Изоляция инфраструктуры E2E тестов. Или "Ходим под себя"

Борис Бенгус
iOS Lead в Dostavista
tg: @bengus

#симулятор vs девайс

#selenium

#appium

#espresso

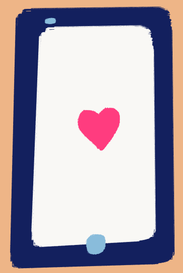
#cucumberish

#xcuitest

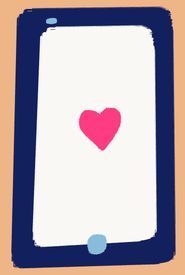
#mocks generation

#cucumber

#calabash

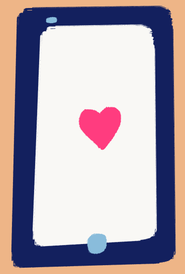


Вася - Разработчик

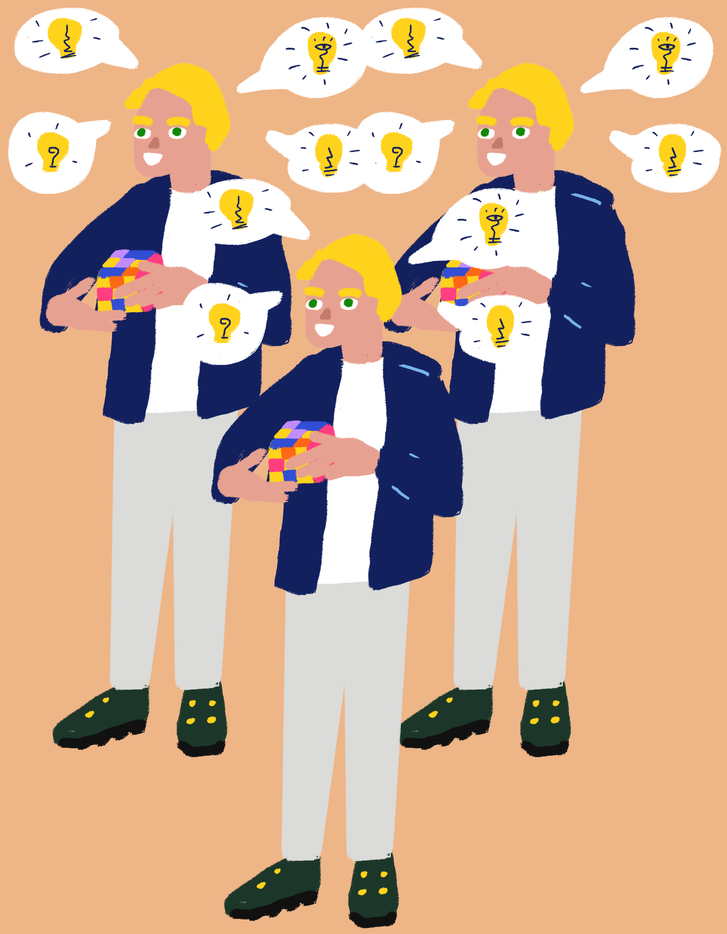


Вася - Разработчик

Product



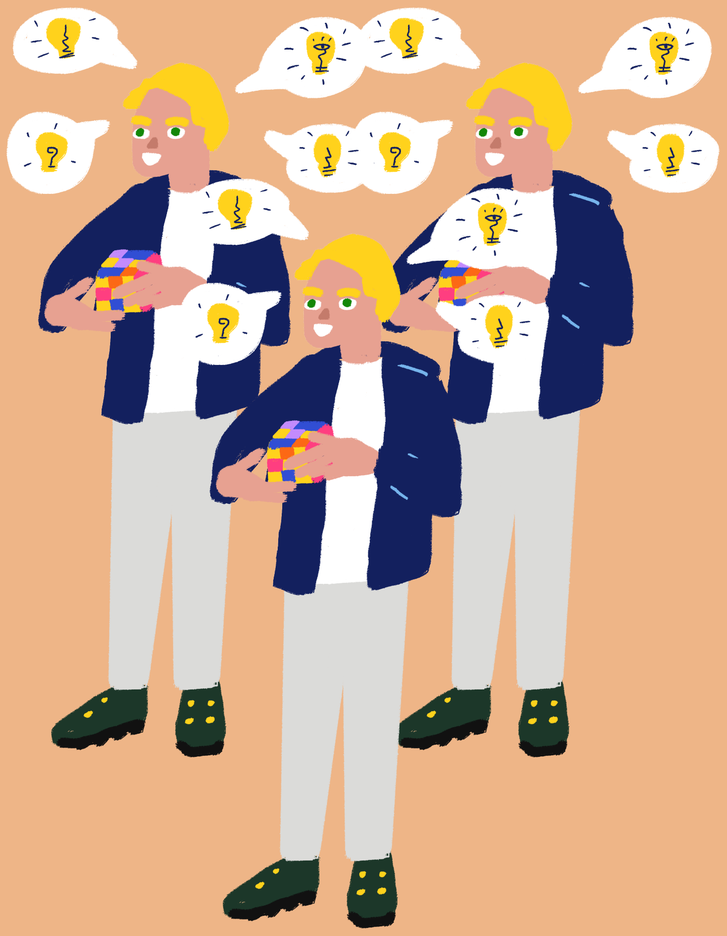
Вася - Разработчик



Product

Вася - Разработчик

QA



Product

Цели, требования

⏳ Time-to-Market

🕰️ Времени полного регресса

🚀 Ничего не сломано в релизе

⏱️/💵 Время VS ресурсы

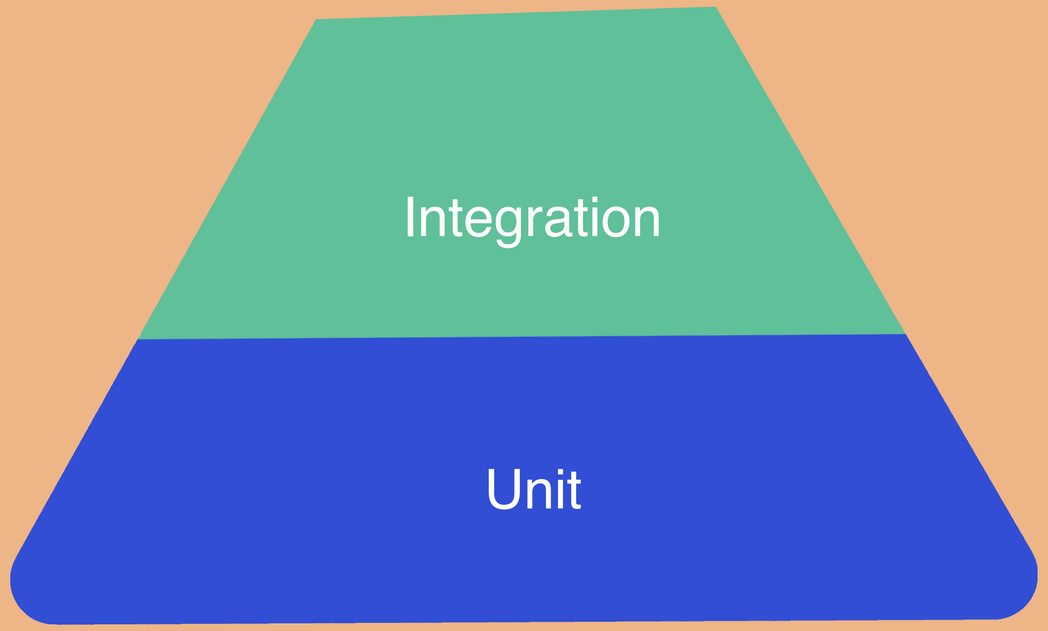
👨‍🔧 Ручные QA -> Автоматизаторы

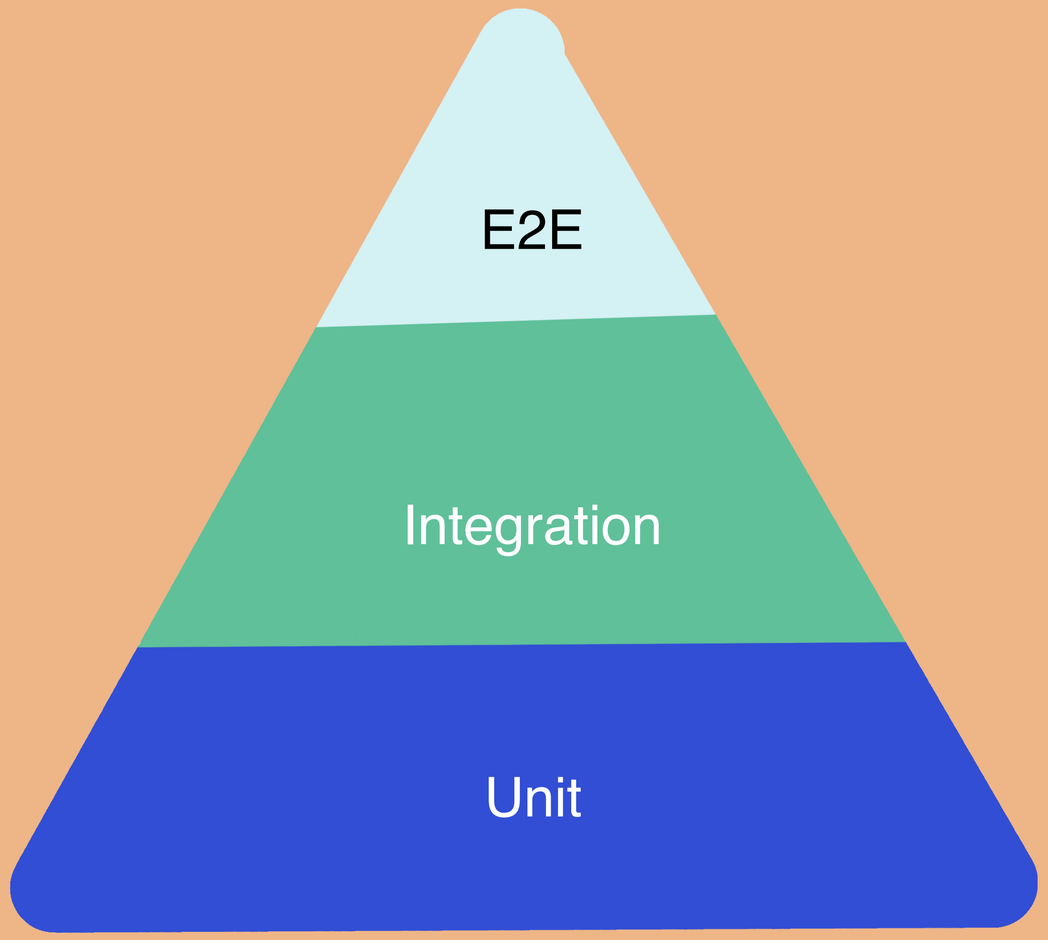
Чеклист

🎯 Цель, ресурсы, результат

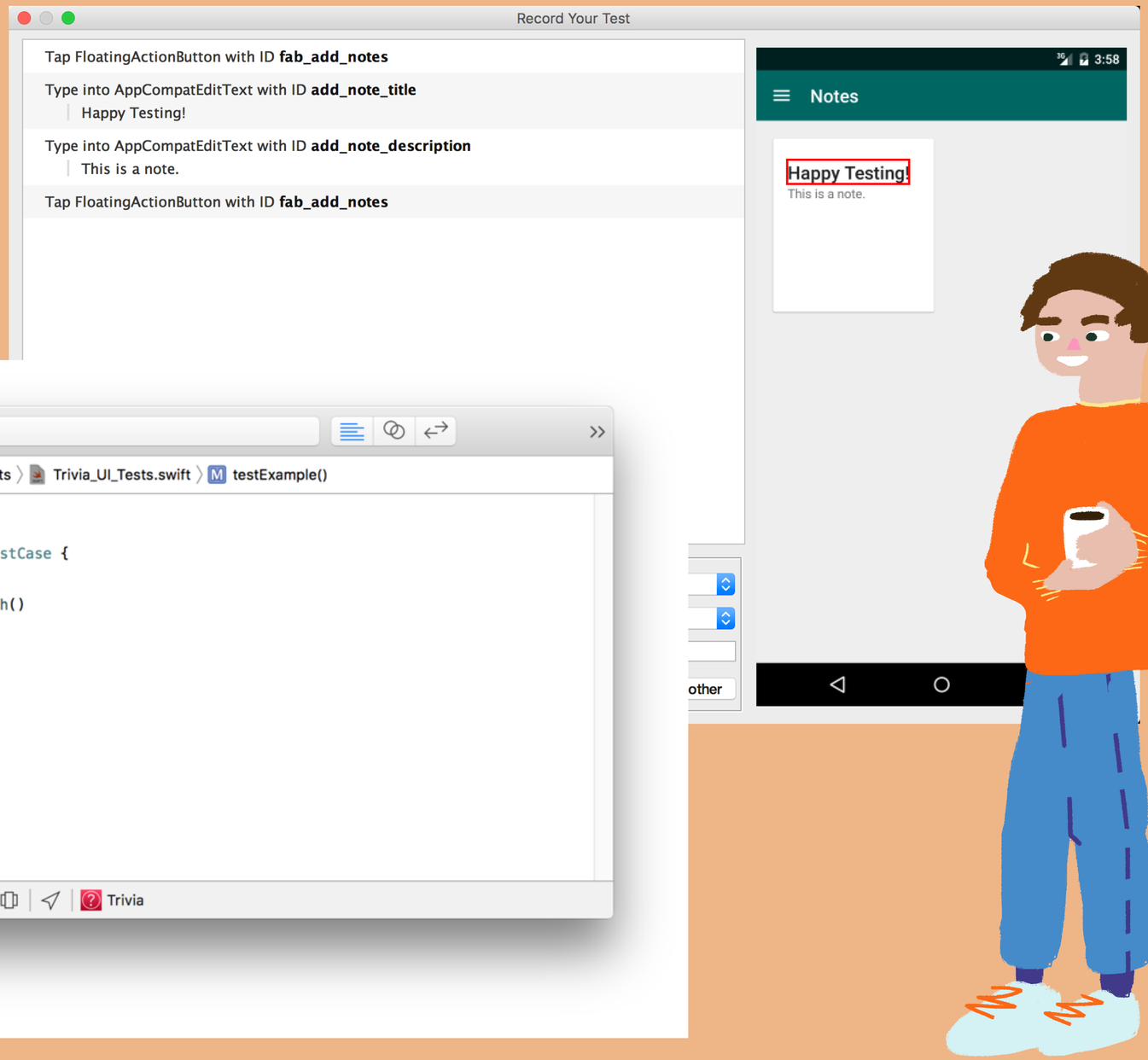
...

