В разработке программного обеспечения непрерывная интеграция – это практика частой сборки и тестирования проекта с целью выявления ошибок на ранней стадии интеграции проекта. Непрерывная интеграция — автоматизированный процесс, в котором, как правило, используется специализированное серверное ПО, отвечающее за поиск изменений в коде в системе контроля версий, сборку, развертывание и тестирование приложения. Эти действия система непрерывной интеграции совершает автоматически при каждом обновлении репозитория системы контроля версий [http://lib.custis.ru/Непрерывная\_интеграция].

Поскольку в качестве системы контроля версий для данного курсового проекта выбрана система Git с расположением репозитория проекта на бесплатном (для публичных репозиториев) хостинге GitHub, то в качестве системы непрерывной интеграции выбран удобный для таких задач сервис Travis CI. Travis CI – это распределенный веб сервис для сборки и тестирования проектов по разработке программного обеспечения на более чем 20-ти языках программирования, использующих GitHub в качестве хостинга исходного кода [https://ru.wikipedia.org/w/index.php?title=Travis\_CI&stable=1].

Процесс настройки системы непрерывной интеграции и первая сборка проекта представляет собой простую последовательность действий, описанную ниже. Для того, чтобы осуществить привязку своего репозитория на GitHub необходимо быть хозяином этого репозитория.

Первым шагом является вход в учетную запись Travis CI через аккаунт на GitHub. Для этого необходимо с сайта сервиса (<https://travis-ci.org/>) нажать на кнопку, представленную на Рис. 1.

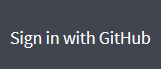


Рис. 1. Кнопка входа в учетную запись Travis CI

Следующим шагом является активация синхронизации репозитория проекта на GitHub с сервисом Travis CI. Для этого необходимо зайти в настройки репозиториев профиля на Travis CI и включить работу сервиса с выбранным репозиторием с помощью переключателя, изображенного на Рис. 2

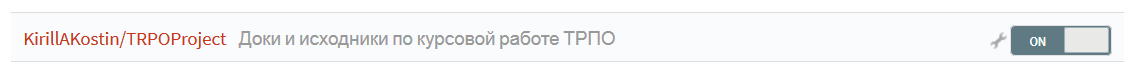


Рис. 2. Переключатель Travis CI для работы с выбранным репозиторием

Следующим шагом, указывающим сервису Travis то, как работать с репозиторием, является добавление в корневую директорию репозитория файла .travis.yml. Данный файл содержит скрипт, указывающий сервису то, как настроен репозиторий проекта: на каком языке написан исходный код, что нужно выполнить перед сборкой проекта, какие скрипты нужно выполнить после сборки проекта и другие опции, о которых можно узнать на сайте Travis CI [http://docs.travis-ci.com/user/languages/python/].

Содержание файла .travis.yml для данного курсового проекта представлено ниже.

language: python

python:

- "3.4"

install: "pip install -r requirements.txt"

script:

- "flake8"

- "coverage run manage.py test"

- "coverage report"

notifications:

email:

on\_failure:

- kiraarik@sibmail.com

Как видно из содержания файла, в нем указывается, что проект написан на языке python. Это необходимо для того, чтобы Travis смог правильно интерпретировать проект и создать необходимое виртуальное окружение на сервере для сборки проекта. Далее указывается версия интерпретатора python, необходимая для выполнения установок и скриптов на сервере.

После этого указываются секции, которые выполняет Travis CI в проекте. В секции install он выполняет скрипт по установке необходимых зависимостей для корректной работы программной системы. В секции scripts указываются скрипты, которые также необходимо выполнить после сборки проекта и установки всех зависимостей. Да данном случае выполняется скрипт flake8, проверяющий стиль написания кода на python (настройки для выполнения данного скрипта находятся в файле проекта tox.ini), затем запускаются тесты и ведется подсчет процента покрытия кода тестами с помощью команд coverage.

При возникновении ошибок сборки Travis CI высылает на почту пользователя, который сделал коммит, на котором возникла ошибка, уведомление о том, что его коммит привел к нарушению работы проекта. В файле Travis CI данного проекта также указывается e-mail, на который, в случае возникновении ошибок, всегда будет высылаться уведомление.

После настройки файла .travis.yml для проверки того, работает ли Travis CI с репозиторием необходимо зайти в настройки репозитория, во вкладку «Webhooks and Services» и выбрать сервис Travis CI, на открывшейся странице нужно нажать на кнопку «Test service» (Рис. 3), чтобы убедится в работоспособности сервиса с вашим репозиторием.



Рис. 3. Тестирование сервиса Travis CI на GitHub

Сама по себе такая проверка не инициирует сборку проекта. Для запуска построения проекта необходимо сделать коммит и отправить его в репозиторий на сервер GitHub.

Увидеть то, как прошла сборка и логи сборки можно на сайте Travis CI, зайдя под своей учетной записью. Если все команды завершились с кодом 0, то сборка проекта считается выполненной успешно и её присваивается статус «Passed», в противном случае ей присваивается статус «Failed».