub Desktop, SourceTree, GitKraken. Однако настоящие аналитики и специалисты по data science работают только с консольным интерфейсом. В освоении консольного git вам поможет интерактивный тур по git <https://githowto.com/ru>. Если вы затрудняетесь работать с консольным интерфейсом, рекомендуем скачать [GitHub Desktop](#) и работать там. В интернете есть множество инструкций по работе с GitHub Desktop, например, вот это [видео](#), или эта [инструкция](#).

Как выложить мой проект?

Какая структура должна быть в репозитории?

- Для каждого проекта создайте отдельную папку;
- В папке должно содержаться два файла: `README.md` и файл с расширением `.ipynb` с вашим решением проекта;
- Файл `README.md` должен содержать краткое описание проекта. Включите в описание рассказ об исследуемом наборе данных, основных пунктах и целях исследования; укажите библиотеки, которые вы использовали для выполнения проекта;
- Файл с расширением `.ipynb` должен содержать тетрадь с вашим выполненным проектом. Все ячейки в тетради должны быть выполнены, выводы ячеек (при наличии) должны быть сохранены;
- Файл с проектом не должен содержать комментариев наставника;

- В корне репозитория обязательно должен быть файл `README.md`. В этом файле должна быть структура репозитория: проекты, их краткое описание, что в какой папке находится.

В итоге репозиторий должен выглядеть так:

Yandex-Practicum / `practicum_data_example_repo`

Unwatch 4 Star 1 Fork 1

<> Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security 0 Insights Settings

Example repository for students of Data Faculty [Edit](#)

[Manage topics](#)

1 commit 1 branch 0 packages 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

DrhF Initial commit	Latest commit 5bc5647 5 days ago
big_cities_music	Initial commit 5 days ago
.gitignore	Initial commit 5 days ago
README.md	Initial commit 5 days ago

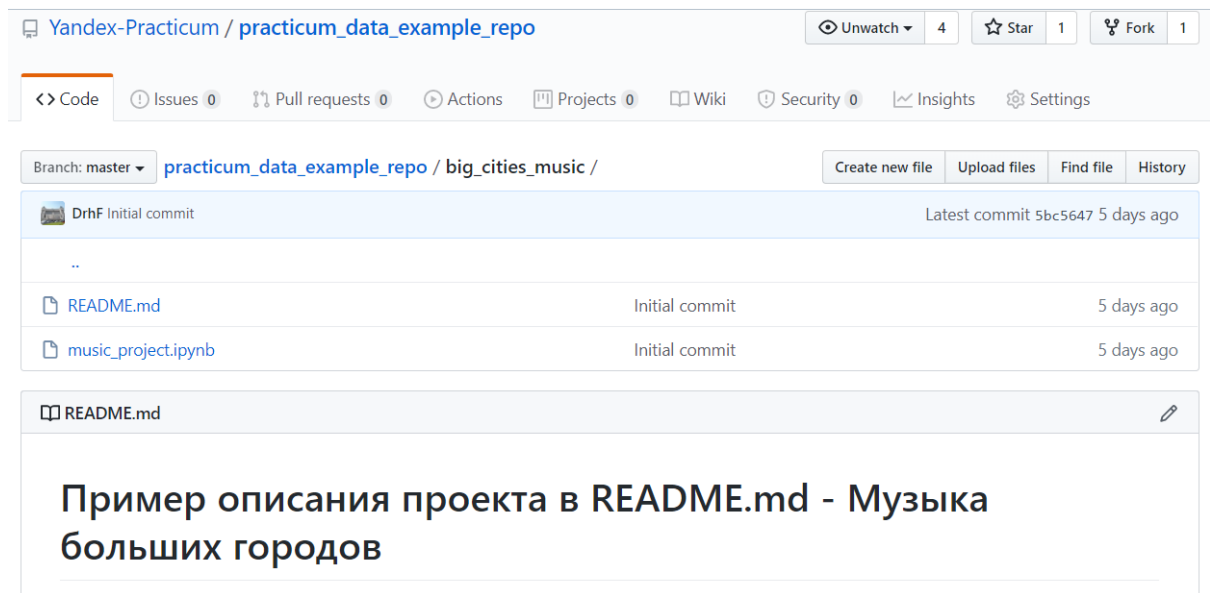
README.md

Пример репозитория для выпускников профессий "Аналитик данных" и "Специалист по Data Science"

Этот репозиторий может служить примером того, как нужно оформлять свои проекты в резюме.