Unit


Integration
Unit


E2E
Integration
Unit

Record 🔴


Record Your Test
Tap FloatingActionButton with ID fab_add_notes
Type into AppCompatEditText with ID add_note_title
Happy Testing!
Type into AppCompatEditText with ID add_note_description
This is a note.
Tap FloatingActionButton with ID fab_add_notes
3:58
Notes
Happy Testing!
This is a note.
Running Trivia on iPhone 6
Trivia
Trivia UI Tests
Trivia_UI_Tests.swift
testExample()
Trivia
Trivia
AppDelegate.swift
Main.storyboard
GameNavigationController.swift
GameNavigatio...rDelegate.swift
Splash Screen
Genre Picker
GenrePickerVi...ontroller.swift
GenreCollectionViewCell.swift
Player Picker
PlayerPickerV...ontroller.swift
PlayerPickerT...ViewCell.swift
Lobby View Controller
Model
Search
Trivia
import XCTest
class TriviaGame_UI_Tests: XCTestCase {
override func setUp() {
XCUIApplication().launch()
}
func testExample() {
}
}
Record button

Чеклист

🎯 Цель, ресурсы, результат

...

Чеклист

🎯 Цель, ресурсы, результат

🎲 Для начала сделай как нибудь

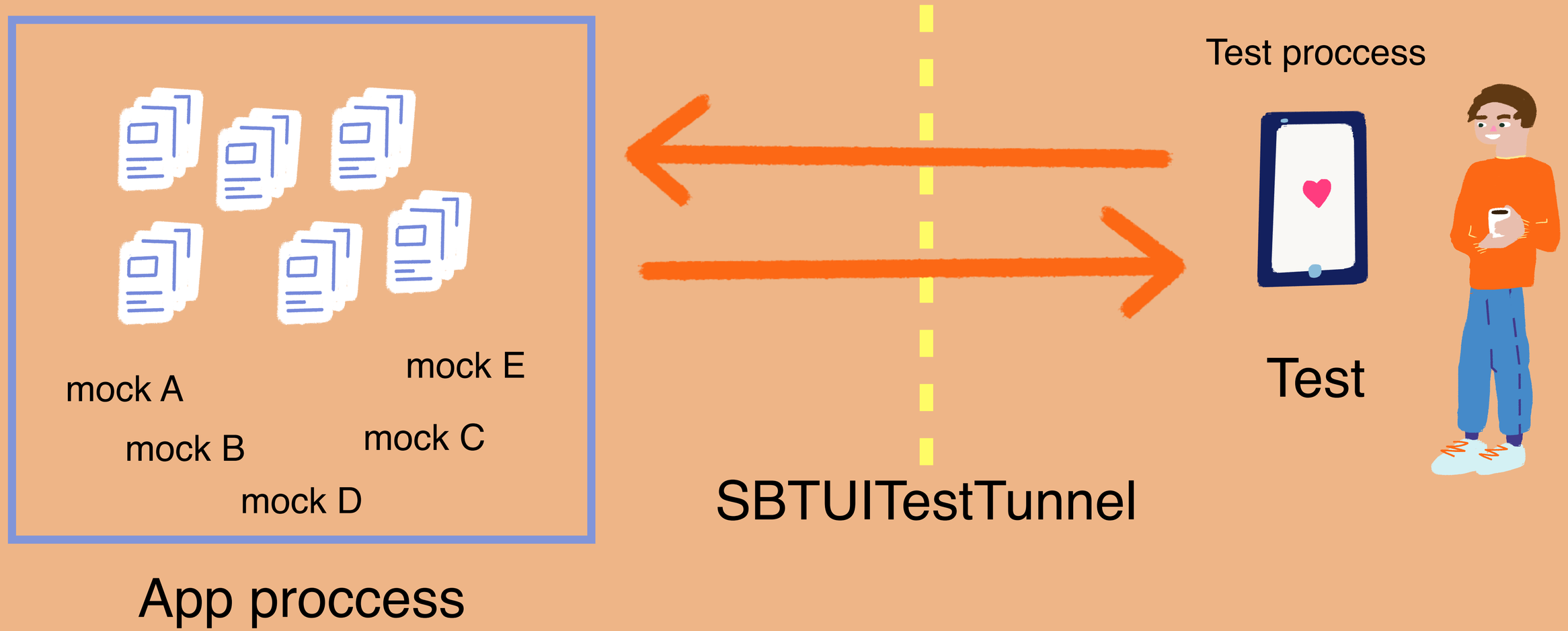
...

Проблемы

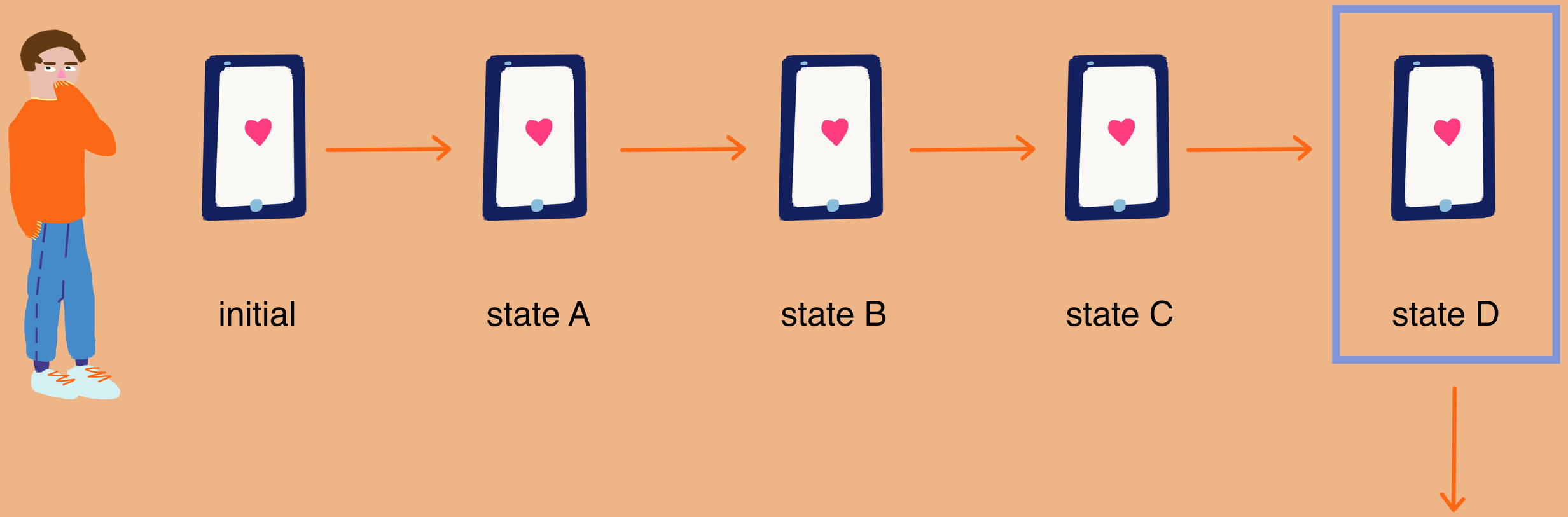
- Mock/stub VS реальный бэкенд?
- Подглядывание в black box
- Флаканье
- Автономность (Workstation/CI)
- Воспроизводимость
- Let’s localise it

Mock/stub VS реальный backend

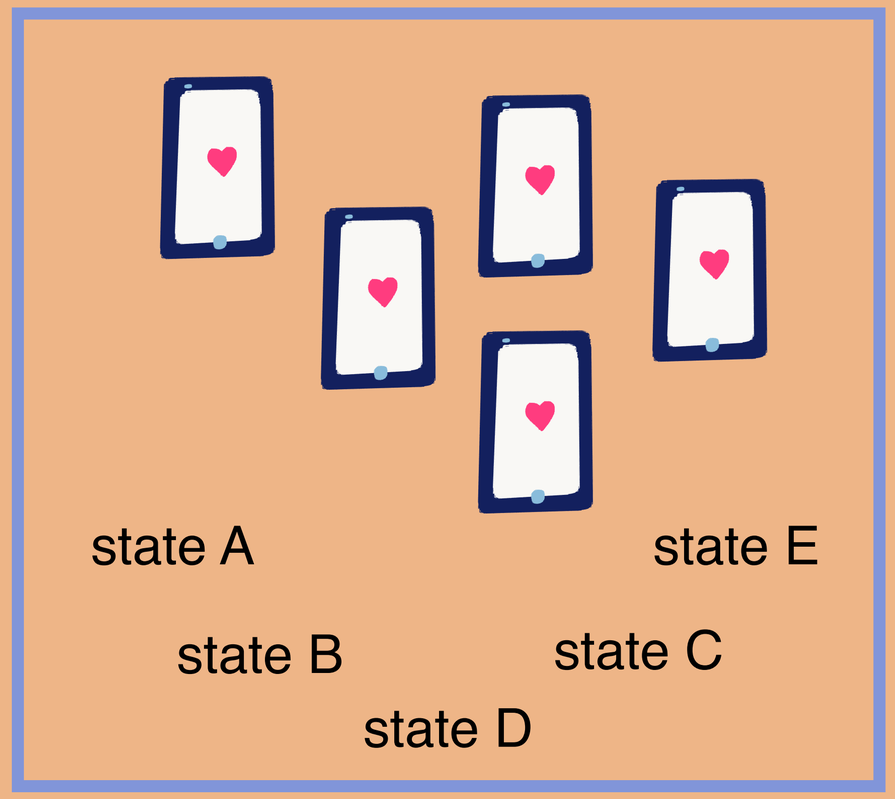
Mocks in code



App state recording 🔴

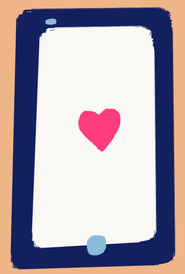


App state recording 🔴



saved states 💾

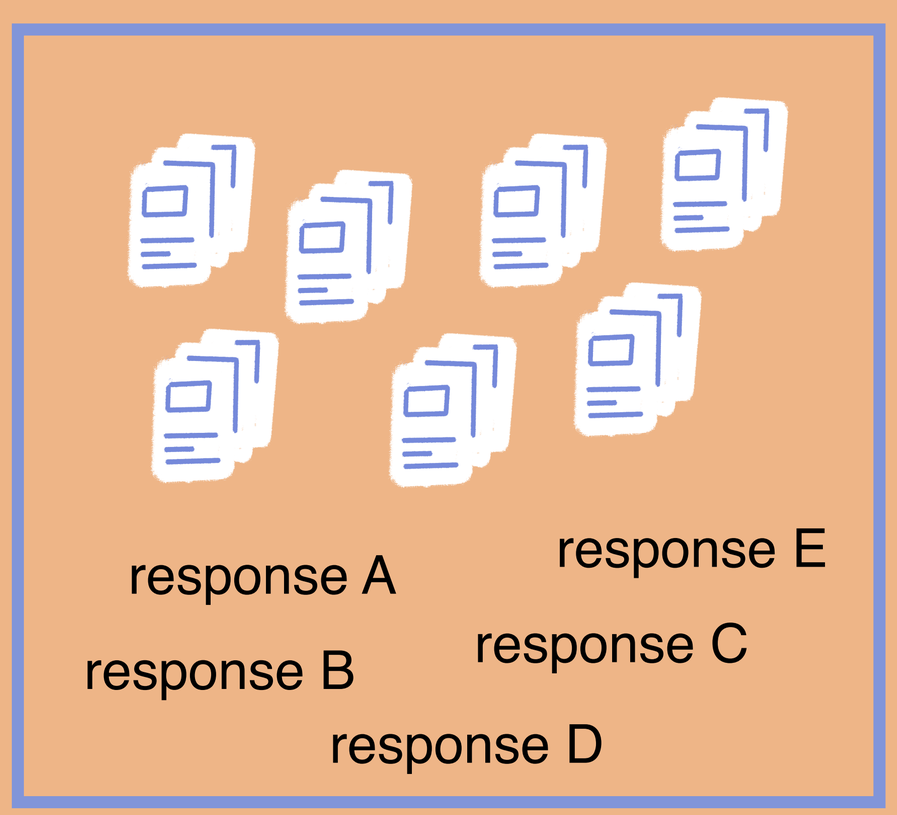
restore



state D

Test

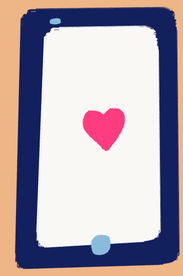
Local mock-proxy



response mocks

request

mocked
response

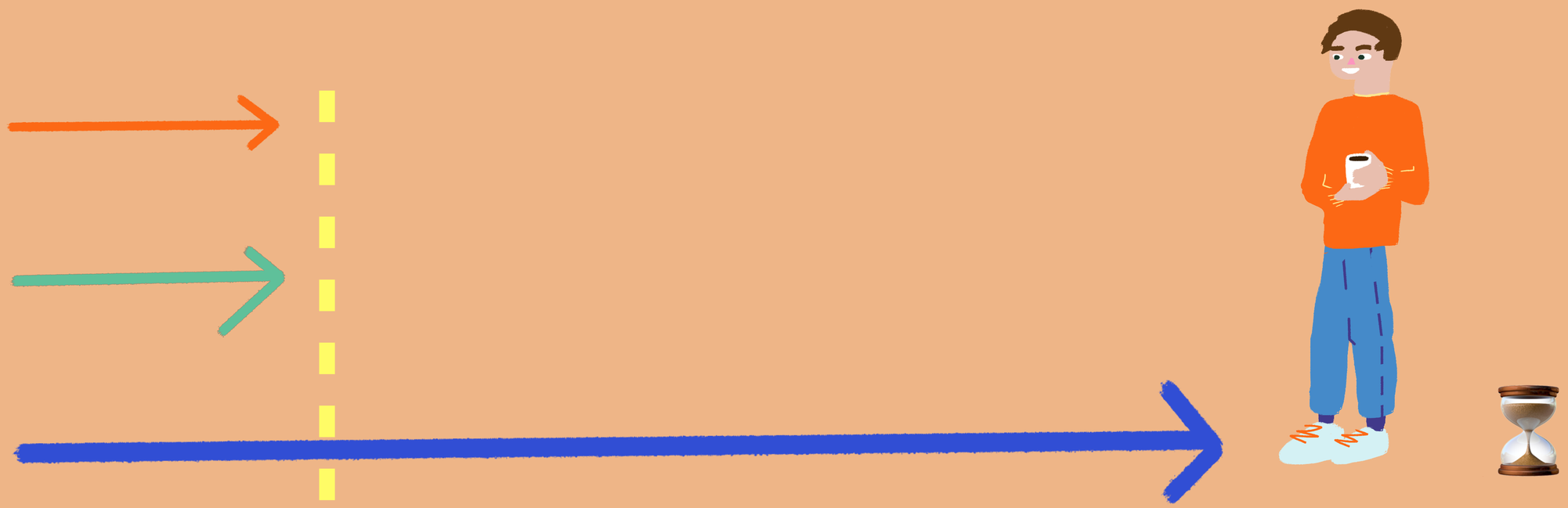


Test

Мокать или не мокать?

