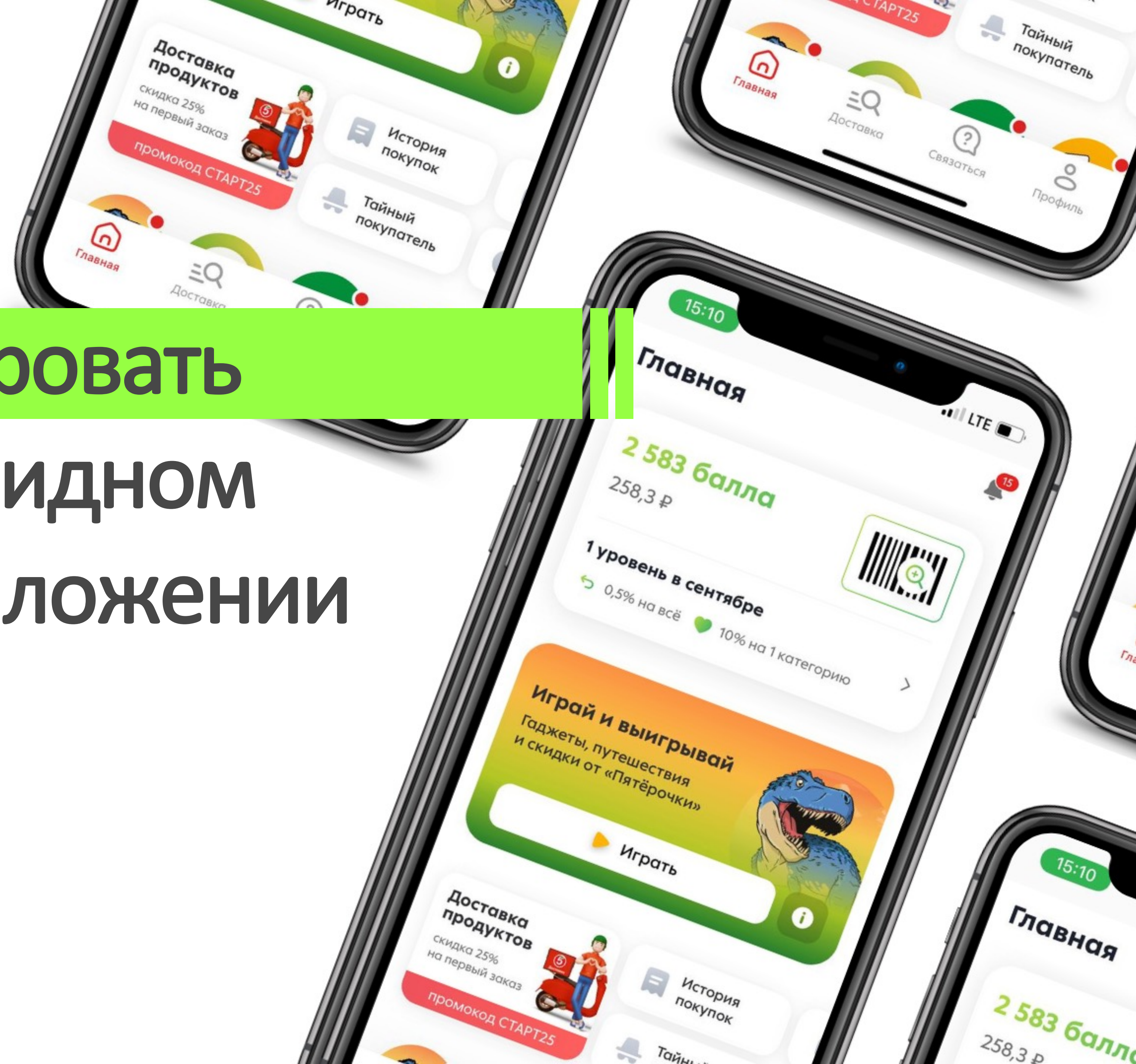


Как автоматизировать WebView в гибридном мобильном приложении

Андрей Попов и Эльвина Гранже



О нас



Андрей Попов

QA Automation (JS, Python)

Больше двух лет занимаюсь автоматизацией



Эльвина Гранже

QA Senior

Более 5 лет в тестировании

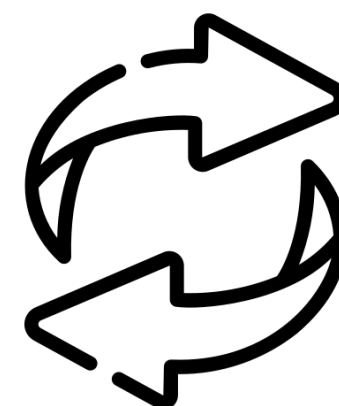
О чем будем говорить?



Какое у нас мобильное приложение?



Выбор инструментов автоматизации



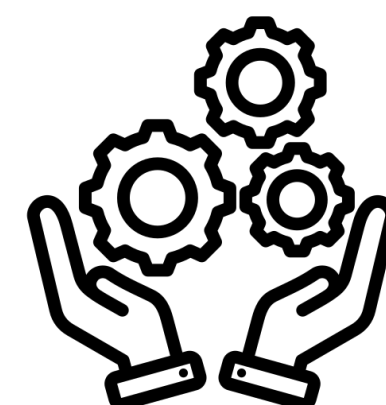
Обмен информацией между web и native?



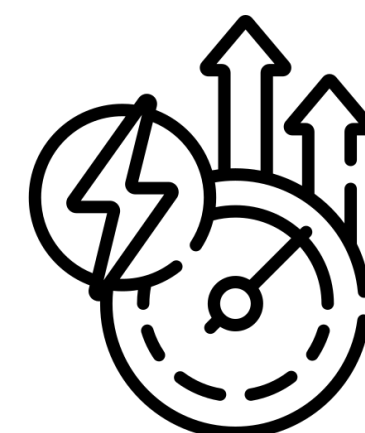
Настройка проекта



Подготовка платформы под начинающих тестировщиков



Примеры



Ускорение автотестов



Проблемы

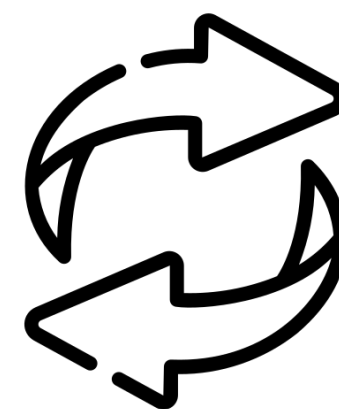
О чем будем говорить?



Какое у нас мобильное приложение?



Выбор инструментов автоматизации



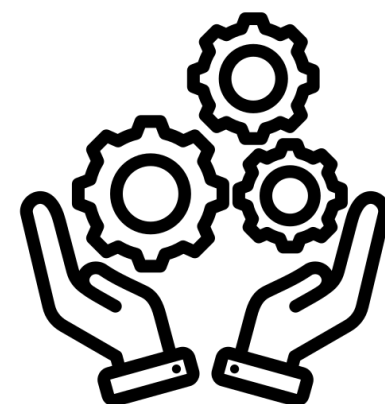
Обмен информацией между web и native?



