

Работа с вашим репозиторием. ChatOps. Интеграция с чатом

Требования

Необходимо выполнить первое домашнее задание и принять приглашения в организацию на GitHub.

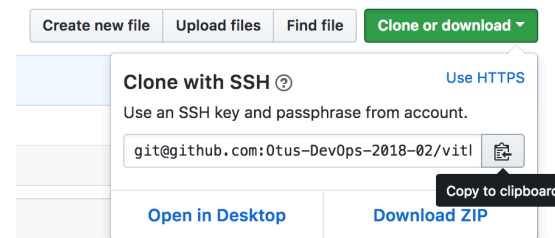
План

1. Закрепляем знания по Git, работаем в своих репозиториях, созданных после выполнения ДЗ№1
2. Создаем интеграцию с чатом для вашего репозитория и подключение TravisCI

Если вы не работали с Github вообще или не ориентируетесь в его интерфейсе, просмотрите слайды к этому занятию из ЛК - "Работа с интерфейсом в Github"

Работа с репозиториями в Otus

- Найдите свой репозиторий в организации [Otus-DevOps-2019-02](#)
- Нажмите **Clone or download**
- Ваш репозиторий должен называться `<GITHUB_USER>_infra` (где `<GITHUB_USER>` - ваш логин на GitHub)
- Нажмите на кнопку с планшетом для бумаг **Copy to clipboard**
- Введите в терминале команду, подставив URL своего репозитория:



```
$ git clone git@github.com:Otus-DevOps-2019-02/<GITHUB_USER>_infra.git
```

Работа с ветками

Перейдите в директорию своего репозитория:

```
$ cd <GITHUB_USER>_infra
```

Создайте ветку, требуемую в слайдах ДЗ:

```
$ git checkout -b play-travis
```

Добавление изменений

Для оформления текста в Pull request этого и следующих заданий, мы воспользуемся функционалом [Pull Request Templates](#) от Github.

После первого коммита, в дальнейшем он будет автоматом применяться для ваших PR. Просьба пользоваться им и заполнять при выполнении домашних заданий

Добавление изменений

1. Создайте директорию `.github`:

```
$ mkdir .github
```

2. Перейдите в директорию `.github`:

```
$ cd .github
```

3. Скачайте внутрь этого каталога файл `PULL_REQUEST_TEMPLATE.md` ([ссылка на файл](#))

```
$ wget http://bit.ly/otus-pr-template -O PULL_REQUEST_TEMPLATE.md
```

Отправка изменений

1. Добавьте файл `PULL_REQUEST_TEMPLATE.md` в индекс Git:

```
$ git add PULL_REQUEST_TEMPLATE.md
```

2. Сделайте коммит

```
$ git commit -m 'Add PR template'
```

3. Отправьте изменения на github

```
$ git push --set-upstream origin play-travis
```

Настройка интеграций используемых сервисов с чатом

! Это важно! Для выполнения этого практического задания мы создали **отдельный тестовый чат Slack**.

Пожалуйста, зарегистрируйтесь в Slack чате по следующей [ссылке](#).

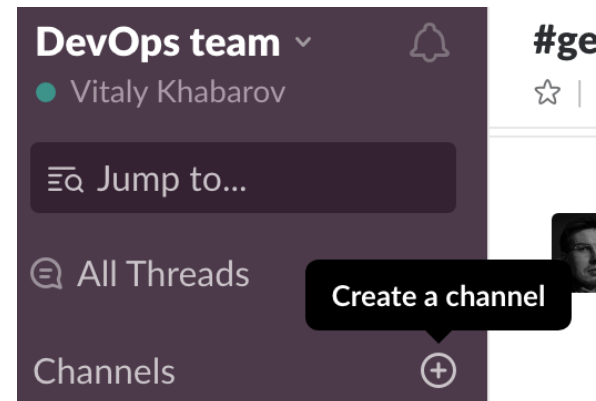
Пожалуйста, **не** используйте чат для общения с преподавателями! Он только для выполнения данной самостоятельной работы!