mocks

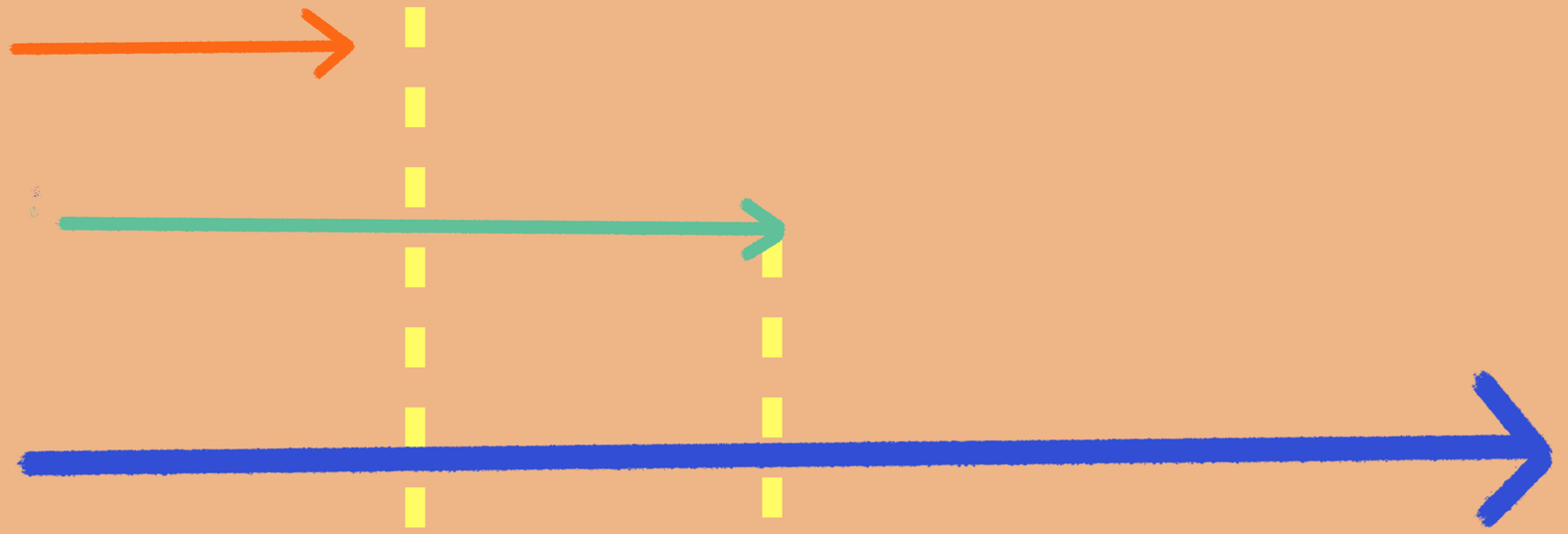
backend



Мокать или не мокать?

mocks

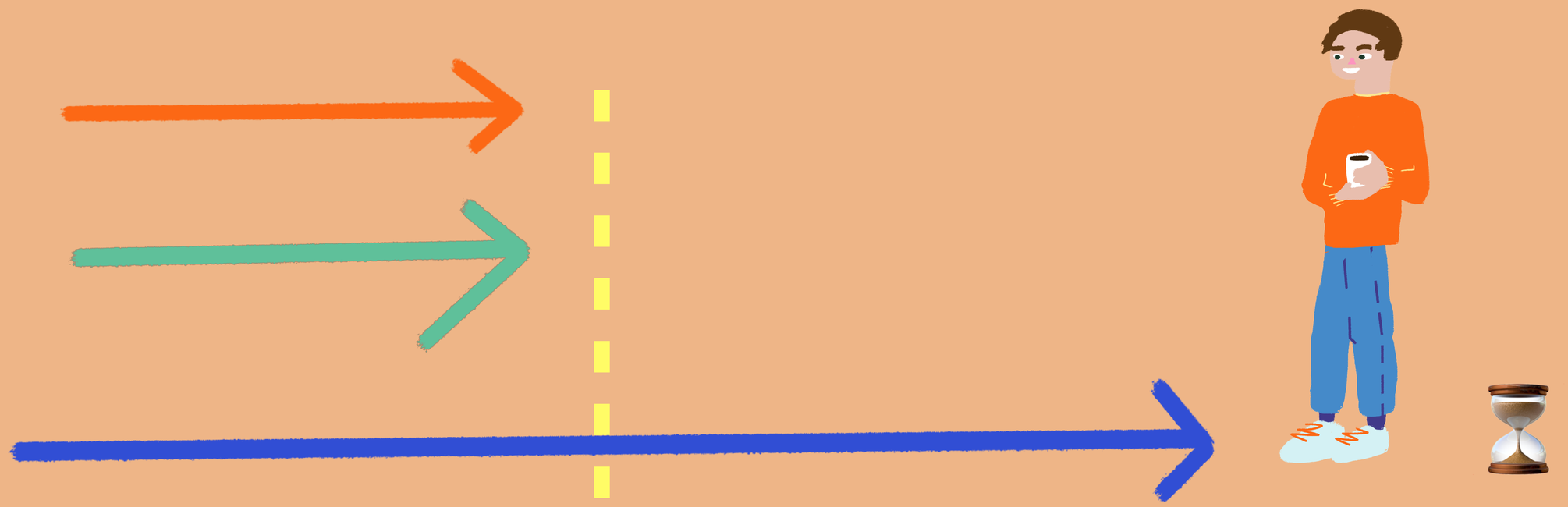
backend



Мокать или не мокать?

mocks

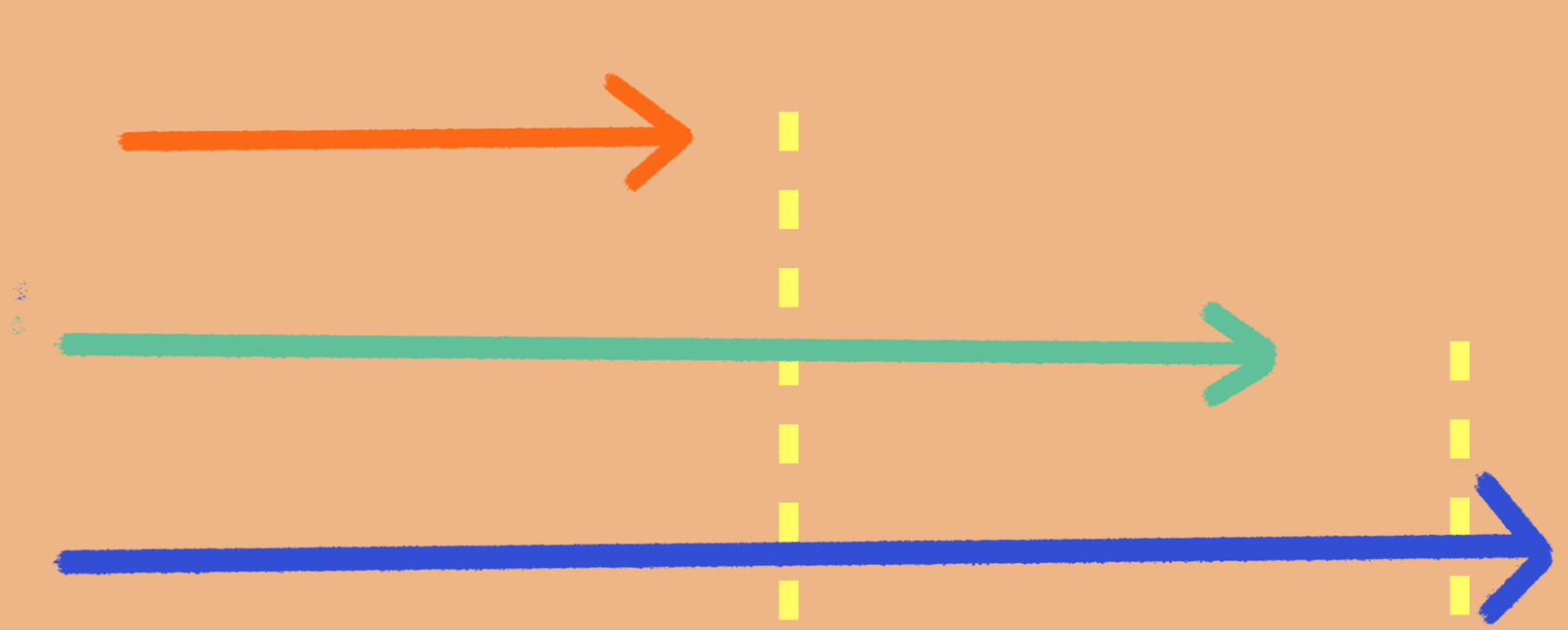
backend



Мокать или не мокать?

mocks

backend



Можем повлиять на зависимость

Main backend API

Own feature toggle

Own addition services

Ходим реально 😎

НЕ можем повлиять на зависимость

(например 3rd-party)

3rd party AB testing tool

Payment processing

Sms phone verification

Мокаем 😭

Чеклист

🎯 Цель, ресурсы, результат

🎲 Для начала сделай как нибудь

…

Чеклист

🎯 Цель, ресурсы, результат

🎲 Для начала сделай как нибудь

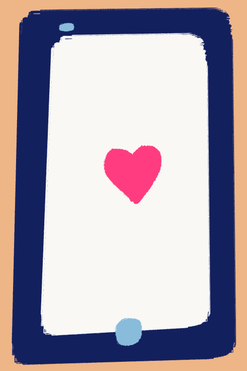
🔄 Свой backend — реально, а 3rd party — mock

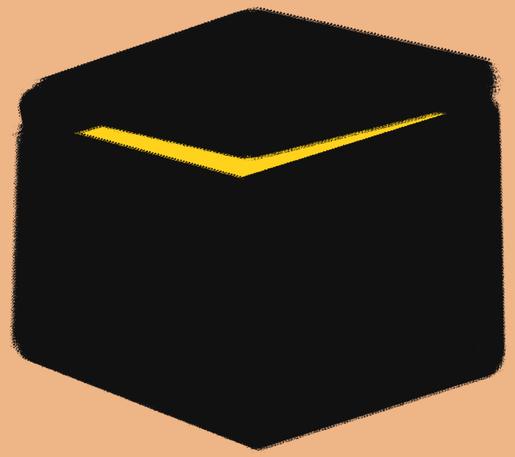
...

Проблемы

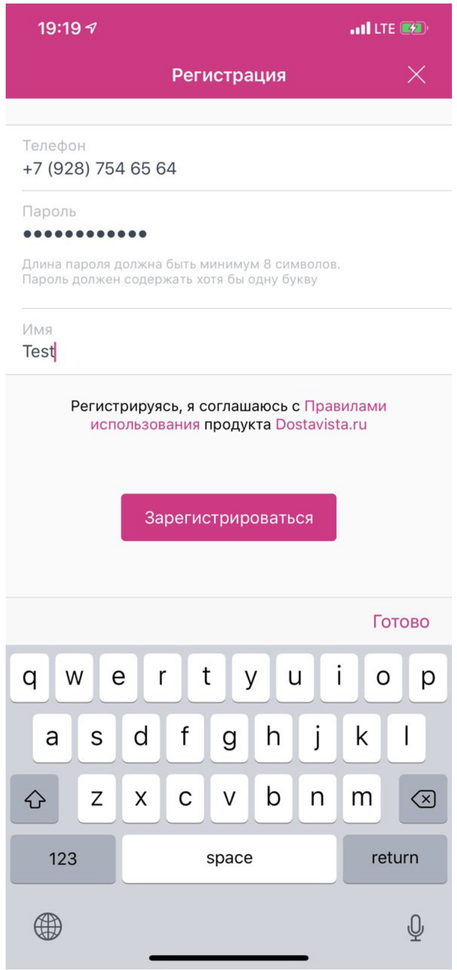
- ~~Mock/stub VS реальный бэкенд?~~
- Подглядывание в black box
- Флаканье
- Автономность (Workstation/CI)
- Воспроизводимость
- Let’s localise it

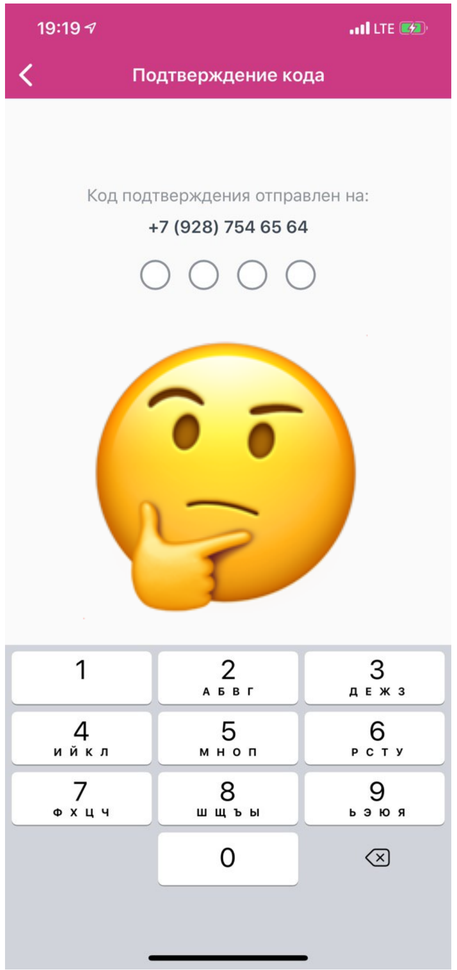
Подглядывание в Bbox

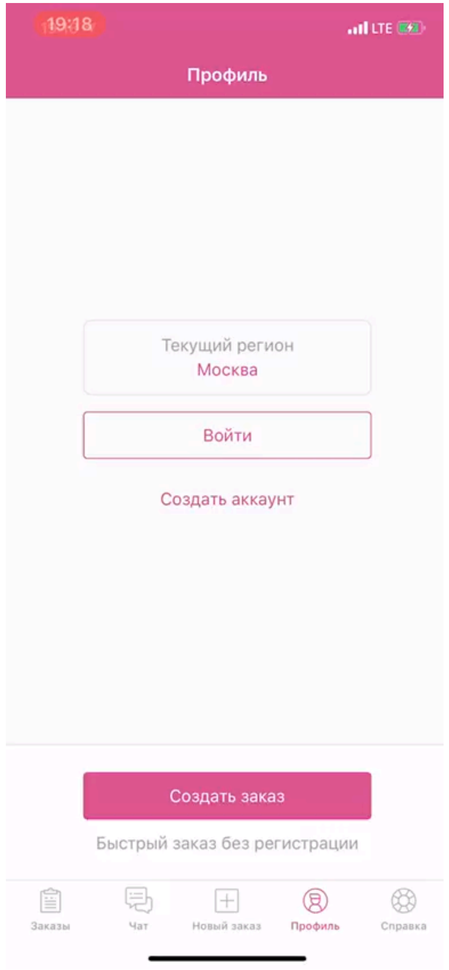




Код смс???