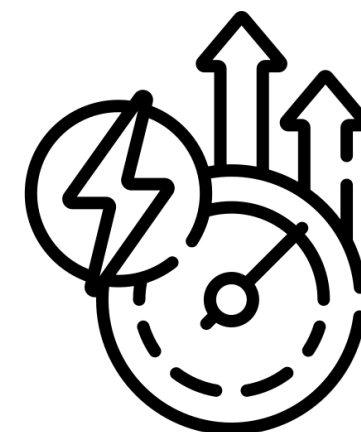
Настройка проекта



Подготовка платформы под начинающих тестировщиков



Примеры



Ускорение автотестов



Проблемы

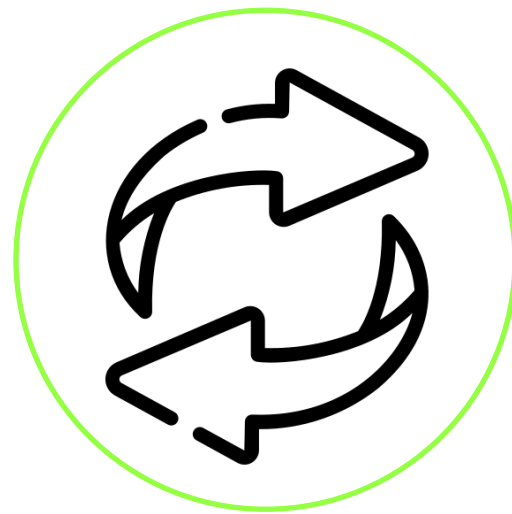
О чем будем говорить?



Какое у нас мобильное приложение?



Выбор инструментов автоматизации



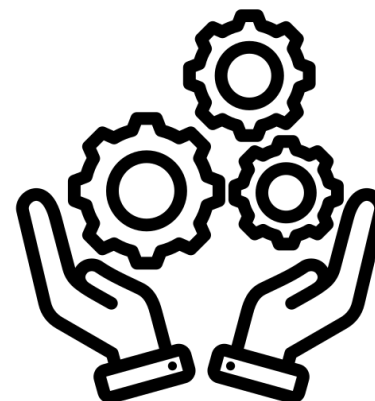
Обмен информацией между web и native?



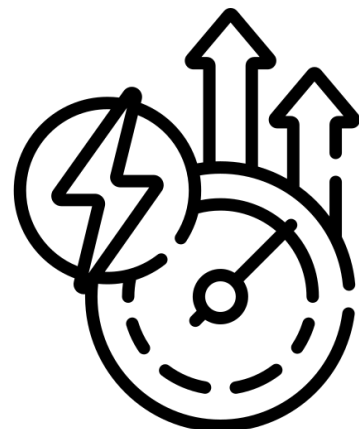
Настройка проекта



Подготовка платформы под начинающих тестировщиков



Примеры



Ускорение автотестов



Проблемы

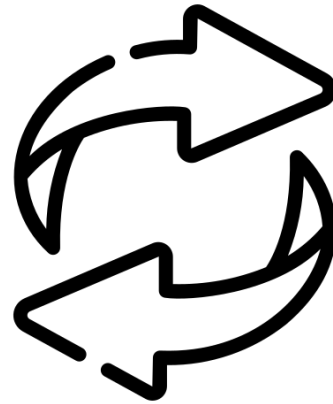
О чем будем говорить?



Какое у нас мобильное приложение?



Выбор инструментов автоматизации



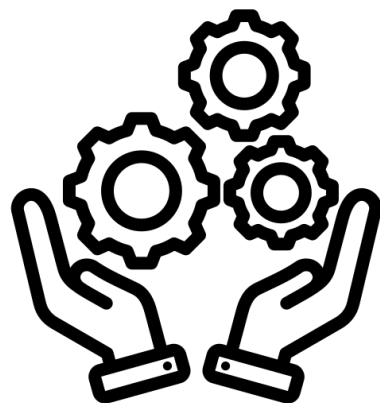
Обмен информацией между web и native?



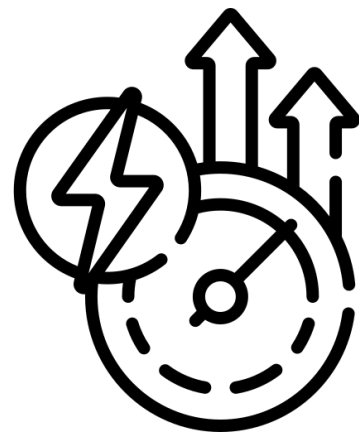
Настройка проекта



Подготовка платформы под начинающих тестировщиков



Примеры



Ускорение автотестов



Проблемы

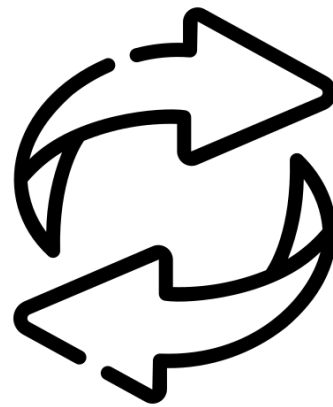
О чем будем говорить?



Какое у нас мобильное приложение?



Выбор инструментов автоматизации



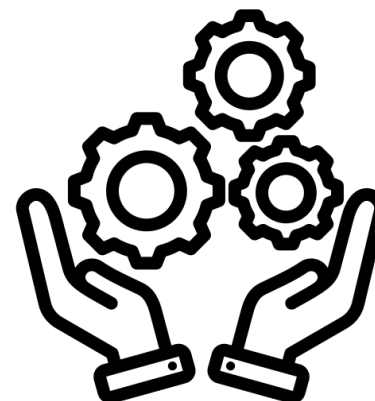
Обмен информацией между web и native?



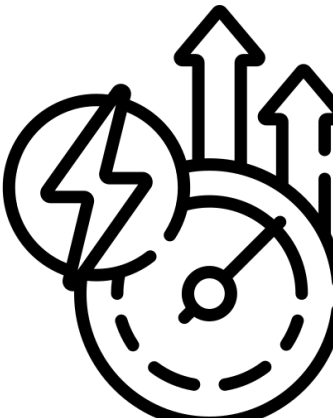
Настройка проекта



Подготовка платформы под начинающих тестировщиков



Примеры



Ускорение автотестов



Проблемы

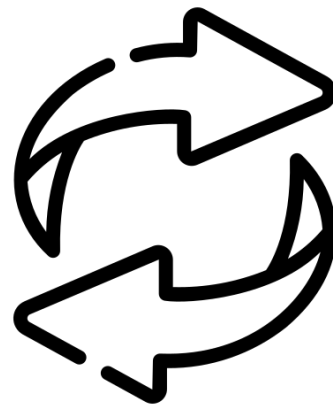
О чем будем говорить?



Какое у нас мобильное приложение?



Выбор инструментов автоматизации



Обмен информацией между web и native?