Создание канала

1. Создайте канал Slack.
Название должно быть в формате **<имя>_<фамилия>**.
2. Для последующей проверки ДЗ, пригласите в канал преподавателей:

@vitkhab

@mmikhaylov



Create a channel

Channels are where your members communicate. They're best when organized around a topic — #leads, for example.

Public Anyone in your workspace can view and join this channel.

Name

vitaly_khabarov

Names must be lowercase, without spaces or periods, and shorter than 22 characters.

Purpose (optional)

Канал для интеграций

What's this channel about?

Send invites to: (optional)

Search by name

Select up to 1000 people to add to this channel.


Cancel

Create Channel

Интеграция с GitHub

Мы уже *подключили* интеграцию нашей организации GitHub к нашему тестовому чату Slack.

В общем случае нужно создать интеграцию [по инструкции](#).

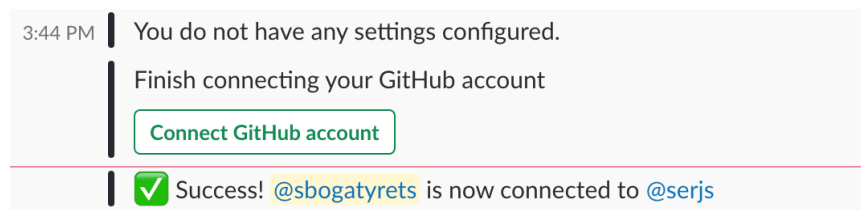
 Вам ее выполнять *не нужно*, это была справочная информация.

Интеграция с GitHub

Наберите в своем канале Slack команду-сообщение:

```
/github subscribe Otus-DevOps-2019-02/<GITHUB_USER>_infra commits:all
```

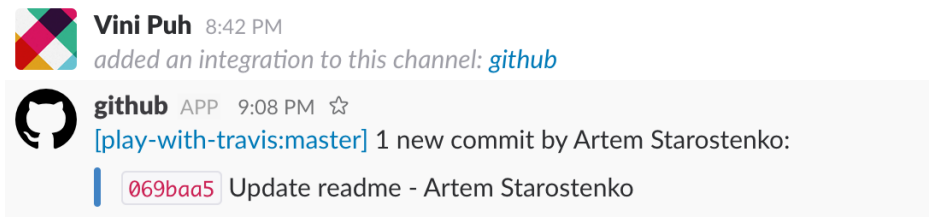
GitHub может запросить пройти авторизацию с выдачей прав на чтение информации из репозитория. Выдаем в процессе права, если требуется.



Приложение GitHub подтвердит создание интеграции с вашим репозиторием

Тестируем интеграцию

- Создайте директорию `play-travis` в корне репозитория
- Скачайте `test.py` в директорию `play-travis` ([ссылка](#))
- Сделайте коммит этого файла в ветку `play-travis` вашего репозитория на GitHub
- Проверьте, что в ваш канал приходят уведомления о новых коммитах



Travis CI

Это бесплатный сервис непрерывной сборки и тестирования для проектов, размещенных на GitHub

Мы будем использовать его для проверок ваших заданий тестами, а также в текущем ДЗ для интеграции статусов коммитов с чатом.

Инструкции по сборке

Все шаги по тому, как производить сборку и тестирование нашего проекта описываются в файле `.travis.yml`, расположенном в корне репозитория

Создайте в корне репозитория файл `.travis.yml` с содержимым:

```
dist: trusty
sudo: required
language: bash
before_install:
  - curl https://raw.githubusercontent.com/express42/otus-homeworks/2019-02/run.sh |
  bash
```

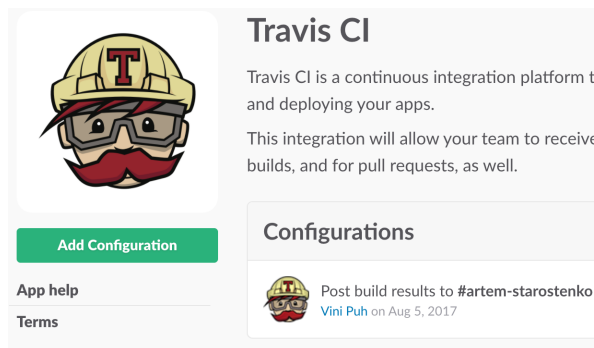
Теперь Travis CI, будет запускать билд при каждом новом коммите. Но прежде, чем мы это протестируем, нужно настроить отправку уведомлений о статусе сборки в чат.

