

# Тесты локально проходят, но на платформе падают: что делать



Ниже представлено два чек-листа — для всех приложений на Python и для Django-приложений. Выберите подходящий вашей ситуации чек-лист и пройдите по нему.

## Чек-лист для всех приложений

### Локально тесты проходят, а на платформе нет

- ☐ Убедитесь, что проект написан с использованием возможностей Python 3.9. Тесты также должны запускаться именно на этой версии Python.
- ☐ Иногда тесты на платформе обновляются, а у вас может остаться их старая версия. Синхронизируйте тесты, которые хранятся локально с тестами в исходном репозитории с проектом от Практикума (как говорят программисты — «спульте»):

```
git pull https://github.com/yandex-praktikum/НАЗВАНИЕ_РЕПОЗИТОРИЯ.git
```

- ☐ Убедитесь, что в файле `.gitignore` нет файлов, которые не относятся к проекту или являются производными проекта, например `.vscode`, `__pycache__`, `*.db`. Все файлы больших размеров, как например файл базы данных, — это производные от проекта, добавлять их не нужно.
- ☐ Убедитесь, что все необходимые файлы вашего проекта были включены в коммит и попали в репозиторий.
- ☐ Если в исходном репозитории проекта есть константы, убедитесь, что их названия не изменялись. Эти константы используются при тестировании на платформе. Например:

```
PRACTICUM_TOKEN = os.getenv('PRACTICUM_TOKEN')
```

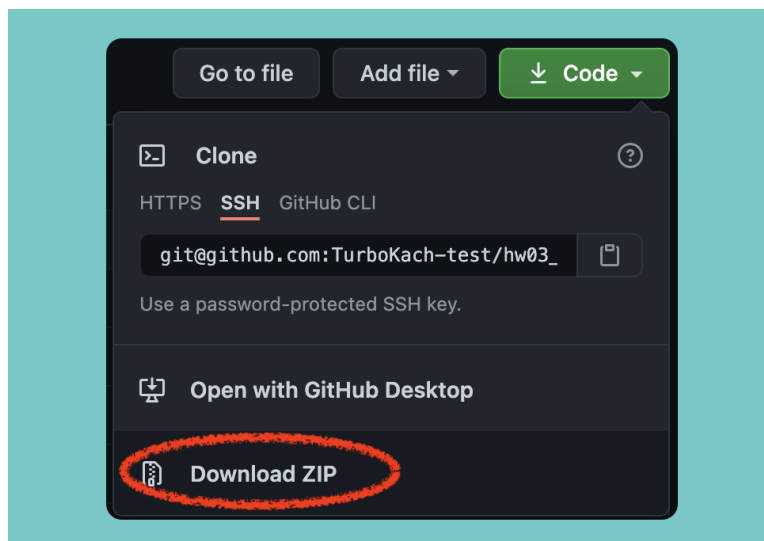
- ☐ Убедитесь, что для констант, которые берутся из переменных окружения, указаны значения по умолчанию. Они используются в тестах на платформе. Если нет .env-файла, значения по умолчанию должны быть валидными и проходить тесты. Например:

```
SECRET_KEY = os.getenv('SECRET_KEY', default="SUP3R-S3CR3T-K3Y-F0R-MY-PR0J3CT")
```

- ☐ Проверьте регистр названий папок и файлов. Git не увидит изменения регистра в названиях файлов и папок. Команда `git config core.ignorecase false` сделает Git чувствительным к регистру.

## Проект не загружается на платформу

- ☐ Убедитесь, что загружаемый вами проект находится именно в ветке master вашего репозитория
- ☐ Убедитесь, что репозиторий не был переименован.
- ☐ Если проект загружается архивом, убедитесь, что формат архива .zip. Другие форматы платформа не поддерживает. Лучше всего загружать архивы, созданные из репозитория github:



# Чек-лист для Django-приложений

## Локально тесты проходят, а на платформе нет

- ☐ Убедитесь, что все модули, которые используются в проекте, есть в зависимостях в файле requirements.txt.
- ☐ Убедитесь, что приложение (или приложения, если их несколько) было добавлено в `INSTALLED_APPS` в файле settings.py.
- ☐ Убедитесь, что в `INSTALLED_APPS` сначала были подключены стандартные приложения, а потом все остальные. Вот список стандартных приложений:
  - django.contrib.admin,
  - django.contrib.auth,
  - django.contrib.contenttypes,
  - django.contrib.sessions,
  - django.contrib.messages,
  - django.contrib.staticfiles.
- ☐ Убедитесь, что в репозитории есть все миграции. Как создать миграции:

```
python3 manage.py makemigrations
```

- ☐ Проверьте, как указаны пути в файле settings.py. Пути вида `BASE_DIR / 'static'` могут вызывать ошибку в последних версиях Django в сочетании с Python 3.9

```
TypeError: argument of type 'PosixPath' is not iterable
```

Если у вас возникает такая ошибка, укажите пути иначе:

```
# Первый вариант.  
str(BASE_DIR / 'static')  
# Второй вариант.  
os.path.join(BASE_DIR, 'static'),
```