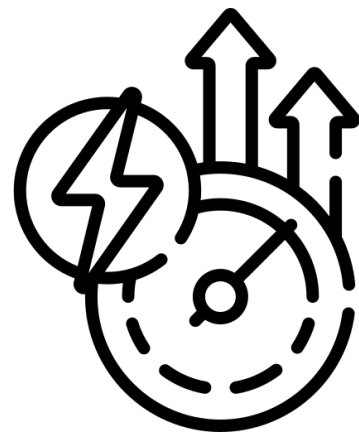
Настройка проекта



Подготовка платформы под начинающих тестировщиков



Примеры



Ускорение автотестов



Проблемы

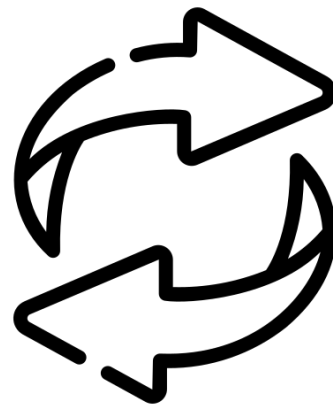
О чем будем говорить?



Какое у нас мобильное приложение?



Выбор инструментов автоматизации



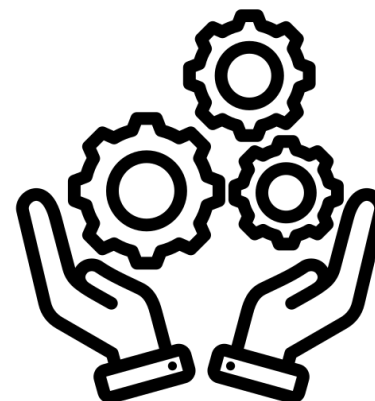
Обмен информацией между web и native?



Настройка проекта



Подготовка платформы под начинающих тестировщиков



Примеры



Ускорение автотестов



Проблемы

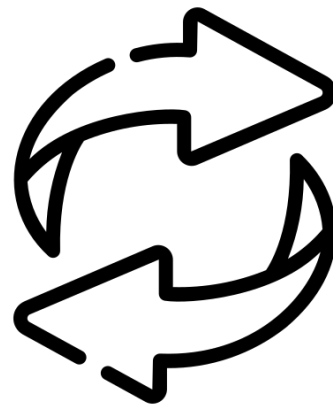
О чем будем говорить?



Какое у нас мобильное приложение?



Выбор инструментов автоматизации



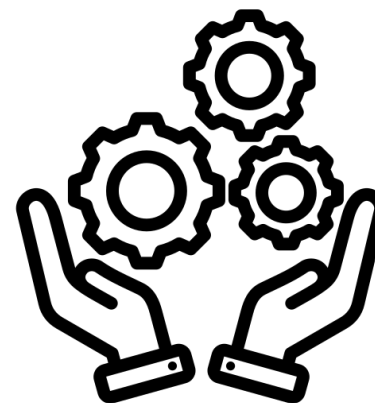
Обмен информацией между web и native?



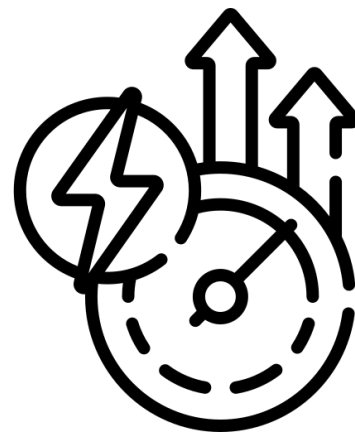
Настройка проекта



Подготовка платформы под начинающих тестировщиков



Примеры

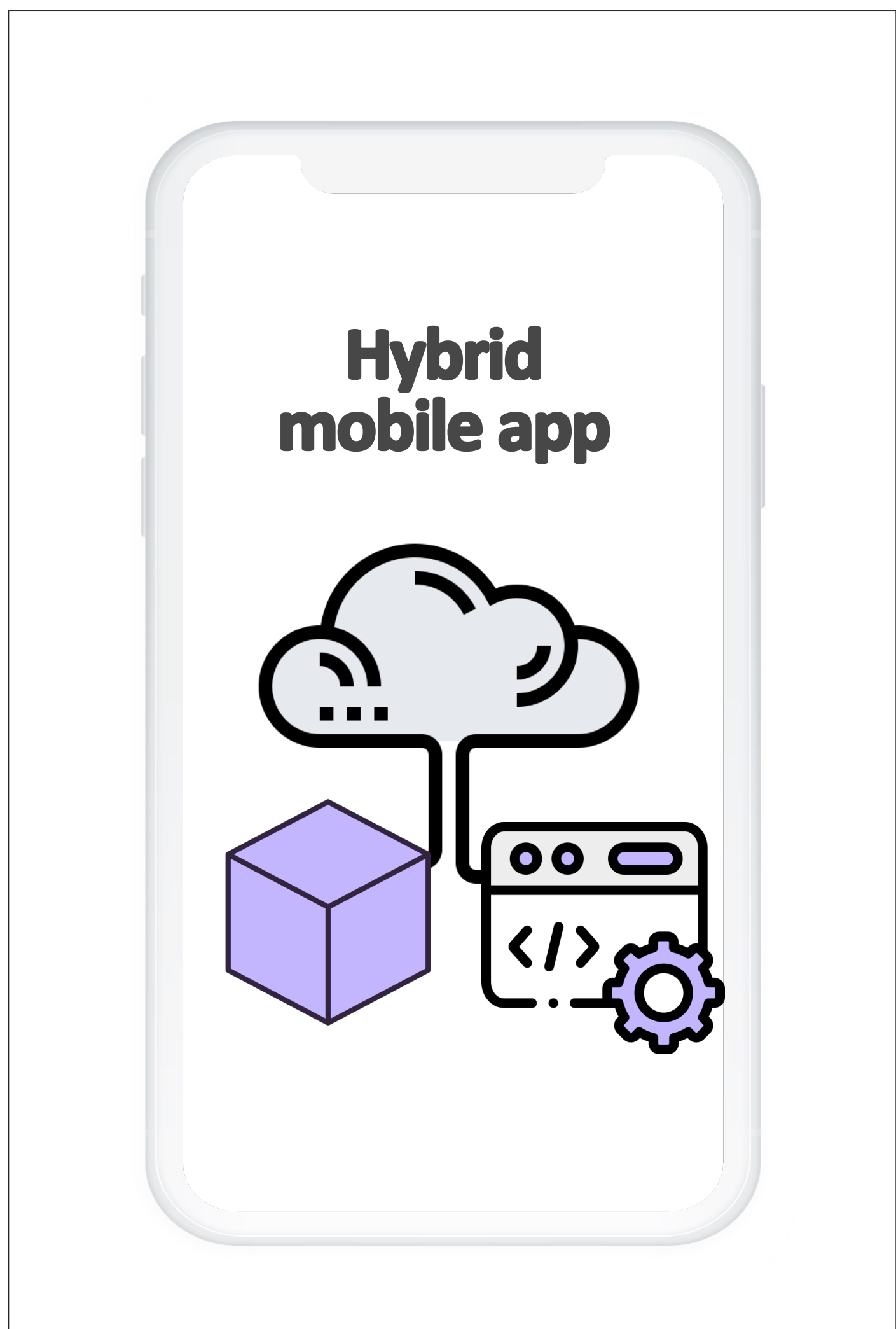
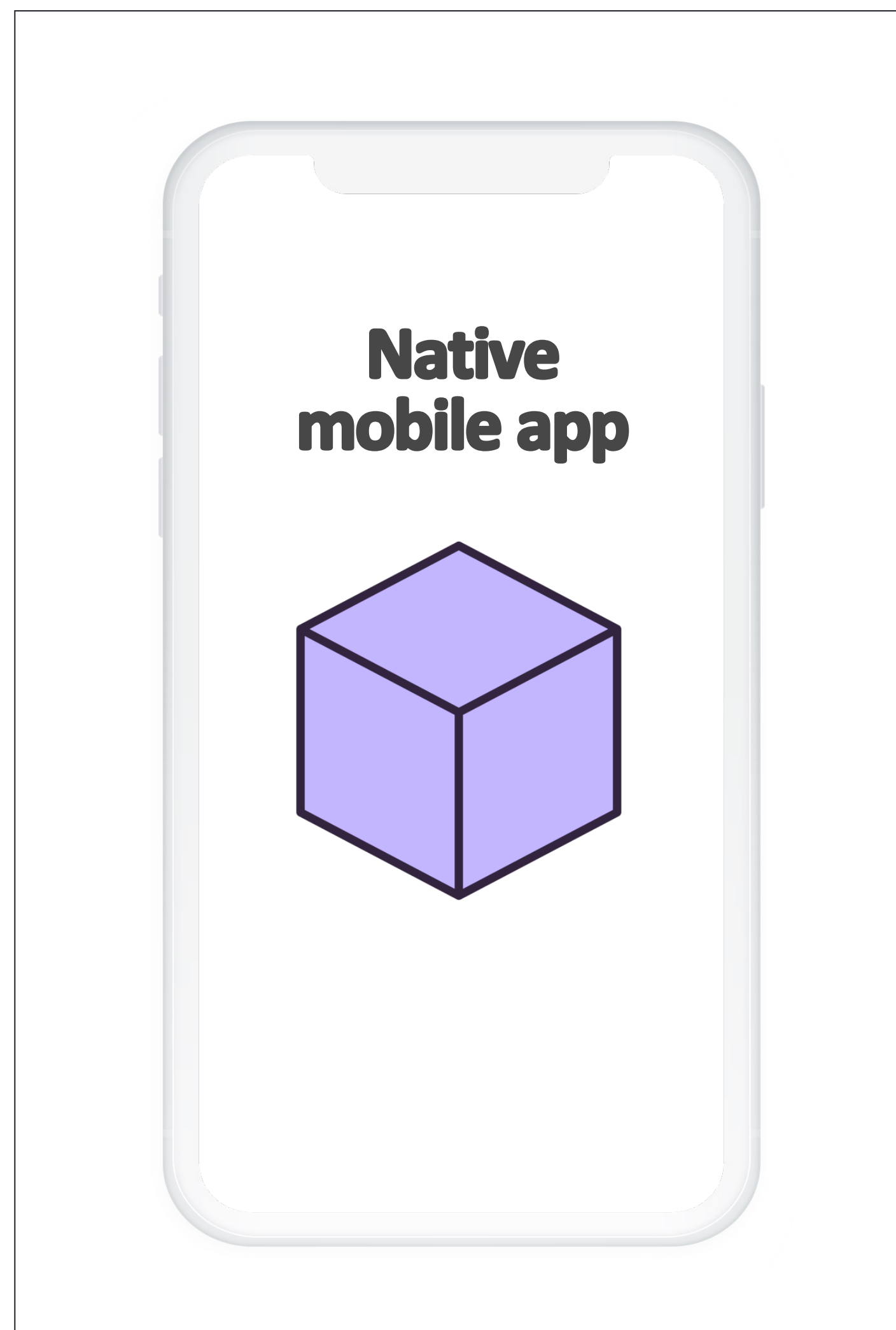