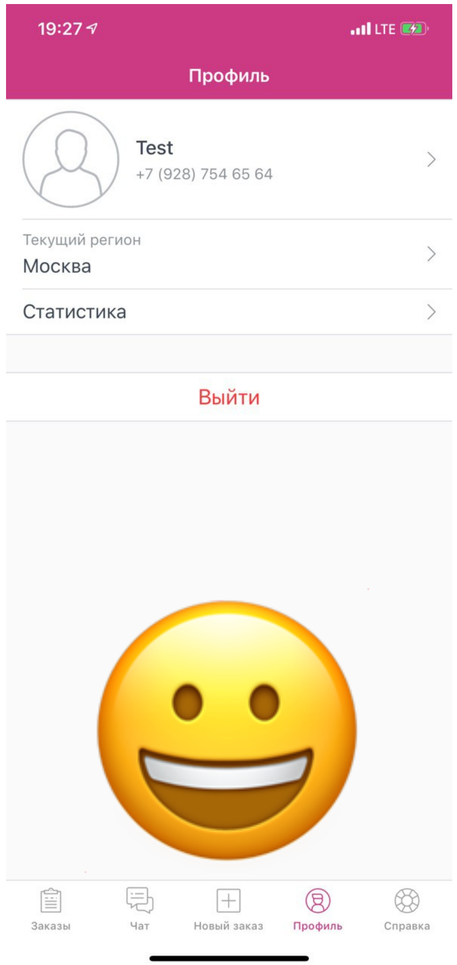




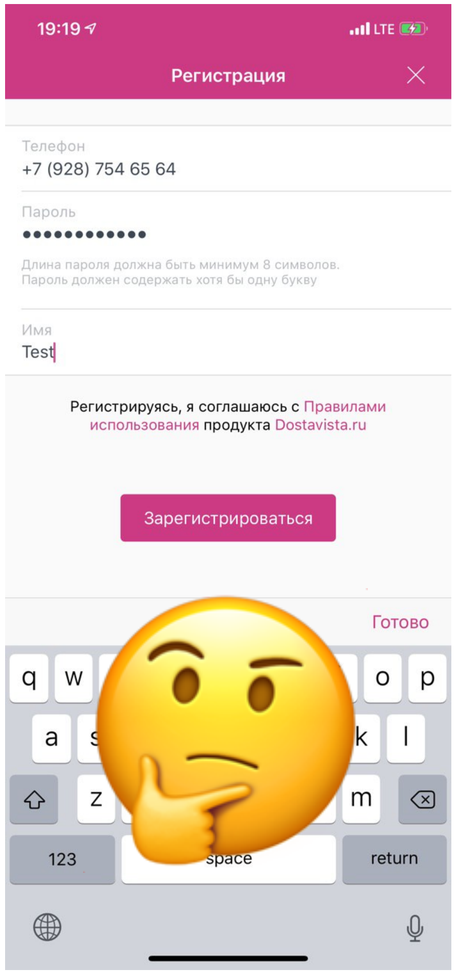


Такой номер уже занят???

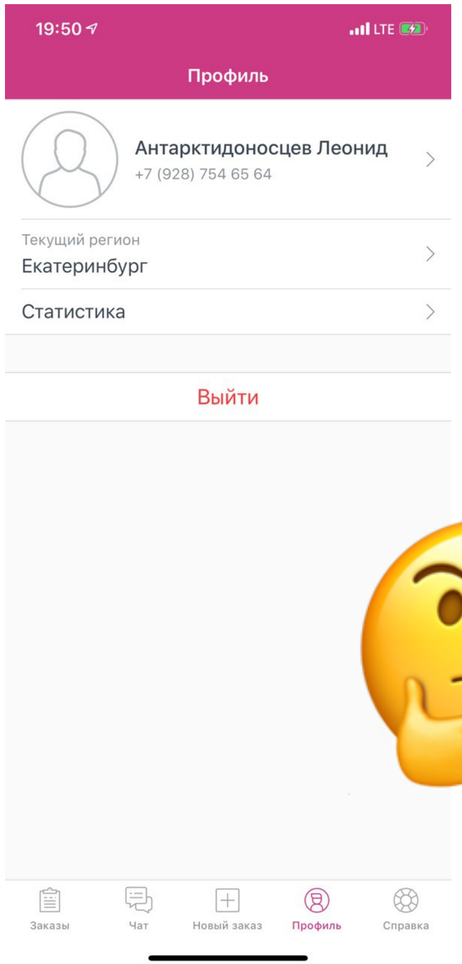
1)

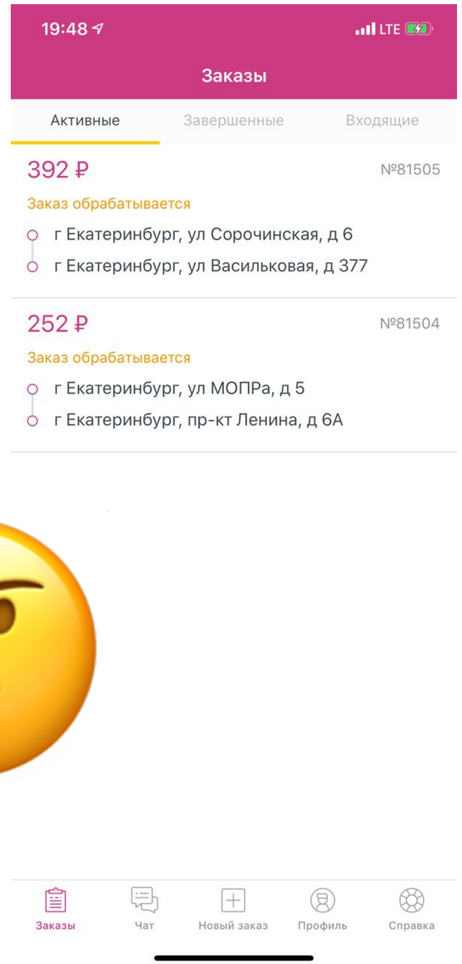


2)

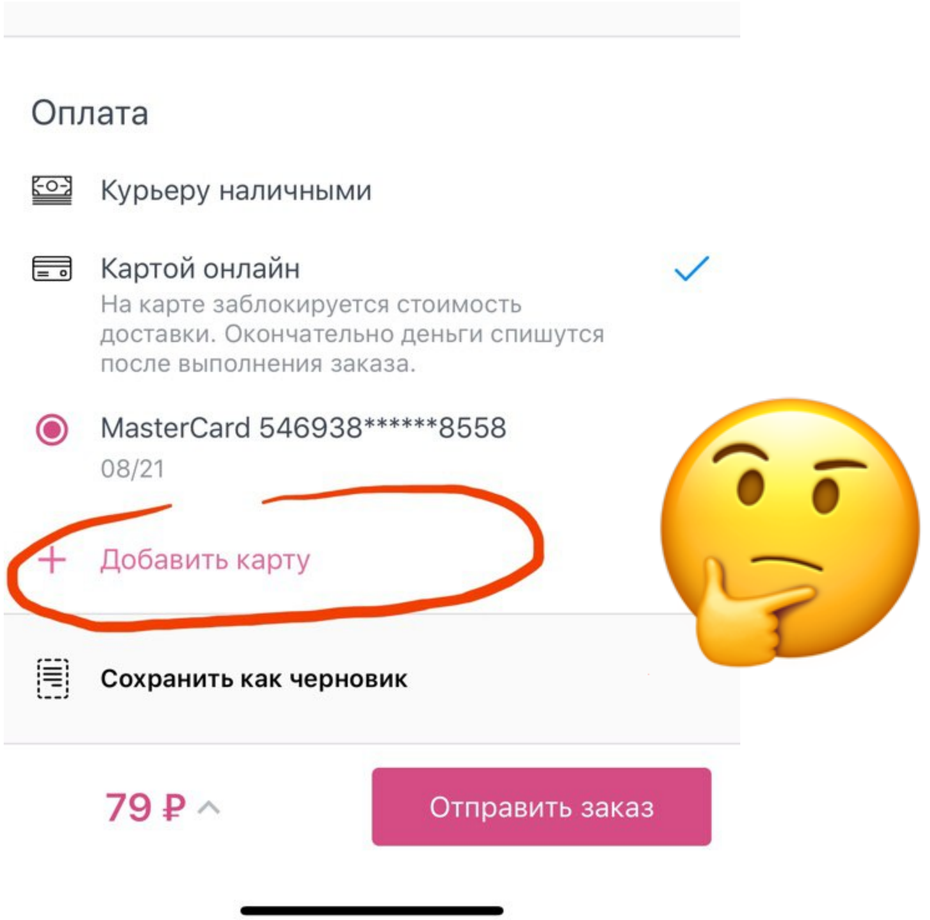


Хочу клиента в Екате с 2 заказами





Привязка банковской карты?



Интеграционное API 😎


```
protocol AutotestHelperApiClient {
  ...

    func getClientPhoneVerificationSmsCode(phone: String)
     -> Result<String, AutotestHelperError>

  ...
}
```

Взаимодействие с командой Backend

Взаимодействие с командой Backend

Чеклист

🎯 Цель, ресурсы, результат

🎲 Для начала сделай как нибудь

🔄 Свой backend — реально, а 3rd party — mock

...

Чеклист

🎯 Цель, ресурсы, результат

🎲 Для начала сделай как нибудь

🔄 Свой backend — реально, а 3rd party — mock

🧰 Интеграционное API

…

Чеклист

🎯 Цель, ресурсы, результат

🎲 Для начала сделай как нибудь

🔄 Свой backend — реально, а 3rd party — mock

🧰 Интеграционное API

👬 Взаимодействие с командой Backend

…

Проблемы

- ~~Mock/stub VS реальный бэкенд?~~
- ~~Подглядывание в black box~~
- Флаканье
- Автономность (Workstation/CI)
- Воспроизводимость
- Let’s localise it

Флаканье, автономность и воспроизводимость

🕰️ Network latency

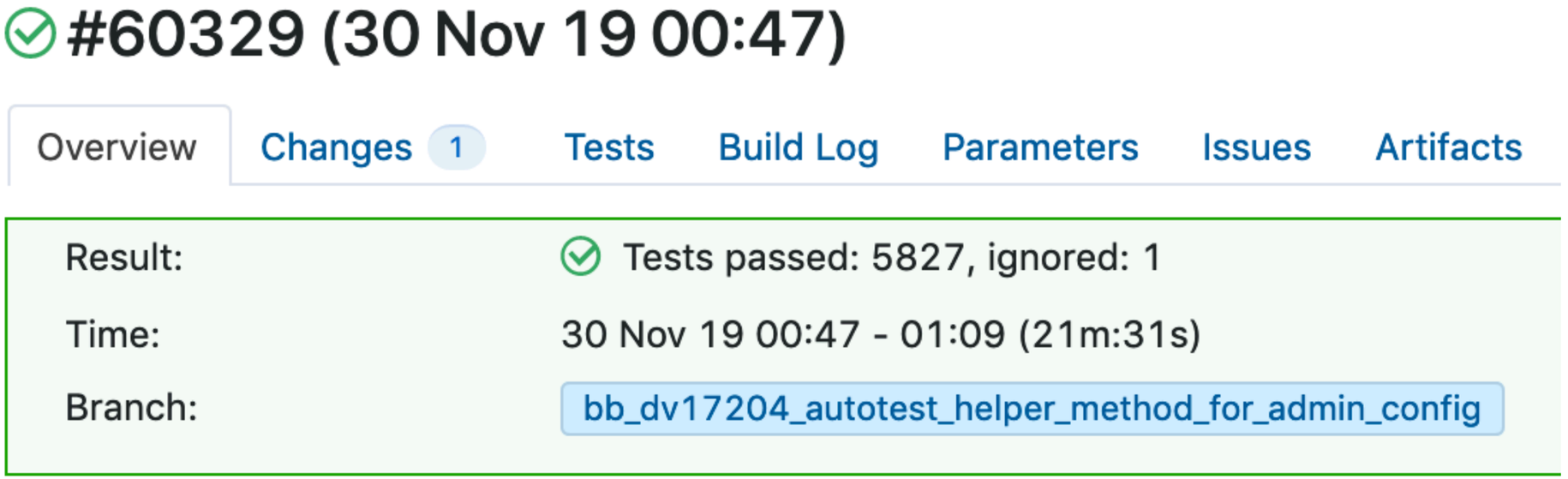
💻 VPN hell при удаленной работе

❌ Кто-то удалил ваши данные со стенда

🚦 Не эксклюзивный доступ к стенду

🎲 Вынужденная рандомизация

Backend, шо у вас по тестам?

