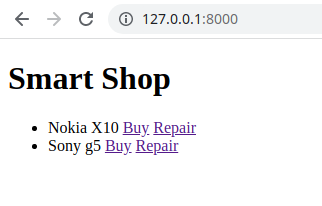
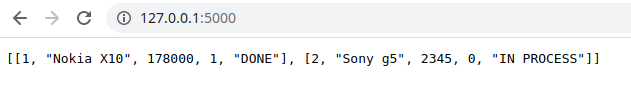
У нас имеется 3 микросервиса. Первый работает на Django, второй на Flask, третий на python3. Роль первого быть магазином для клиента. Роль второго заниматься заказами. Роль третьего передавать о процессе ремонта. Тематика смартфоны. Клиент в магазине может либо заказать телефон либо заказать ремонт в магазине. Я стартую приложение на Django, запускаю его в браузере – <http://127.0.0.1:8000/>.

У нас появляется данное окно.



Тут мы можем выбрать либо купить либо починить телефон. Нам доступны два на выбор. Купим Nokia (Buy) и починим Sony (Repair). После нажатия отправятся два post запроса для второго приложения. После этого нам нужно будет обновить страницу с заказами – <http://127.0.0.1:5000/>.

У нас получиться такой вывод.



Мы видим что у товаров разный статус. У телефона, который мы купили DONE, а у телефона который мы чиним статус In PROCESS. Один готов, а другой в процессе. В процессе у нас ремонт телефона. Вернее телефон отремонтируется не сразу поэтому он в процессе находится.  
  
Теперь взглянем на третье запущенное наше приложение на питоне.

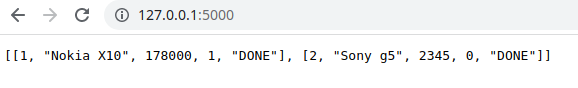


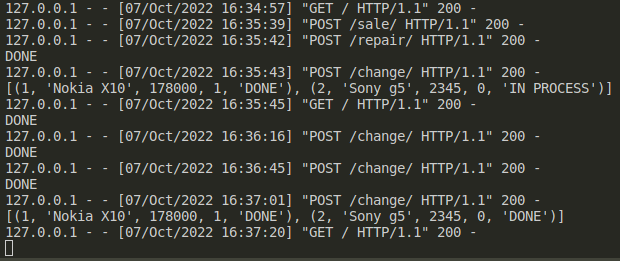
Перед тем как я его запустил я выполнил следующие команды:

sudo apt-get update

sudo apt-get install rabbitmq-server

Я установил rabbitmq-server это брокер сообщений, работающий на протоколе AMQP. Он нужен для того чтоб данные можно было принимать из очереди сообщений. Для того чтоб питон мог работать с брокером сообщений у нас есть модуль pika. Вьюха repair работающая с урлом /repair/ при помощи модуля pika отправляет данные в очередь rabbitmq. В третьем микросервисе есть функция callback, которая после ожидания (имитации ремонта телефона) делает post запрос на обновление данных и там меняется статус данных In PROCESS на DONE потому что это означает что телефон починен. Функционал из модуля pika подключает функцию callback к очереди repair.  
  
Обновим страницу <http://127.0.0.1:5000/>.

Видим что статус изменился у Sony на DONE. 

Посмотрим на запросы микросервиса с заказами.  
  
  
Мы видим все запросы к микросервису и видим изменение в статусе Sony телефона.  
  
Вывод: я на 9 уроке познакомился с микросервисами, установил нужные библиотеки для работы, установил брокер сообщений, запустил все три микросервиса, посмотрел как они взаимодействуют между собой (путем отправки друг другу запросов post и работой с очередью rabbitmq).

Пункт по выполнению самого курса:

Проект сейчас работоспособен. Долгов по теме у меня нет. Сейчас работаю над его расширением до уровня хорошего портфолио. Исправил косяк с сохранением форм теперь у меня если none в имени, то происходит raise ошибки. То есть я исправил антипаттерн. Планирую починить настроить загрузку статики, сделать выдвигающуюся админ панель. Сейчас работаю над мапперами для некоторых оставшихся моделей, которые работали без бд до этого.