Ускорение автотестов



Проблемы

Типы мобильных приложений



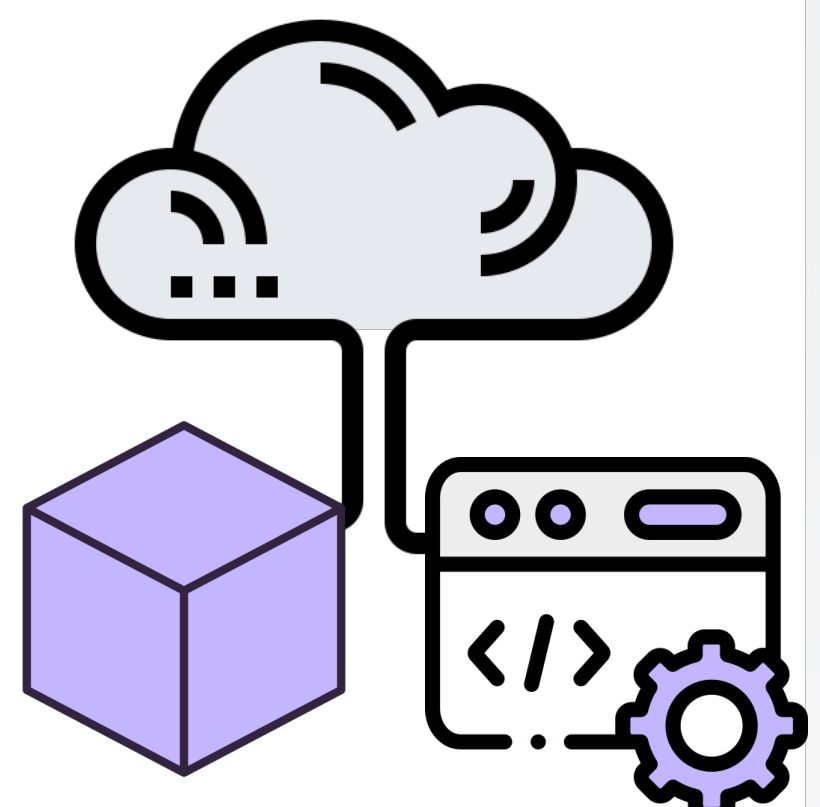
Типы мобильных приложений

Native mobile app

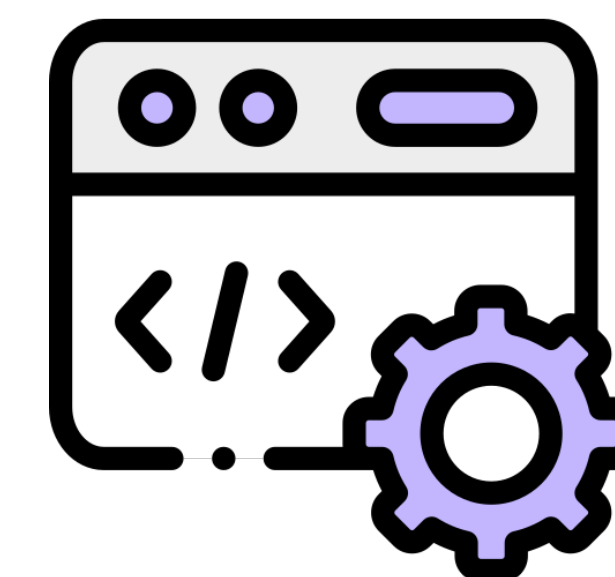


- ✓ Производительность и надежность
- ✓ Оффлайн-доступ
- ✓ Интеграция с железом (камера, список контактов, геолокация)
- ✓ Исключена кроссплатформенность
- ✓ Невысокая скорость разработки
- ✓ Узконаправленность разработки