Интеграция Slack с Travis CI

- Перейдите в управление приложениями в Slack'e (+ **Add apps** внизу панели)
- В списке найдите TravisCI и нажмите кнопку **View**
- Нажмите кнопку **Settings** рядом с лого TravisCI (это блок *About this app*)

Интеграция Slack с Travis CI

- Создайте новую конфигурацию нажав на кнопку **Add configuration**



- Выберите канал для уведомлений, который вы создали:

Post to Channel

Start by choosing a channel where Travis CI build results will be posted.

Add Travis CI integration

Хранение секретов

Хранение секретов в репозиториях всегда плохая идея. Особенно недопустимо их хранение в публичных репозиториях в незашифрованном виде!

При настройке интеграции нам выдают токен для записи в `.travis.yml`. Мы воспользуемся функционалом Travis CI для шифрования токена, чтобы спокойно его закоммитить в наш публичный репозиторий.

Детали на следующем слайде 

Хранение секретов

- Для использования команды `travis`, вам придется предварительно установить `gem` (требуется `ruby` и `rubygems`) при помощи команды:

```
$ gem install travis
```

- Затем надо авторизоваться через утилиту `travis`:

```
$ travis login --com
```

Хранение секретов

- Запоминаем полученный токен из страницы с интеграцией:

Token

This token is used as the key to your Travis CI integration.

3GMgXnY0ezblEiJXW7PsJ45x

[Regenerate](#)

- Выполните команду из консоли с шифрованием вашей интеграции

```
$ travis encrypt "devops-team-otus: <ваш_токен>#<имя_вашего_канала>" \
  --add notifications.slack.rooms --com
```

Хранение секретов

В результате ваш файл `.travis.yml` должен выглядеть подобным образом:

```
dist: trusty
sudo: required
language: bash
before_install:
- curl https://raw.githubusercontent.com/express42/otus-homeworks/2019-02/run.sh | bash
notifications:
  slack:
    rooms:
      secure:
N0aYWn5sLoZJkEW4kRsU8BwLvDzEsGALlgNsFlZntmq++S/YoHckcHQjvgLYJ0FhdkgKEEkX8aHaDTXqyDjhmg4c
+teQziqVyF+Qy+Nh7iz58D/R//m6243VMgIx+SHcw7+ghAqbEejZYV/P1kS9W9rFW6k16YjlSRjgMiavgx
+cu14Xrs5Amy3eNVnFevK5aKIJrjcrvrA6VcZX/+jCcobUknpZqKL iIVmpH+5T5NbD74BX/v9ceob1jYAz6w8hQt
+/sQQ5sJWx+kSi+qY56dJoVrit9w30mxuM3Lo9iZYn8zAse7XFyKQnSHi06iI9IInalzwOgDfanb0Vgvs7m
+xCN9UmPR2Ce31Aq7dQQ7ajOF++qETvX5B1jUuKa+z1H5VsYirRZ+4fKjQ6BNy+Xr=
```