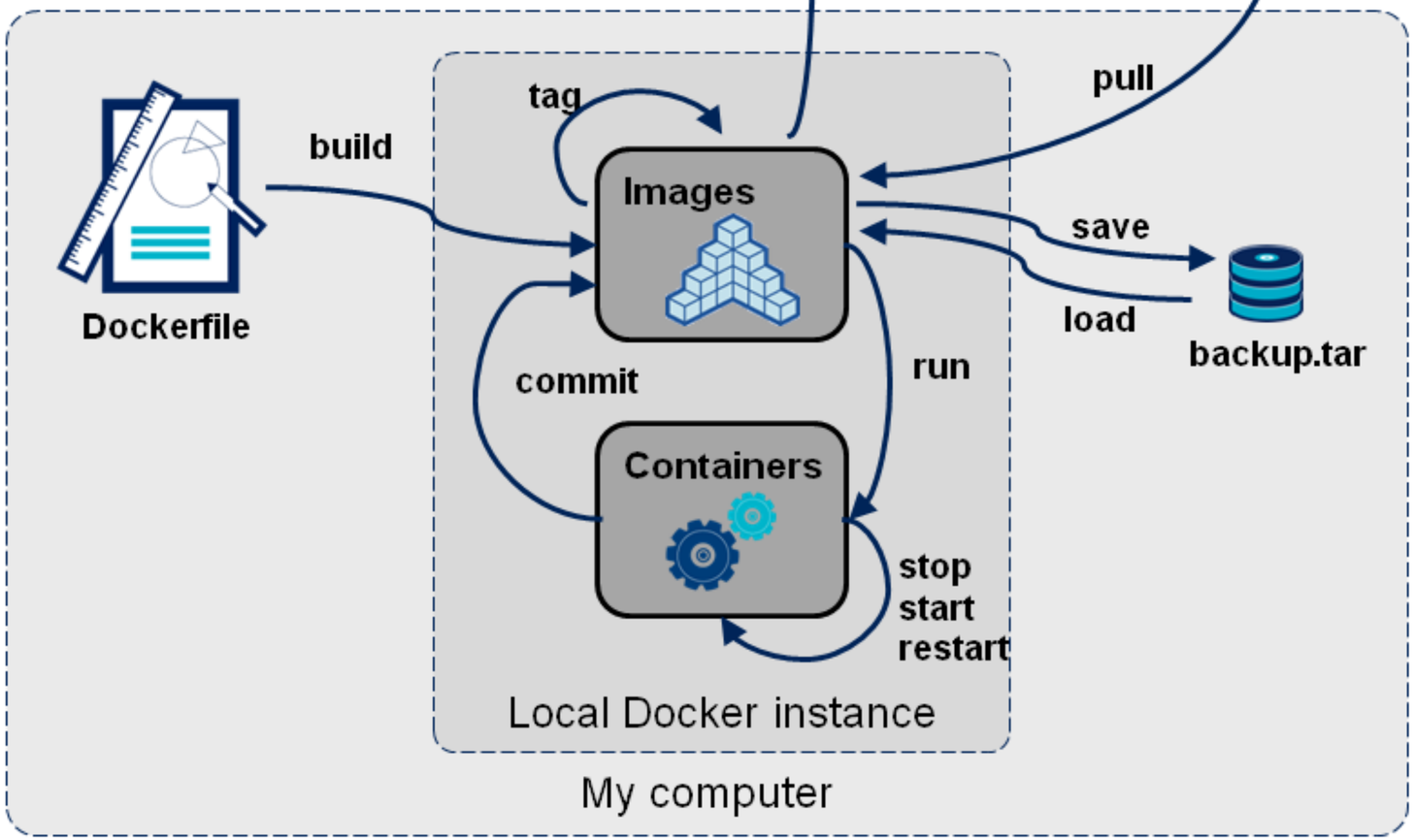


~98% Integration, ~2% Unit

Backend, шо у вас по тестам?

- 🔄 Pull docker image «dv-autotests» and run container
- 🔄 Pull the freshest Backend code
- 🔄 Restore DB from test-dumb.sql
- 🔄 Build code and apply DB Migrations
- 🚀 Run tests

docker



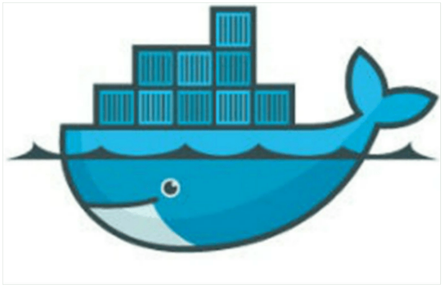
Основные команды docker

```
vasya@VasyaBook pirozhkista (master) $ docker ps
CONTAINER ID    IMAGE                                          COMMAND                  CREATED         STATUS          PORTS                                                  NAMES
f89e2e5024ac    registry.pirozhkista.net/pr-autotests:2.21171  "/usr/sbin/bootstrap…"   28 seconds ago  Up 26 seconds   0.0.0.0:4080->8080/tcp, 0.0.0.0:4088->8081/tcp   pr-autotests-ios



vasya@VasyaBook pirozhkista (master) $ docker images
REPOSITORY                                      TAG                IMAGE ID           CREATED             SIZE
registry.pirozhkista.net/pr-autotests            2.21171            4ef6dbe51d25       5 days ago          4.82GB



vasya@VasyaBook pirozhkista (master) $ docker exec -ti pr-autotests-ios bash
🐋 root@f89e2e5024ac:/#
```

Синтаксис Dockerfile

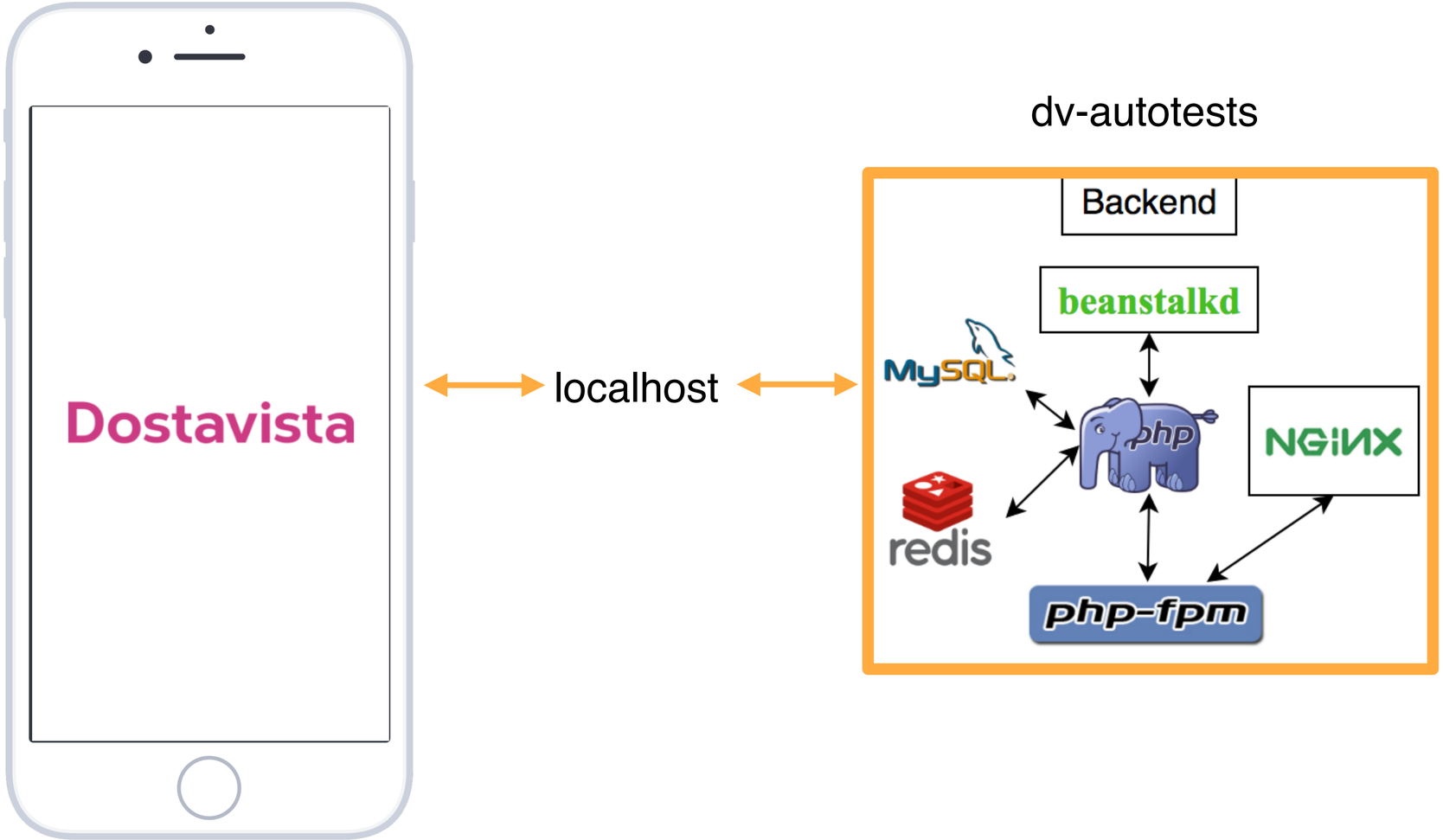
```
MAINTAINER Vasya Pirozhkov <vasya@pirozhkista.ru>
FROM debian:jessie-20191118

RUN apt-get update \
    && apt-get -y install --no-install-recommends \
       openjdk-8-jdk \
    && rm -rf /var/lib/apt/lists/*

COPY target/app.jar /app
CMD ["java", "-jar", "/app/app.jar"]
```

Причем тут вообще docker 🐳?



docker + android iOS

Свой собственный бэкенд 🤩



1) Устанавливаем ENV

DST_ENCRYPTED_CONFIG_VALUES_CIPHER_KEY
DST_GIT_PRIVATE_KEY

2) Запускаем Docker контейнер

```
./build-scripts/autotests.docker/start-container.sh
./build-scripts/autotests.docker/mimic-container-to-country.sh ru
```

3) Запускаем схему тестов

```
killall Simulator 2> /dev/null || true
./build-scripts/install-dependencies.sh
bundle exec fastlane ru_ui_testing
```

1) Устанавливаем ENV

```
DST_ENCRYPTED_CONFIG_VALUES_CIPHER_KEY
DST_GIT_PRIVATE_KEY
```

2) Запускаем Docker контейнер

```
./build-scripts/autotests.docker/start-container.sh
./build-scripts/autotests.docker/mimic-container-to-country.sh ru
```

3) Запускаем схему тестов

```
killall Simulator 2> /dev/null || true
./build-scripts/install-dependencies.sh
bundle exec fastlane ru_ui_testing
```

1) Устанавливаем ENV

```
DST_ENCRYPTED_CONFIG_VALUES_CIPHER_KEY
DST_GIT_PRIVATE_KEY
```

2) Запускаем Docker контейнер

```
./build-scripts/autotests.docker/start-container.sh
./build-scripts/autotests.docker/mimic-container-to-country.sh ru
```

3) Запускаем схему тестов

```
killall Simulator 2> /dev/null || true
./build-scripts/install-dependencies.sh
bundle exec fastlane ru_ui_testing
```

start-container.sh

```
33  # Сохраняем данные из переменной DST_GIT_PRIVATE_KEY в /build-scripts/autotests.docker/id_rsa
34  DST_GIT_PRIVATE_KEY_PATH="${DOCKER_IMAGE_BASE_PATH}/id_rsa"
35  yes | rm -rf "${DST_GIT_PRIVATE_KEY_PATH}" || true
36  echo "$DST_GIT_PRIVATE_KEY" >> "${DST_GIT_PRIVATE_KEY_PATH}"
37  chmod 400 "${DST_GIT_PRIVATE_KEY_PATH}"
38
39  # Стартуем контейнер из образа dv-autotests подсовывая нужные вещи
40  CONTAINER_ID="$(docker run \
41      --detach \
42      --name $CONTAINER_NAME \
43      -v "${DST_GIT_PRIVATE_KEY_PATH}:/root/.ssh/id_rsa:ro" \
44      -v "${DOCKER_IMAGE_BASE_PATH}/application.php:/var/www/XXX/application.php:ro" \
45      -v "${DOCKER_IMAGE_BASE_PATH}/custom-configs:/var/www/XXX/custom-configs:ro" \
46      -v "${DOCKER_IMAGE_BASE_PATH}/local-config.php:/var/www/XXX/frontend/local-config.php:ro" \
47      -e "DST_ENCRYPTED_CONFIG_VALUES_CIPHER_KEY=${DST_ENCRYPTED_CONFIG_VALUES_CIPHER_KEY}" \
48      -p 4080:8080 \
49      -p 4088:8081 \
50      "$IMAGE_NAME")"
```

start-container.sh

```
33  # Сохраняем данные из переменной DST_GIT_PRIVATE_KEY в /build-scripts/autotests.docker/id_rsa
34  DST_GIT_PRIVATE_KEY_PATH="${DOCKER_IMAGE_BASE_PATH}/id_rsa"
35  yes | rm -rf "${DST_GIT_PRIVATE_KEY_PATH}" || true
36  echo "$DST_GIT_PRIVATE_KEY" >> "${DST_GIT_PRIVATE_KEY_PATH}"
37  chmod 400 "${DST_GIT_PRIVATE_KEY_PATH}"
38
39  # Стартуем контейнер из образа dv-autotests подсовывая нужные вещи
40  CONTAINER_ID="$(docker run \
41      --detach \
42      --name $CONTAINER_NAME \
43      -v "${DST_GIT_PRIVATE_KEY_PATH}:/root/.ssh/id_rsa:ro" \
44      -v "${DOCKER_IMAGE_BASE_PATH}/application.php:/var/www/XXX/application.php:ro" \
45      -v "${DOCKER_IMAGE_BASE_PATH}/custom-configs:/var/www/XXX/custom-configs:ro" \
46      -v "${DOCKER_IMAGE_BASE_PATH}/local-config.php:/var/www/XXX/frontend/local-config.php:ro" \
47      -e "DST_ENCRYPTED_CONFIG_VALUES_CIPHER_KEY=${DST_ENCRYPTED_CONFIG_VALUES_CIPHER_KEY}" \
48      -p 4080:8080 \
49      -p 4088:8081 \
50      "$IMAGE_NAME")"
```