Hybrid mobile app

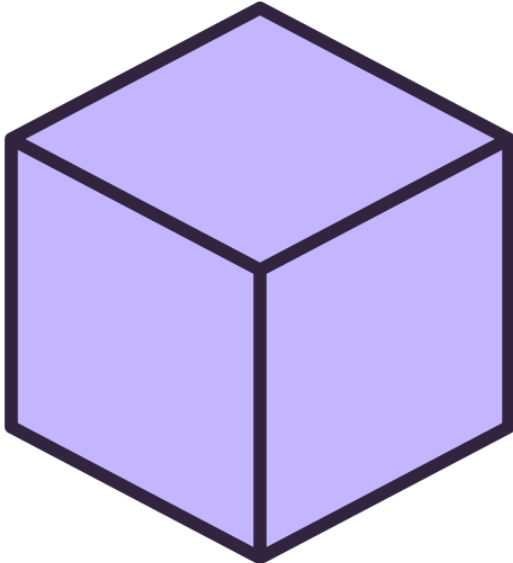


Web mobile app

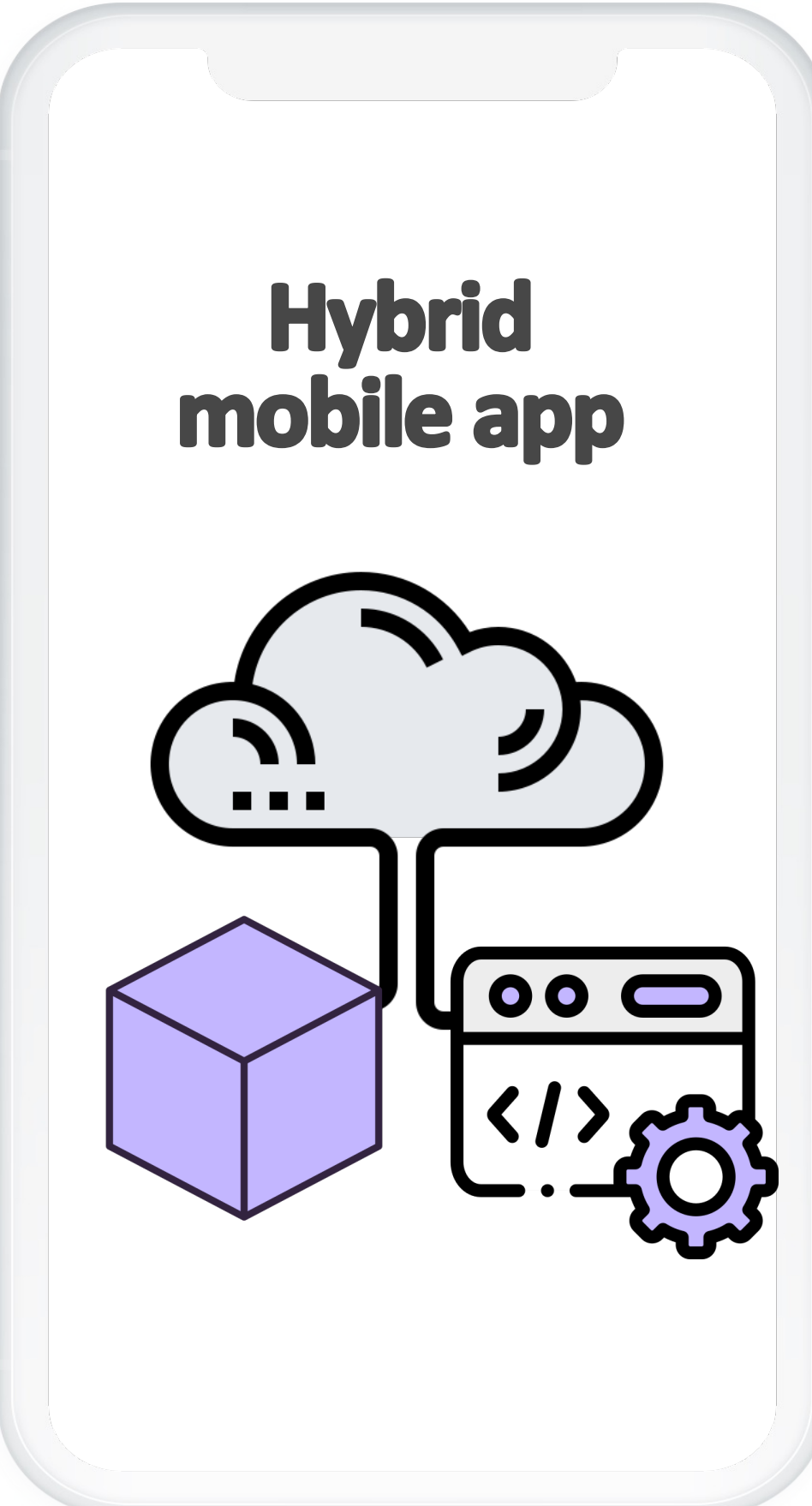


Типы мобильных приложений




Native mobile app



Hybrid mobile app

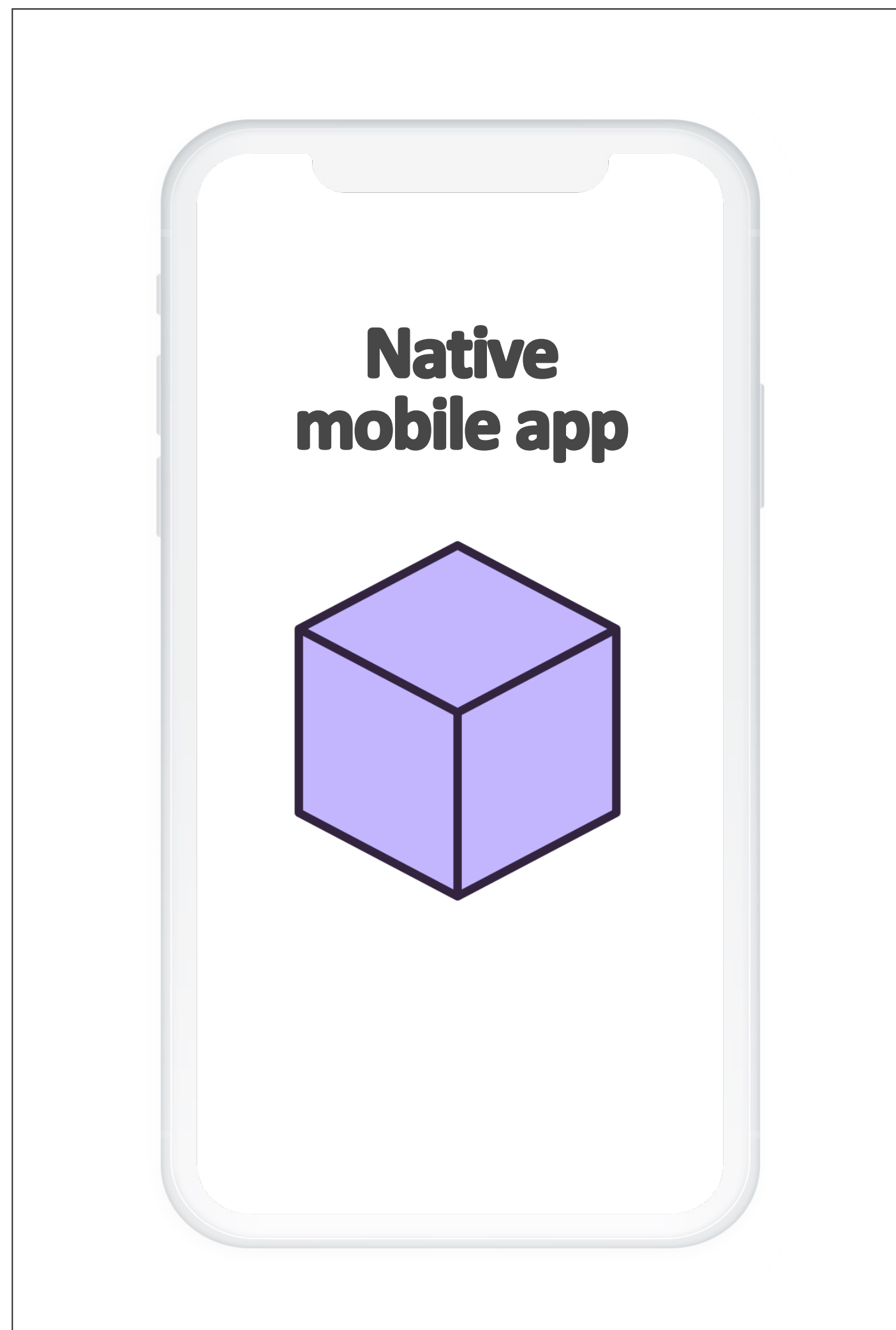


Web mobile app





   Google Analytics

- Кроссплатформенность
- Адаптивная верстка
- Не требуется установка отдельного приложения
- Скорость отклика