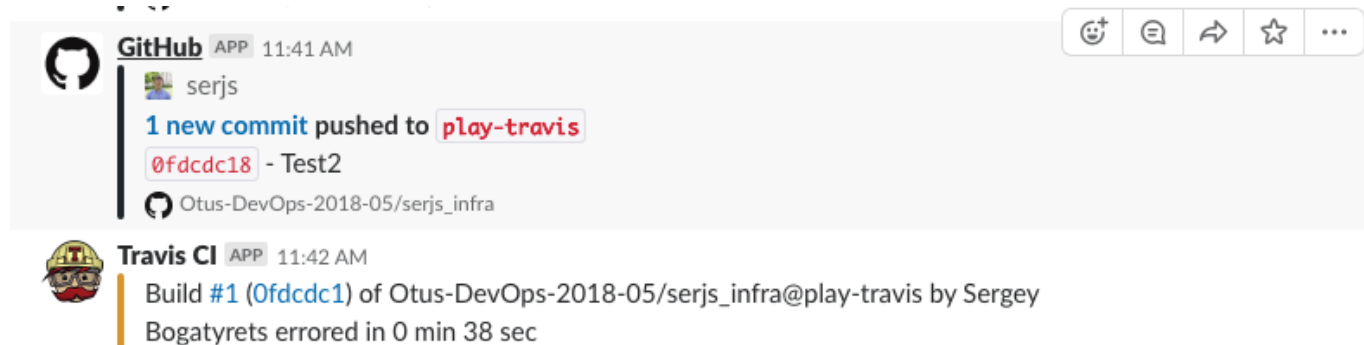
Тестируем интеграцию

- Проверьте, какие строки добавились в файл `.travis.yml` после выполнения предыдущей команды
- Сделайте коммит изменений `.travis.yml` в репозиторий на GitHub.
- Это должно запустить билд Travis CI, информация о котором будет доступна на странице Travis-a:

```
https://travis-ci.com/Otus-DevOps-2019-02/<githubusername>_infra
```

Тестируем интеграцию

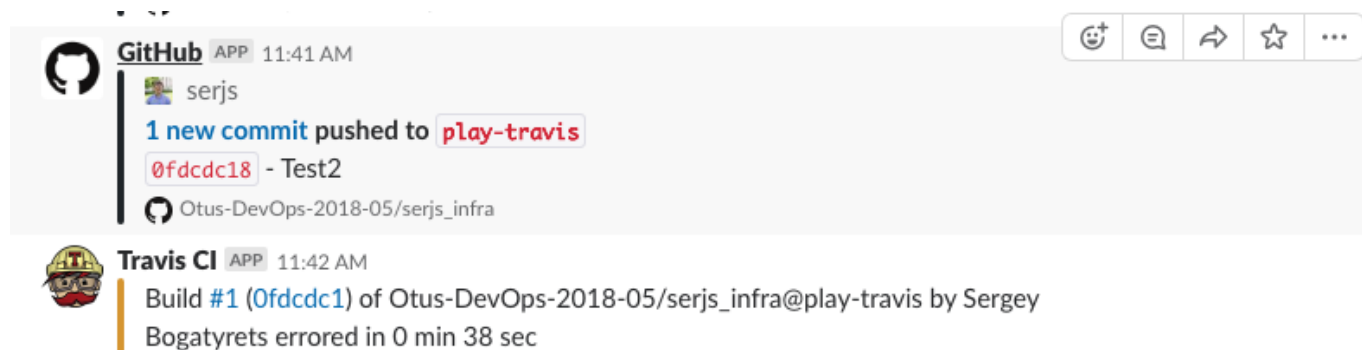
- В нашем канале Slack должна появиться информация о статусе билда после его окончания



- Билд должен провалиться. Посмотреть информацию об ошибке можно тыкнув на номер билда и перейдя на страницу сборки Travis CI.

Самостоятельное задание: почините тесты

- Учитывая лог билда из Travis CI, разберитесь, где ошибка и исправьте ее
- Нужно добиться успешного (зеленого) билда:



Проверка

- Создайте PR из ветки **play-travis** в **master**. Проставьте на Pull request Label **play-travis**
- В PR добавлять в Reviewers *никого не нужно*
- Проверьте, что ваш профиль **GitHub** привязан в личном кабинете **ОТУС**
- В вашем PR должны быть следующие файлы
 - `.github/PULL_REQUEST_TEMPLATE.md`
 - `.travis.yml`
 - `play-travis/test.py`

Итого

- Рассмотрели работу с интеграцией на примерах
- Вы также подключили свои репозитории к тестам в Travis для помощи в проверке последующих ДЗ 😊
- Добавили шаблон для последующих PR в GitHub (будет работать начиная со следующего ДЗ)