start-container.sh

```
33  # Сохраняем данные из переменной DST_GIT_PRIVATE_KEY в /build-scripts/autotests.docker/id_rsa
34  DST_GIT_PRIVATE_KEY_PATH="${DOCKER_IMAGE_BASE_PATH}/id_rsa"
35  yes | rm -rf "${DST_GIT_PRIVATE_KEY_PATH}" || true
36  echo "$DST_GIT_PRIVATE_KEY" >> "${DST_GIT_PRIVATE_KEY_PATH}"
37  chmod 400 "${DST_GIT_PRIVATE_KEY_PATH}"
38
39  # Стартуем контейнер из образа dv-autotests подсовывая нужные вещи
40  CONTAINER_ID="$(docker run \
41      --detach \
42      --name $CONTAINER_NAME \
43      -v "${DST_GIT_PRIVATE_KEY_PATH}:/root/.ssh/id_rsa:ro" \
44      -v "${DOCKER_IMAGE_BASE_PATH}/application.php:/var/www/XXX/application.php:ro" \
45      -v "${DOCKER_IMAGE_BASE_PATH}/custom-configs:/var/www/XXX/custom-configs:ro" \
46      -v "${DOCKER_IMAGE_BASE_PATH}/local-config.php:/var/www/XXX/frontend/local-config.php:ro" \
47      -e "DST_ENCRYPTED_CONFIG_VALUES_CIPHER_KEY=${DST_ENCRYPTED_CONFIG_VALUES_CIPHER_KEY}" \
48      -p 4080:8080 \
49      -p 4088:8081 \
50      "$IMAGE_NAME")"
```

start-container.sh

```
33  # Сохраняем данные из переменной DST_GIT_PRIVATE_KEY в /build-scripts/autotests.docker/id_rsa
34  DST_GIT_PRIVATE_KEY_PATH="${DOCKER_IMAGE_BASE_PATH}/id_rsa"
35  yes | rm -rf "${DST_GIT_PRIVATE_KEY_PATH}" || true
36  echo "$DST_GIT_PRIVATE_KEY" >> "${DST_GIT_PRIVATE_KEY_PATH}"
37  chmod 400 "${DST_GIT_PRIVATE_KEY_PATH}"
38
39  # Стартуем контейнер из образа dv-autotests подсовывая нужные вещи
40  CONTAINER_ID="$(docker run \
41      --detach \
42      --name $CONTAINER_NAME \
43      -v "${DST_GIT_PRIVATE_KEY_PATH}:/root/.ssh/id_rsa:ro" \
44      -v "${DOCKER_IMAGE_BASE_PATH}/application.php:/var/www/XXX/application.php:ro" \
45      -v "${DOCKER_IMAGE_BASE_PATH}/custom-configs:/var/www/XXX/custom-configs:ro" \
46      -v "${DOCKER_IMAGE_BASE_PATH}/local-config.php:/var/www/XXX/frontend/local-config.php:ro" \
47      -e "DST_ENCRYPTED_CONFIG_VALUES_CIPHER_KEY=${DST_ENCRYPTED_CONFIG_VALUES_CIPHER_KEY}" \
48      -p 4080:8080 \
49      -p 4088:8081 \
50      "$IMAGE_NAME")"
```

start-container.sh

```
33  # Сохраняем данные из переменной DST_GIT_PRIVATE_KEY в /build-scripts/autotests.docker/id_rsa
34  DST_GIT_PRIVATE_KEY_PATH="${DOCKER_IMAGE_BASE_PATH}/id_rsa"
35  yes | rm -rf "${DST_GIT_PRIVATE_KEY_PATH}" || true
36  echo "$DST_GIT_PRIVATE_KEY" >> "${DST_GIT_PRIVATE_KEY_PATH}"
37  chmod 400 "${DST_GIT_PRIVATE_KEY_PATH}"
38
39  # Стартуем контейнер из образа dv-autotests подсовывая нужные вещи
40  CONTAINER_ID="$(docker run \
41      --detach \
42      --name $CONTAINER_NAME \
43      -v "${DST_GIT_PRIVATE_KEY_PATH}:/root/.ssh/id_rsa:ro" \
44      -v "${DOCKER_IMAGE_BASE_PATH}/application.php:/var/www/XXX/application.php:ro" \
45      -v "${DOCKER_IMAGE_BASE_PATH}/custom-configs:/var/www/XXX/custom-configs:ro" \
46      -v "${DOCKER_IMAGE_BASE_PATH}/local-config.php:/var/www/XXX/frontend/local-config.php:ro" \
47      -e "DST_ENCRYPTED_CONFIG_VALUES_CIPHER_KEY=${DST_ENCRYPTED_CONFIG_VALUES_CIPHER_KEY}" \
48      -p 4080:8080 \
49      -p 4088:8081 \
50      "$IMAGE_NAME")"
```

start-container.sh

```
33  # Сохраняем данные из переменной DST_GIT_PRIVATE_KEY в /build-scripts/autotests.docker/id_rsa
34  DST_GIT_PRIVATE_KEY_PATH="${DOCKER_IMAGE_BASE_PATH}/id_rsa"
35  yes | rm -rf "${DST_GIT_PRIVATE_KEY_PATH}" || true
36  echo "$DST_GIT_PRIVATE_KEY" >> "${DST_GIT_PRIVATE_KEY_PATH}"
37  chmod 400 "${DST_GIT_PRIVATE_KEY_PATH}"
38
39  # Стартуем контейнер из образа dv-autotests подсовывая нужные вещи
40  CONTAINER_ID="$(docker run \
41      --detach \
42      --name $CONTAINER_NAME \
43      -v "${DST_GIT_PRIVATE_KEY_PATH}:/root/.ssh/id_rsa:ro" \
44      -v "${DOCKER_IMAGE_BASE_PATH}/application.php:/var/www/XXX/application.php:ro" \
45      -v "${DOCKER_IMAGE_BASE_PATH}/custom-configs:/var/www/XXX/custom-configs:ro" \
46      -v "${DOCKER_IMAGE_BASE_PATH}/local-config.php:/var/www/XXX/frontend/local-config.php:ro" \
47      -e "DST_ENCRYPTED_CONFIG_VALUES_CIPHER_KEY=${DST_ENCRYPTED_CONFIG_VALUES_CIPHER_KEY}" \
48      -p 4080:8080 \
49      -p 4088:8081 \
50      "$IMAGE_NAME")"
```

Задаем ENV на CI

Environment variables

Environment variables are applied to environments via the runner. They can be protected by only exposing them to protected branches or tags. You can use environment variables for passwords, secret keys, or whatever you want. You may also add variables that are made available to the running application by prepending the variable key with `K8S_SECRET_`. More information

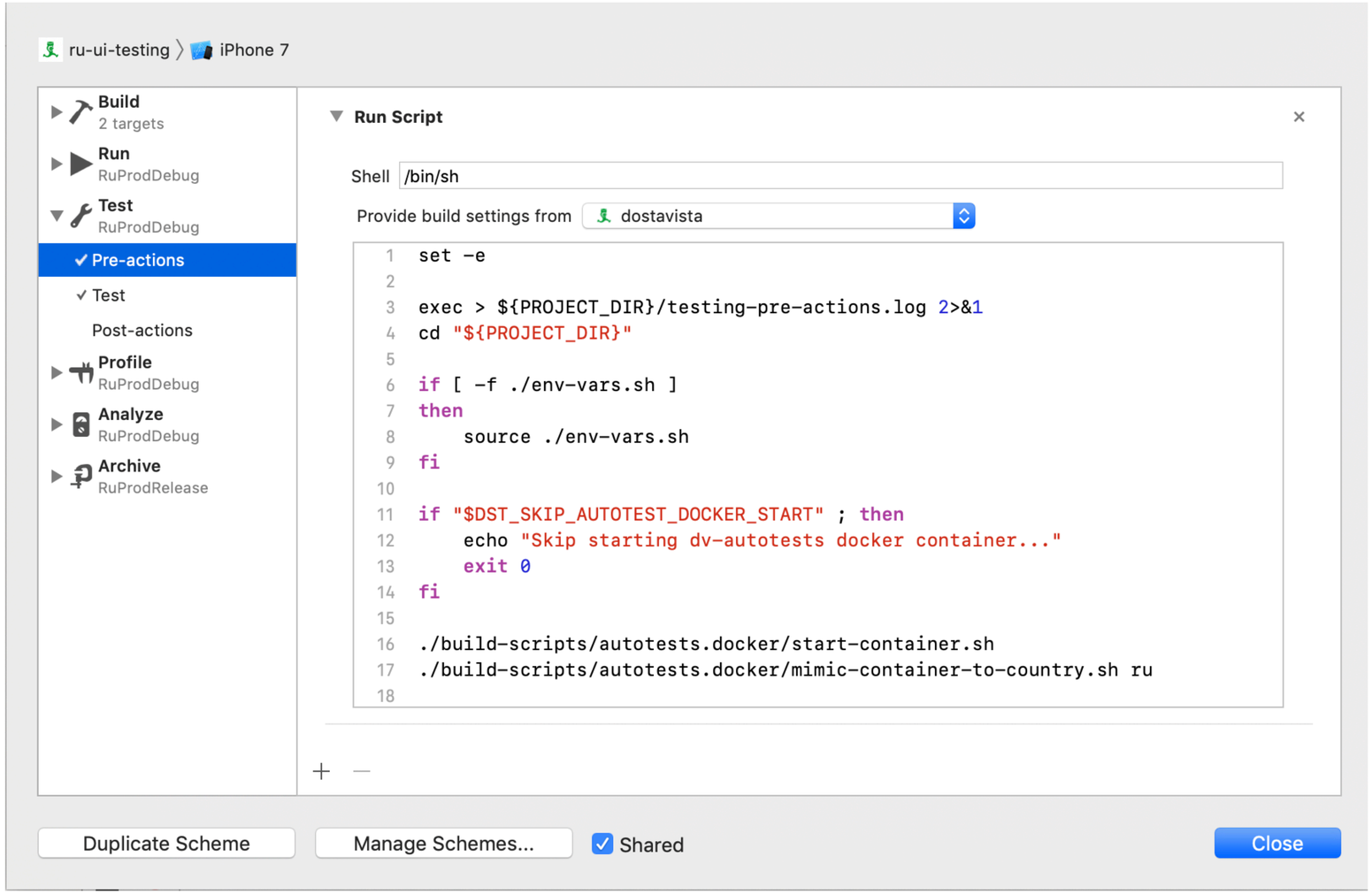
DST_ENCRYPTED_CONFIG_VALUES_CIPHER_KEY	**********	Protected
DST_GIT_PRIVATE_KEY	**********	Protected

Задаем ENV в Xcode

env-vars.sh 〉 No Selection

```
1  #!/bin/bash
2
3  export DST_ENCRYPTED_CONFIG_VALUES_CIPHER_KEY="XXX"
4  export DST_GIT_PRIVATE_KEY=`cat $HOME/.ssh/id_rsa_docker_autotests`
5  # Раскомментировать следующую строку если dv-autotests
6  # docker контейнером управляешь самостоятельно
7  # export DST_SKIP_AUTOTEST_DOCKER_START=true
8
```

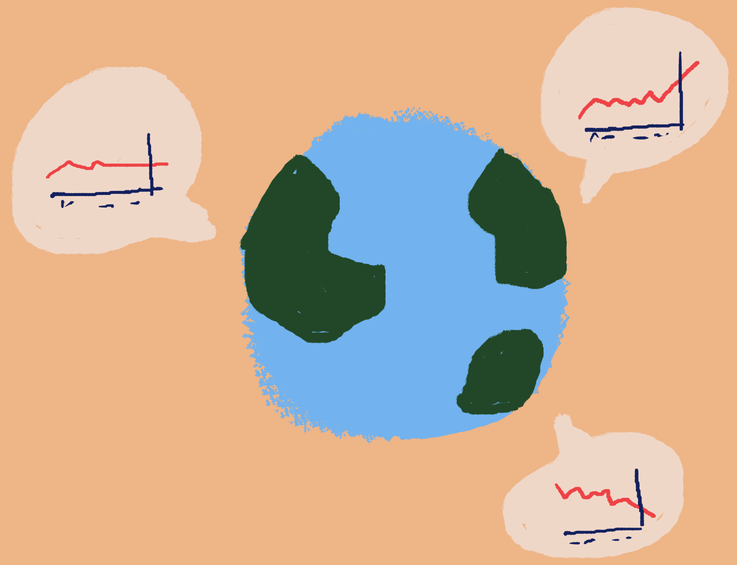
Pre-actions, логи и код возврата 😭



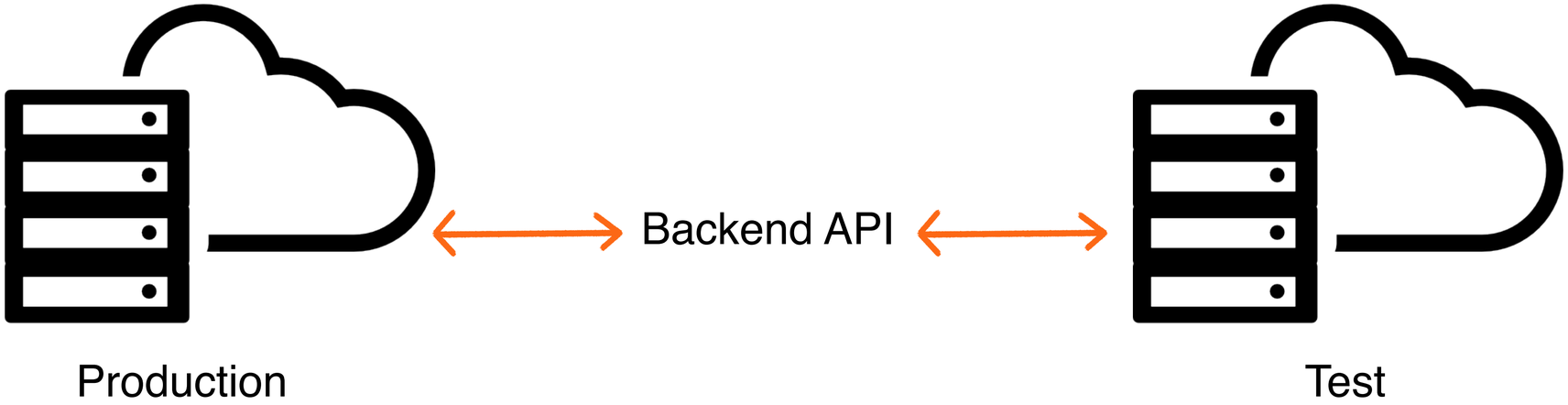
Взаимодействие с командой DevOps

Взаимодействие с командой DevOps

История про синхронизацию справочников



Environment data sync



Чеклист

🎯 Цель, ресурсы, результат

🎲 Для начала сделай как нибудь

🔄 Свой backend — реально, а 3rd party — mock

🗃️ Интеграционное API

👬 Взаимодействие с командой Backend

…

Чеклист

🎯 Цель, ресурсы, результат

🎲 Для начала сделай как нибудь

🔄 Свой backend — реально, а 3rd party — mock

🗃️ Интеграционное API

👬 Взаимодействие с командой Backend

🐋 Изоляция окружения с Docker

…

Чеклист

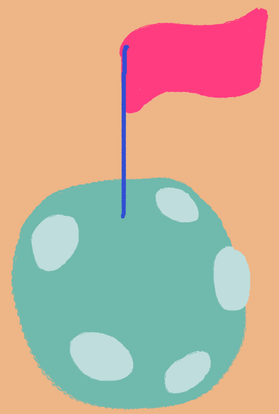
- 🎯 Цель, ресурсы, результат
- 🎲 Для начала сделай как нибудь
- 🔄 Свой backend — реально, а 3rd party — mock
- 🗃️ Интеграционное API
- 👬 Взаимодействие с командой Backend
- 🐋 Изоляция окружения с Docker
- 👬 Взаимодействие с командой DevOps