- Отсутствие доступа к нативным функциям девайса

Типы мобильных приложений

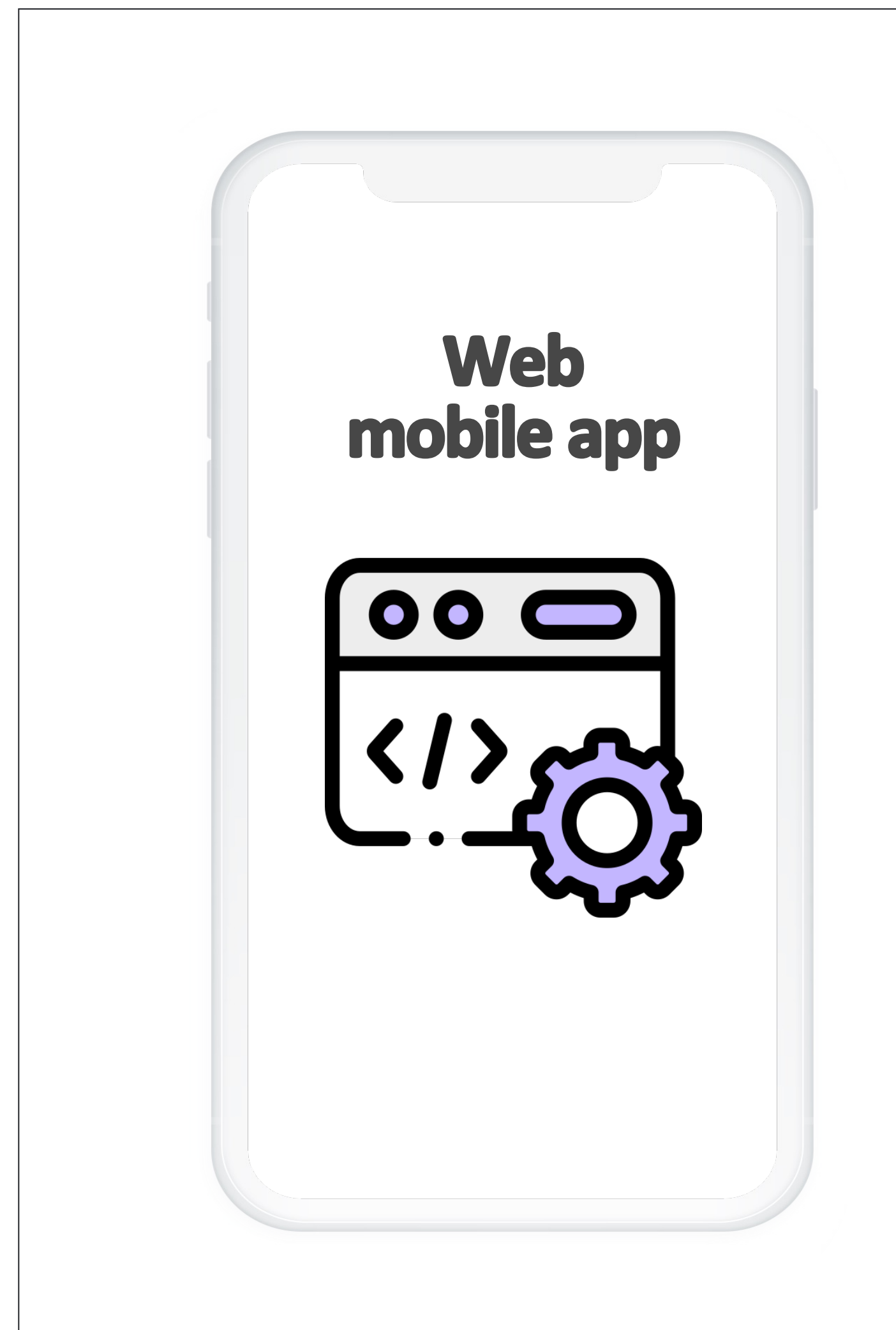


Hybrid mobile app

- Скорость разработки
- Доступ к нативным возможностям устройств
- Дистрибьюция для нескольких платформ
- Независимое обновление

- Обратная совместимость
- Производительность ниже
- Интеграция сторонних сервисов
- Слишком много тестирования