Пример описания проектов в README.md

А папка с проектом — так:



Пример оформления репозитория: https://github.com/Yandex-Practicum/practicum_data_example_repo

Разметка Markdown

Как вы могли заметить, для описания создают файлы с расширением `.md`. Это текстовые файлы, записанные с помощью упрощённого языка разметки — **Markdown**. Этот язык разметки имеет особый синтаксис для оформления текста:

```
# Это заголовок первого уровня
## Это заголовок второго уровня
### Это заголовок третьего уровня
#### Это заголовок четвёртого уровня

**этот текст имеет полужирное начертание**
*А этот текст имеет курсивное начертание*
***И ничто нам не мешает сделать и курсивное, и полужирное начертание***

- Это первый элемент маркированного списка
- Это второй элемент маркированного списка
  - вложенный элемент маркированного списка

> Именно так оформляются цитаты

1. Нумерованный список начинается так
2. Можно продолжить так
  1. Вложенные элементы описываются так
  2. И так

| Markdown | Поддерживает | Таблицы |
| :----- | :-----: | :-----: |
| с выравниванием слева | с выравниванием справа | и с выравниванием по центру |
```

```
```python
print("Это фрагмент кода на python")
```
```

[Текст ссылки](адрес://ссылки.здесь "Заголовок ссылки")
[Подробнее о Markdown по ссылке](https://daringfireball.net/projects/markdown/)

Оформите файлы описания проектов на языке разметки Markdown.

Если у вас возникли проблемы с языком разметки Markdown, пройдите интерактивный тренажёр по Markdown:

<https://commonmark.org/help/tutorial/>

Чеклист готовности репозитория для резюме

- ☐ Все проекты, которые я делал в профессии, скачаны
- ☐ Из всех проектов удалены комментарии ревьюера и другая ненужная информация
- ☐ Создан репозиторий на github/gitlab/bitbucket
- ☐ Создана локальная копия репозитория
- ☐ В локальном репозитории сделаны папки под каждый проект
- ☐ В каждой папке проекта лежит файл проекта и описание README.md
- ☐ Для каждого проекта описание в файле README.md содержит основные пункты и цели исследования, используемые библиотеки
- ☐ В корне локального репозитория лежит описание всех проектов README.md
- ☐ В файле с описанием всех проектов README.md для каждого проекта указан путь и краткое описание
- ☐ Локальный репозиторий выгружен на github/gitlab/bitbucket

Часто задаваемые вопросы

▼ Стоит ли выкладывать переделанные проекты на гитхаб? Как это сделать?

Если вы понимаете, что сделали проект не самым лучшим образом и что он показывает ваши знания и навыки не в самом лучшем свете, то лучше заменить новым. Если же вы понимаете, что что-то сделали бы по-другому, то можете в той же папке разместить новый проект с указанием в *README.md*, что изменилось.

▼ Стоит ли выкладывать решённые тестовые задания? Какие правила?

Вы можете выкладывать решённые тестовые, но при этом работают те же правила: выкладывать нужно только то, что относится к вашему труду. Не стоит указывать, какая компания выдала вам тестовое задание, а также полностью копировать описание задания в *README.md*.

▼ Стоит ли выкладывать свои решения с соревнований по DS?

Вы можете выкладывать свои решения с соревнований, но перед этим прочитайте правила: возможно, они запрещают публикацию решений. Однозначно не рекомендуем выкладывать решения текущих соревнований. Не забывайте в описании оставлять ссылку на соревнование.