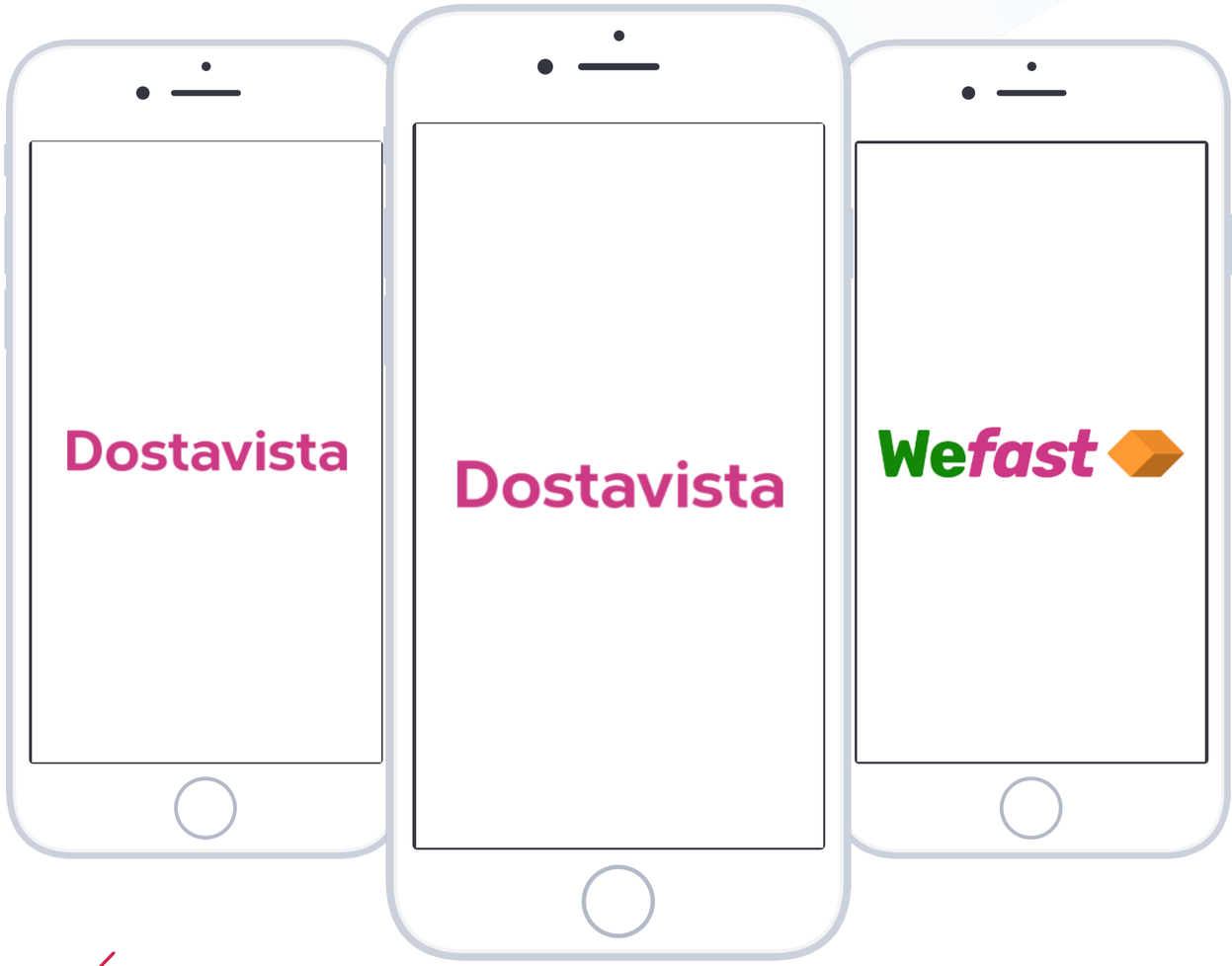
…

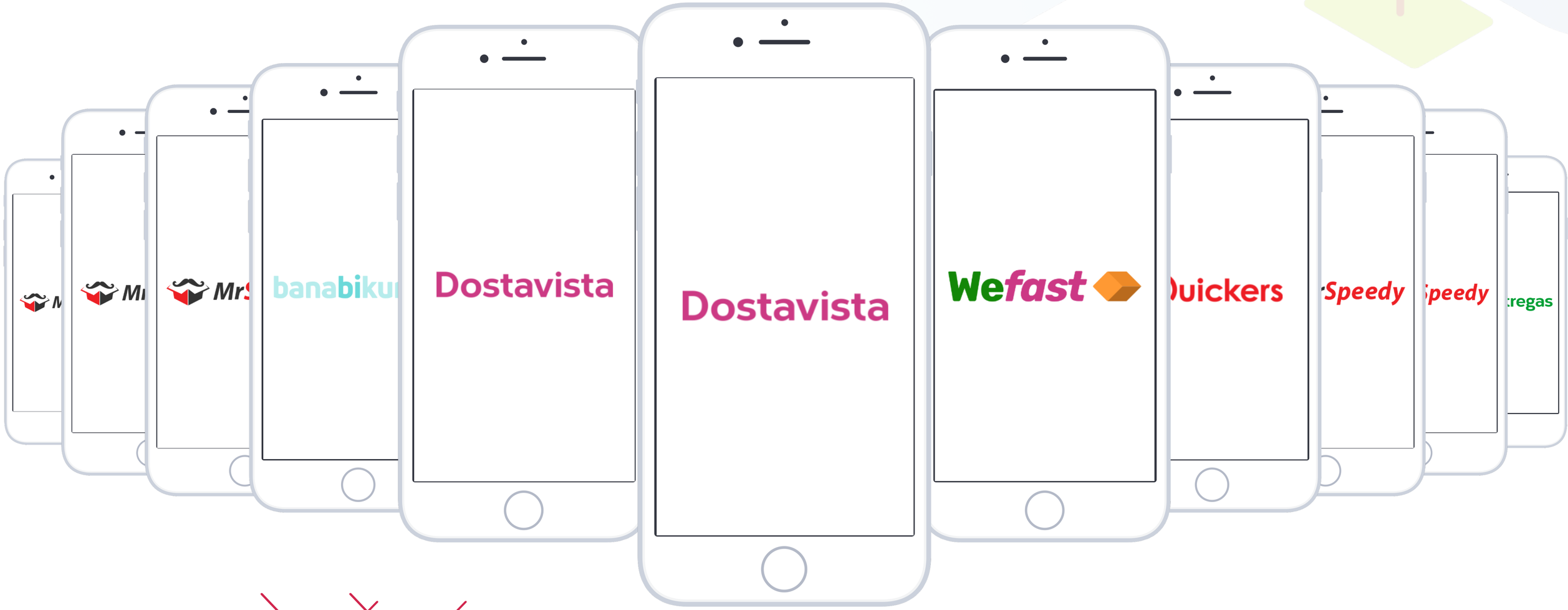
Проблемы

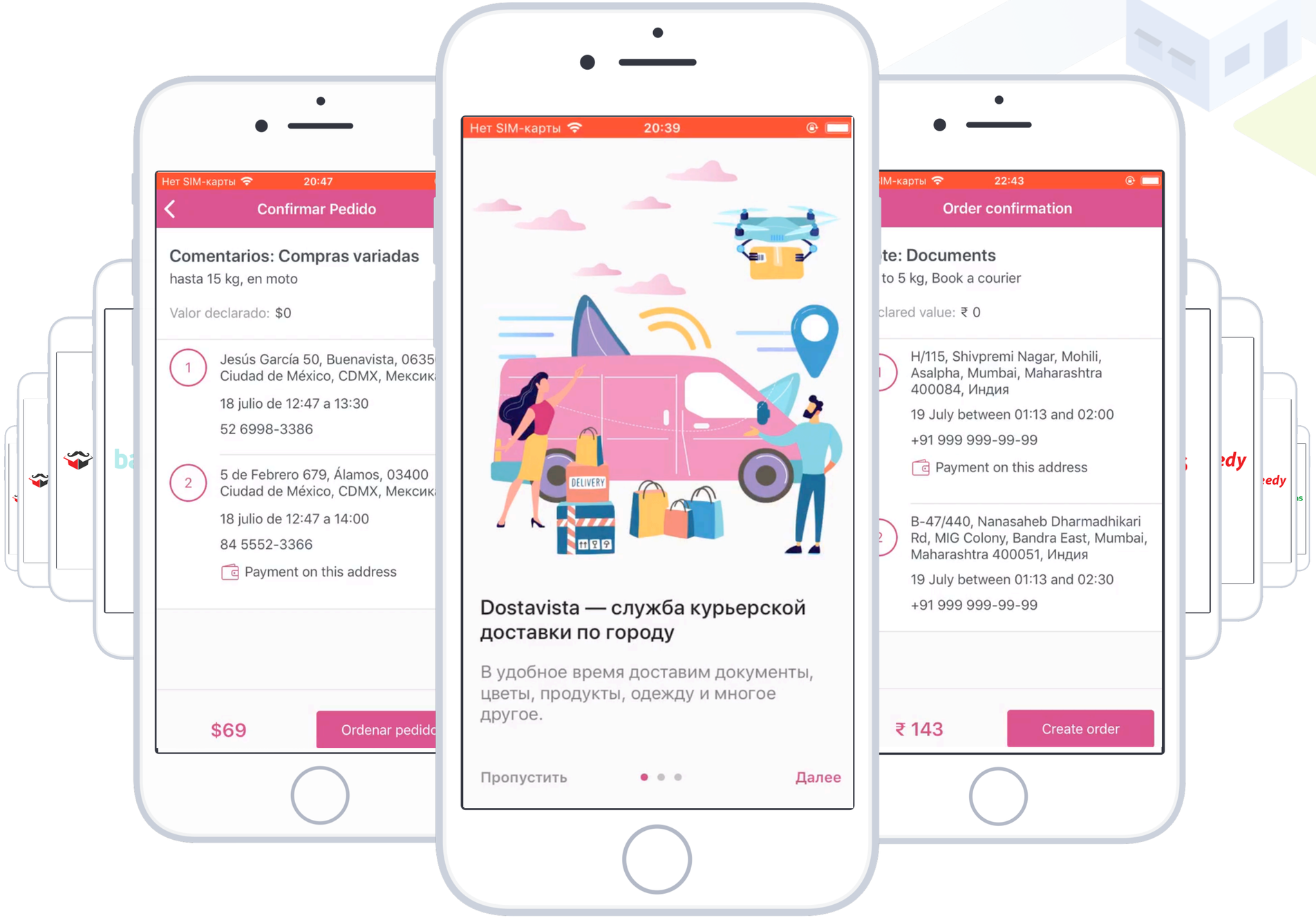
- ~~Mock/stub VS реальный бэкенд?~~
- ~~Подглядывание в black box~~
- ~~Флаканье~~
- ~~Автономность (Workstation/CI)~~
- ~~Воспроизводимость~~
- Let’s localise it