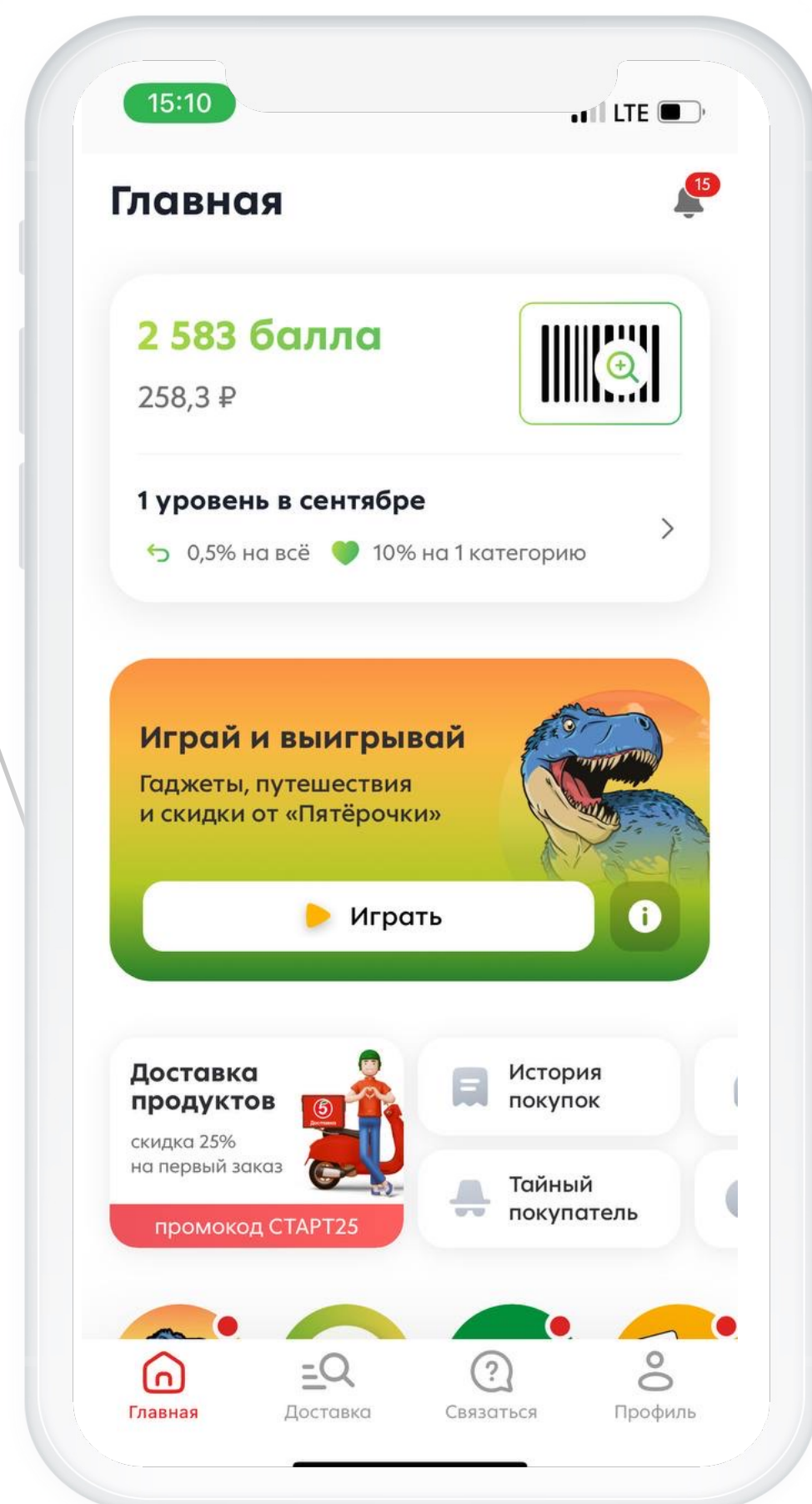
Мобильное приложение «Пятёрочка»

является гибридным МП, что позволяет сокращать time-to-market, и не зависеть от обновления в сторах



Google Play

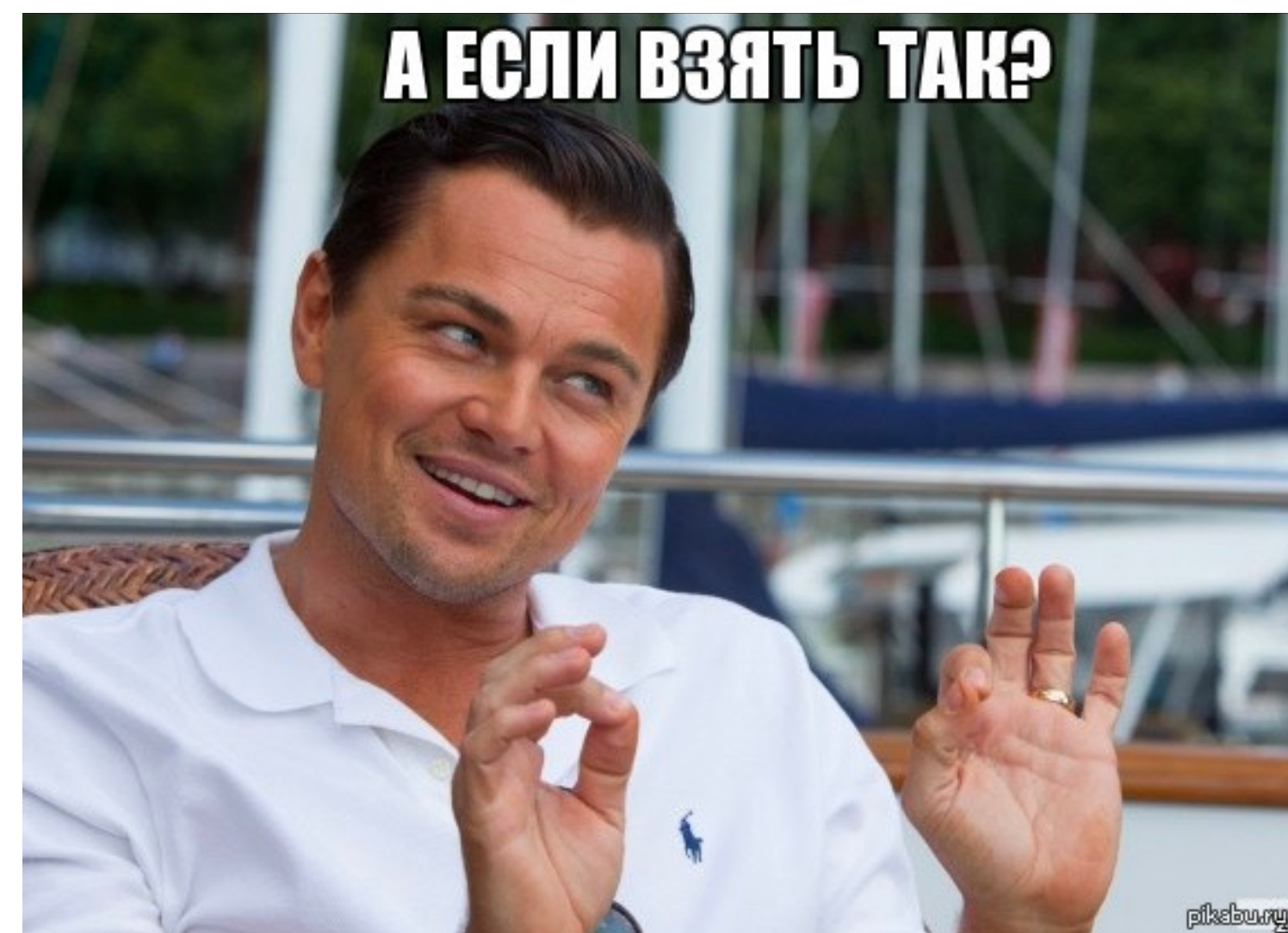
- 4,2 оценка
- 131 тыс отзывов
- 10 млн + скачиваний



App Store

- 4,5 оценка
- 24,8 тыс отзывов

Как тестировать такое приложение?



