

## README.md

## Задание

### uWSGI daemon

*Задание:* Написать uWSGI демона (для CentOS 7), который по запросу на IPv4 адрес возвращает текущую погоду в городе, к которому относится IP, в виде json

- <https://ipinfo.io/developers>
- <https://openweathermap.org/current>
- запросы на демон должен проксировать nginx, демон должен запускаться через systemd
  - <https://www.digitalocean.com/community/tutorials/how-to-set-up-uwsgi-and-nginx-to-serve-python-apps-on-centos-7>
  - в докере systemctl не работает из коробки, так что протестировать запуск через systemd там не получится (если не найдете какой-нибудь workaround)
- демон должен собираться в пакет
  - [https://www.centos.org/docs/5/html/Deployment\\_Guide-en-US/s1-filesystem-fhs.html](https://www.centos.org/docs/5/html/Deployment_Guide-en-US/s1-filesystem-fhs.html)
  - бинарь (исполняемый файл(ы)) должен лежать в отдельной папке внутри /usr/local/
  - конфиг должен лежать в /usr/local/etc/
  - лог должен писаться в отдельную папку внутри /var/log/
  - см. шаблон ip2w.спис, там троеточием отмечены места, где нужно что-то дописать
    - зависимости просто указать, не нужно их собирать отдельно
  - чтобы собрать пакет используйте (скрипт предполагает, что код закоммичен в git)

```
./buildrpm.sh путь/до/ip2w.спис
```

- у демона должен быть конфиг, демон должен писать лог (имеется в виду не uwsgi конфиг и лог)
- должны быть написаны, как минимум, функциональные тесты

```
curl http://localhost/ip2w/176.14.221.123  
{ "city": "Moscow", "temp": "+20", "conditions": "небольшой дождь" }
```

### Что на выходе:

1. скрипт
2. конфиг uwsgi
3. .service файл для systemd
4. .спис файл для rpm пакета
5. собранный rpm пакет
6. конфиг nginx, где описан локейшн ip2w

*Цель задания:* получить навык создания демонов, соответствующих требованиям продакшена, ознакомиться с редко освещаемыми аспектами разработки, которые так или иначе играют важную роль в жизни каждого программиста.

*Критерии успеха:* задание **опционально**, критерием успеха является работающий согласно заданию код, который проходит тесты, для которого проверено соответствие пер8, написана минимальная документация с примерами запуска. Далее успешность определяется code review.

## Django tutorial

*Задание:* пройти <https://docs.djangoproject.com/en/1.11/intro/tutorial01/>.

*Цель задания:* познакомиться с фреймворком, получить навыки и представления о его использовании, чтобы в дальнейшем лучше усваивать материал занятий.

*Критерии успеха:* задание **опционально**, критерием успеха является целиком пройденный tutorial.

## Deadline

---

Задание нужно сдать через неделю. То есть ДЗ, выданное в понедельник, нужно сдать до следующего занятия в понедельник. Код, отправленный на ревью в это время, рассматривается в первом приоритете. Нарушение дедлайна (пока) не карается, но может повлиять на ранжирование при выборе топа студентов при окончании курса, пытаться сдать ДЗ можно до конца курса. Но код, отправленный с опозданием, когда по плану предполагается работа над более актуальным ДЗ, будет рассматриваться в более низком приоритете без гарантий по высокой скорости проверки

## Обратная связь

---

Студент коммитит все необходимое в свой github/gitlab репозиторий. Далее необходимо зайти в ЛК, найти занятие, ДЗ по которому выполнялось, нажать "Чат с преподавателем" и отправить ссылку. После этого ревью и общение на тему ДЗ будет происходить в рамках этого чата.