История про страны

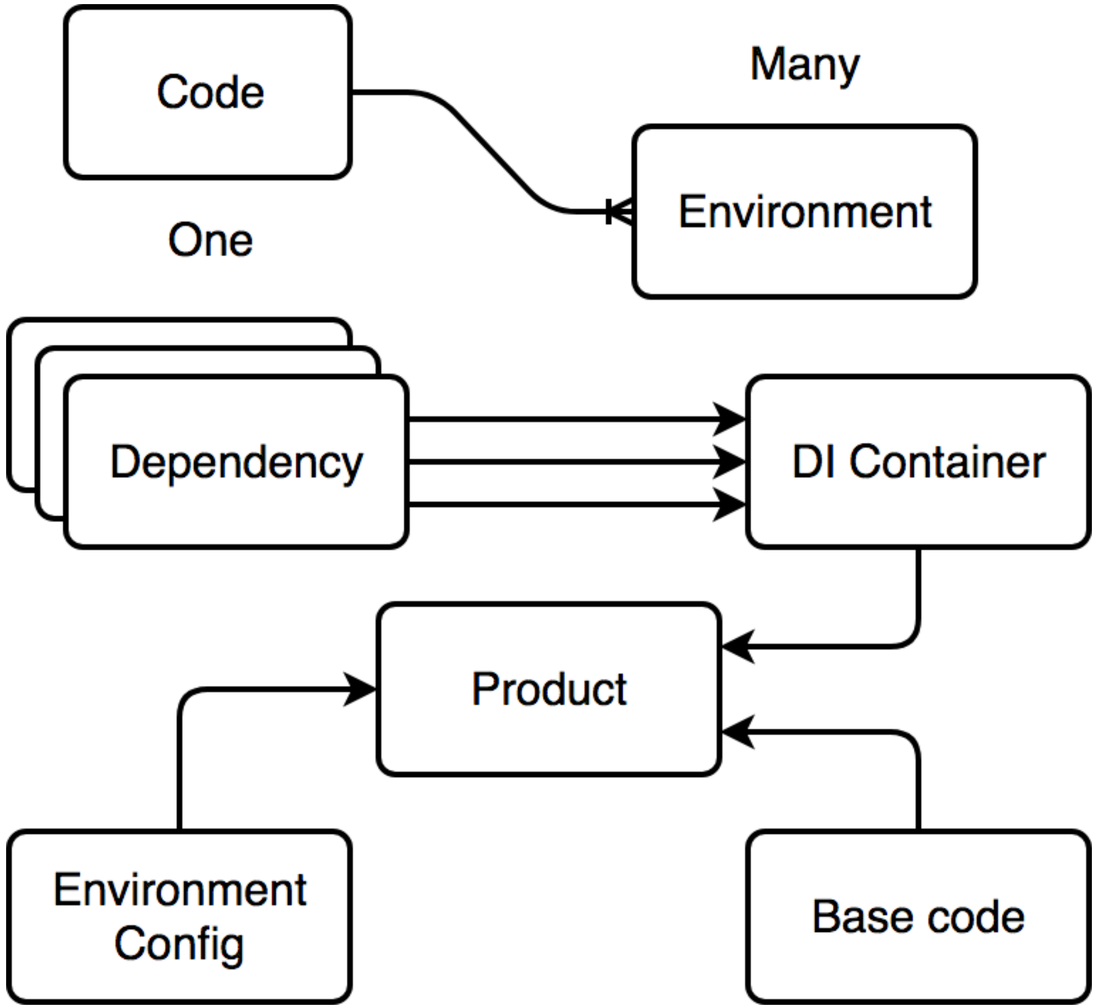








Новая страна😱? Новое окружение😉!



mimic-container-to-country.sh ru

```
23 # Генерируем sed-ом нужный applcation.php монтируемый внутрь контейнера dv-autotests
24 if [ -z $DST_ENCRYPTED_CONFIG_VALUES_CIPHER_KEY ];
25 then
26     echo "DST_ENCRYPTED_CONFIG_VALUES_CIPHER_KEY is required for decrypr encrypted config values"
27     exit 1
28 fi
29 DOCKER_IMAGE_BASE_PATH="`pwd`/build-scripts/autotests.docker"
30 yes | cp -rf "${DOCKER_IMAGE_BASE_PATH}/application.php.template" "${DOCKER_IMAGE_BASE_PATH}/application.php"
31 yes | cp -rf "${DOCKER_IMAGE_BASE_PATH}/local-config.php.template" "${DOCKER_IMAGE_BASE_PATH}/local-config.php"
32 sed -i '' "s/###country_name###/${DST_COUNTRY_NAME}/" "${DOCKER_IMAGE_BASE_PATH}/application.php"
33 sed -i '' "s/###country_name###/${DST_COUNTRY_NAME}/" "${DOCKER_IMAGE_BASE_PATH}/local-config.php"
34 sed -i '' "s/###encrypted_config_values_cipher_key###/${DST_ENCRYPTED_CONFIG_VALUES_CIPHER_KEY}/" \
35           "${DOCKER_IMAGE_BASE_PATH}/application.php"
```

mimic-container-to-country.sh ru

```
23 # Генерируем sed-ом нужный applcation.php монтируемый внутрь контейнера dv-autotests
24 if [ -z $DST_ENCRYPTED_CONFIG_VALUES_CIPHER_KEY ];
25 then
26     echo "DST_ENCRYPTED_CONFIG_VALUES_CIPHER_KEY is required for decrypr encrypted config values"
27     exit 1
28 fi
29 DOCKER_IMAGE_BASE_PATH="`pwd`/build-scripts/autotests.docker"
30 yes | cp -rf "${DOCKER_IMAGE_BASE_PATH}/application.php.template" "${DOCKER_IMAGE_BASE_PATH}/application.php"
31 yes | cp -rf "${DOCKER_IMAGE_BASE_PATH}/local-config.php.template" "${DOCKER_IMAGE_BASE_PATH}/local-config.php"
32 sed -i '' "s/###country_name###/${DST_COUNTRY_NAME}/" "${DOCKER_IMAGE_BASE_PATH}/application.php"
33 sed -i '' "s/###country_name###/${DST_COUNTRY_NAME}/" "${DOCKER_IMAGE_BASE_PATH}/local-config.php"
34 sed -i '' "s/###encrypted_config_values_cipher_key###/${DST_ENCRYPTED_CONFIG_VALUES_CIPHER_KEY}/" \
35           "${DOCKER_IMAGE_BASE_PATH}/application.php"
```

mimic-container-to-country.sh ru

```
23  # Генерируем sed-ом нужный applcation.php монтируемый внутрь контейнера dv-autotests
24  if [ -z $DST_ENCRYPTED_CONFIG_VALUES_CIPHER_KEY ];
25  then
26      echo "DST_ENCRYPTED_CONFIG_VALUES_CIPHER_KEY is required for decrypr encrypted config values"
27      exit 1
28  fi
29  DOCKER_IMAGE_BASE_PATH="`pwd`/build-scripts/autotests.docker"
30  yes | cp -rf "${DOCKER_IMAGE_BASE_PATH}/application.php.template" "${DOCKER_IMAGE_BASE_PATH}/application.php"
31  yes | cp -rf "${DOCKER_IMAGE_BASE_PATH}/local-config.php.template" "${DOCKER_IMAGE_BASE_PATH}/local-config.php"
32  sed -i '' "s/###country_name###/${DST_COUNTRY_NAME}/" "${DOCKER_IMAGE_BASE_PATH}/application.php"
33  sed -i '' "s/###country_name###/${DST_COUNTRY_NAME}/" "${DOCKER_IMAGE_BASE_PATH}/local-config.php"
34  sed -i '' "s/###encrypted_config_values_cipher_key###/${DST_ENCRYPTED_CONFIG_VALUES_CIPHER_KEY}/" \
35            "${DOCKER_IMAGE_BASE_PATH}/application.php"
```

mimic-container-to-country.sh ru

```
23  # Генерируем sed-ом нужный applcation.php монтируемый внутрь контейнера dv-autotests
24  if [ -z $DST_ENCRYPTED_CONFIG_VALUES_CIPHER_KEY ];
25  then
26      echo "DST_ENCRYPTED_CONFIG_VALUES_CIPHER_KEY is required for decrypr encrypted config values"
27      exit 1
28  fi
29  DOCKER_IMAGE_BASE_PATH="`pwd`/build-scripts/autotests.docker"
30  yes | cp -rf "${DOCKER_IMAGE_BASE_PATH}/application.php.template" "${DOCKER_IMAGE_BASE_PATH}/application.php"
31  yes | cp -rf "${DOCKER_IMAGE_BASE_PATH}/local-config.php.template" "${DOCKER_IMAGE_BASE_PATH}/local-config.php"
32  sed -i '' "s/###country_name###/${DST_COUNTRY_NAME}/" "${DOCKER_IMAGE_BASE_PATH}/application.php"
33  sed -i '' "s/###country_name###/${DST_COUNTRY_NAME}/" "${DOCKER_IMAGE_BASE_PATH}/local-config.php"
34  sed -i '' "s/###encrypted_config_values_cipher_key###/${DST_ENCRYPTED_CONFIG_VALUES_CIPHER_KEY}/" \
35            "${DOCKER_IMAGE_BASE_PATH}/application.php"
```

Интеграционное API 😎


```
protocol AutotestHelperApiClient {
  ...

    func setAdminConfigParameterValue(parameterName: AdminConfigParameter,
                                      parameterValue: String)
     -> Result<Void, AutotestHelperError>

  ...
}
```

Чеклист

- 🎯 Цель, ресурсы, результат
- 🎲 Для начала сделай как нибудь
- 🔄 Свой backend — реально, а 3rd party — mock
- 🗃️ Интеграционное API
- 👬 Взаимодействие с командой Backend
- 🐋 Изоляция окружения с Docker
- 👬 Взаимодействие с командой DevOps

…

Чеклист

- 🎯 Цель, ресурсы, результат
- 🎲 Для начала сделай как нибудь
- 🔄 Свой backend — реально, а 3rd party — mock
- 🗃️ Интеграционное API
- 👬 Взаимодействие с командой Backend
- 🐋 Изоляция окружения с Docker
- 👬 Взаимодействие с командой DevOps
- ✈️ On-the-fly тюнинг окружения (локализация)

3) Launch RU App

4) Set RU test data usage

5) Input test data

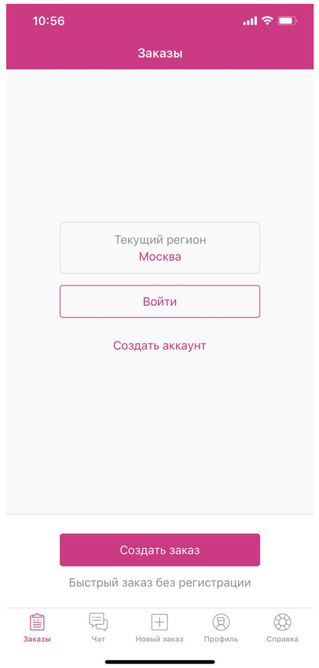
6) Perform registration

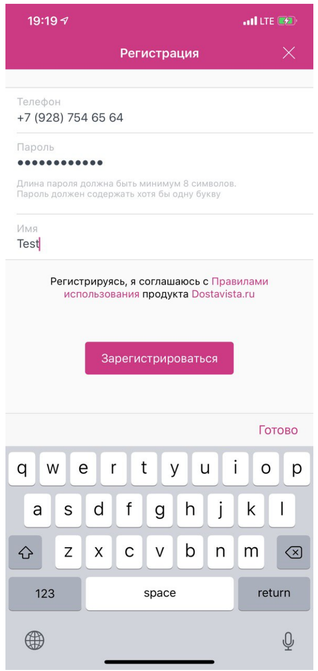
7) request sms code from autotest-helper API

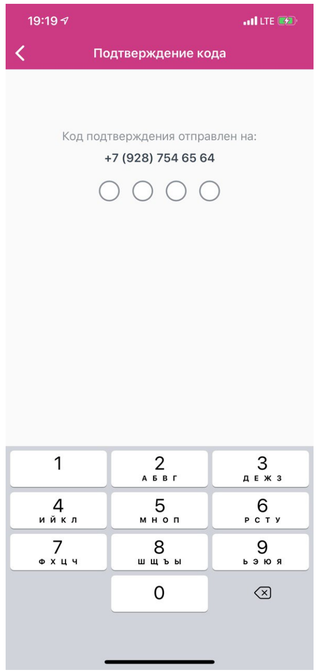
8) Perform registration

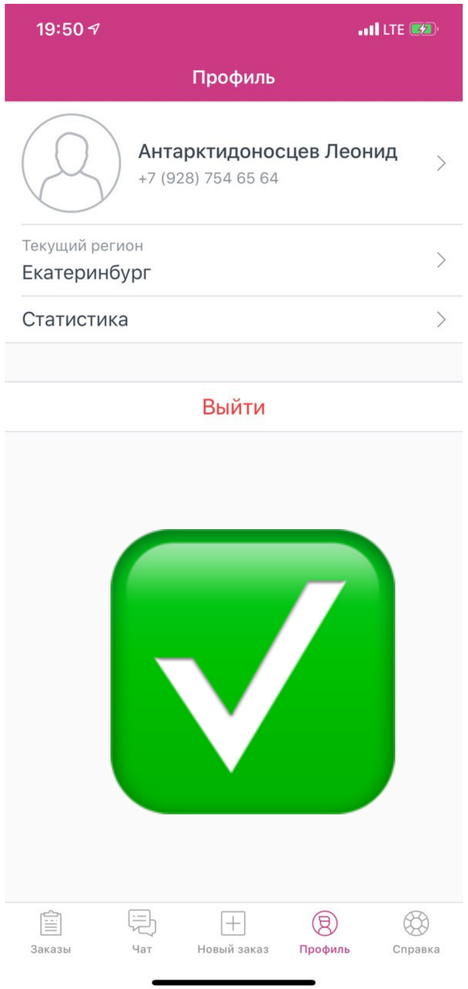
9) Ensure user register

10) Congrat 🎈🍾









1) ./start-container.sh

2) ./mimic-container.sh RU

http://localhost:4041

dv-autotests docker container

RUN TEST

Проблемы

- ~~Mock/stub VS реальный бэкенд?~~
- ~~Подглядывание в black box~~
- ~~Флаканье~~
- ~~Автономность (Workstation/CI)~~
- ~~Воспроизводимость~~
- ~~Let’s localise it~~

Цели, требования

⏳ Time-to-Market

🕰️ Времени полного регресса

🚀 Ничего не сломано в релизе

⏱️/💵 Время VS ресурсы

👨‍🦲 Ручные QA -> Автоматизаторы

Результат

⏳ Time-to-Market —	2 недели 📉 1 неделя
🕰️ Времени полного регресса —	4-5 дней 📉 1-2 дня
🚀 Ничего не сломано в релизе —	Severity 📉 Уверенность📈
⏱️/💵 Время VS ресурсы —	Max результат / Min code change
👨‍🔧 Ручные QA -> Автоматизаторы —	????????????

Выводы

✅ E2E тесты работают сразу

✅ Минимум изменений в коде

✅ Максимальное приближение к prod

✅ Воспроизводимость!

🤔 Танцы с инфраструктурой

🤔 Прокачивание интеграционного API

Чеклист

- 🎯 Цель, ресурсы, результат
- 🎲 Для начала сделай как нибудь
- 🔄 Свой backend — реально, а 3rd party — mock
- 🗃️ Интеграционное API
- 👬 Взаимодействие с командой Backend
- 🐋 Изоляция окружения с Docker
- 👬 Взаимодействие с командой DevOps
- ✈️ On-the-fly тюнинг окружения (локализация)

https://github.com/bengus/e2e-testing-checklist

Виктор Брыксин — Как всё починить и ничего не сломать

Артем Разинов — Нативные UI тесты, которые работают

CocoaHeads Special Event

Артем Разинов, Avito

Нативные UI тесты, которые работают

https://dostavim.lenta.ch/