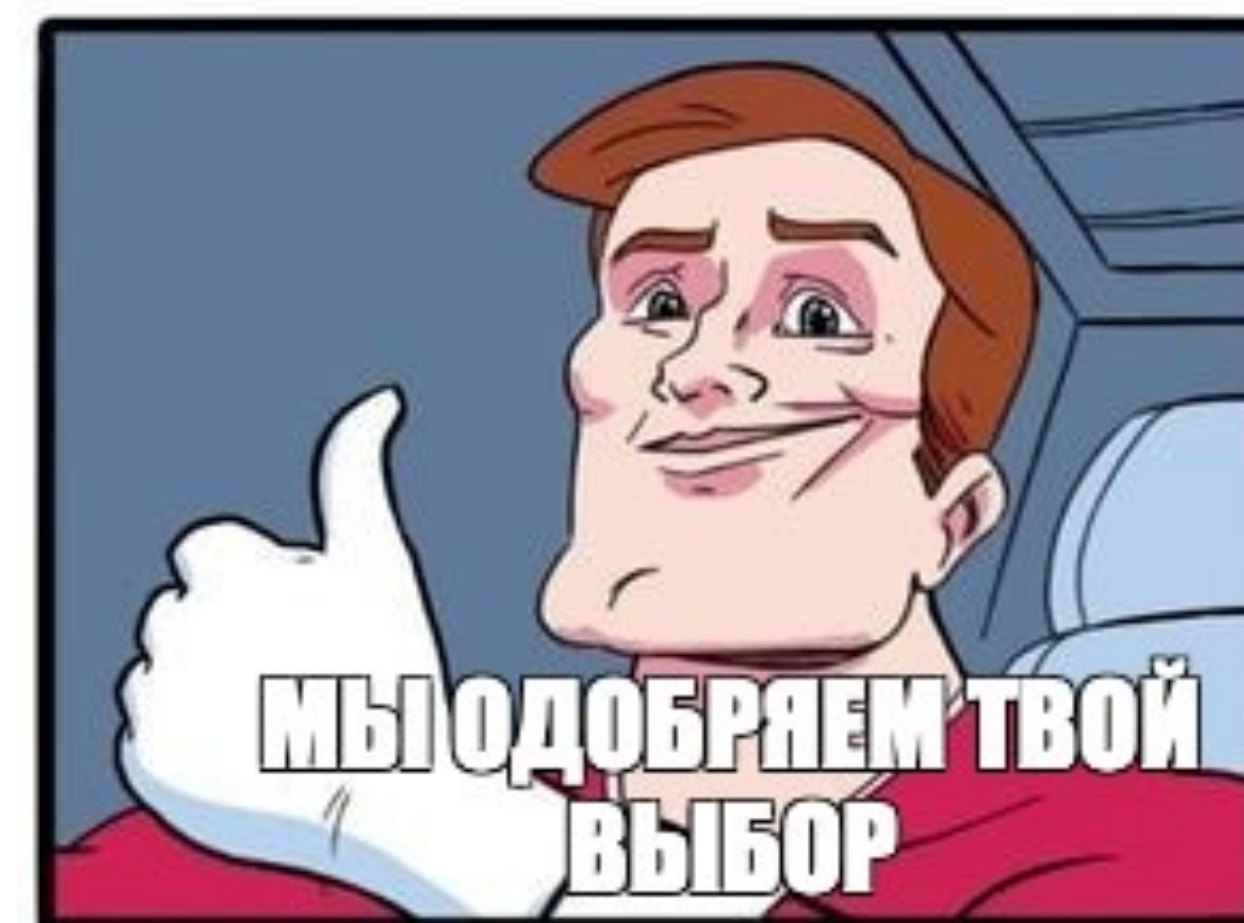
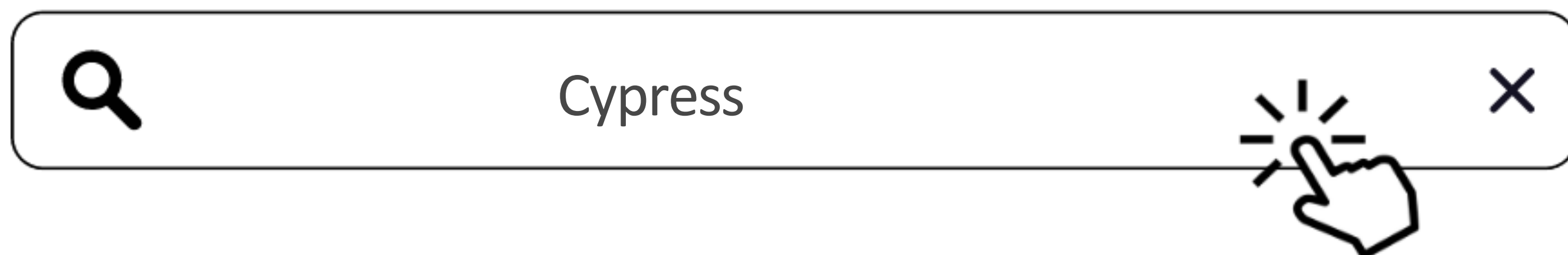
Мы разделили процесс тестирования на:

- ✓ Native
- ✓ Web
- ✓ API

Выбор инструмента для автоматизации web

У нас есть экспертиза и внутреннее комьюнити на Cypress, поэтому мы выбрали его (вы можете использовать любой удобный фреймворк)

Selenium vs Puppeteer vs Cypress vs Playwright



Выбор инструмента для автоматизации web

- ✓ Путешествие во времени
- ✓ Возможность отладки
- ✓ Автоматическое ожидание
- ✓ Шпионы, заглушки и часы
- ✓ Управление сетевым трафиком

```
Specs ✓ 8 ✗ -- 🔄 -- ▼ ■  
  
actions.cy.js 03:55  
  
▼ Actions  
✓ .type() - type into a DOM element  
✓ .focus() - focus on a DOM element  
✓ .blur() - blur off a DOM element  
✓ .clear() - clears an input or textarea element  
✓ .submit() - submit a form  
  
▼ BEFORE EACH  
1 visit http://localhost:8080/commands/actio...  
  
▼ TEST BODY  
1 get .action-form  
2 -find [type="text"]  
3 -type HALFOFF  
4 get .action-form  
5 -submit  
6 -next  
7 -assert expected <p> to contain Your form has been submitted!  
  
✓ .click() - click on a DOM element  
✓ .dblclick() - double click on a DOM element
```

Выбор инструмента для автоматизации web

- ✓ Путешествие во времени
- ✓ Возможность отладки
- ✓ Автоматическое ожидание
- ✓ Шпионы, заглушки и часы
- ✓ Управление сетевым трафиком

Console Emulation Rendering

<top frame> Preserve log

Console was cleared app.js:3

Command: type app.js:3

Typed: Jane Lane app.js:3

Applied To: `<input name="firstName" id="firstName" type="text">` app.js:3

▼ Key Events Table app.js:2

(index)	typed	which	keydown	keypress	textInput	input	keyup
1	"j"	74	true	true	true	true	true
2	"a"	65	true	true	true	true	true
3	"n"	78	true	true	true	true	true
4	"e"	69	true	true	true	true	true
5	" "	32	true	true	true	true	true
6	"L"	76	true	true	true	true	true
7	"a"	65	true	true	true	true	true
8	"n"	78	true	true	true	true	true
9	"e"	69	true	true	true	true	true

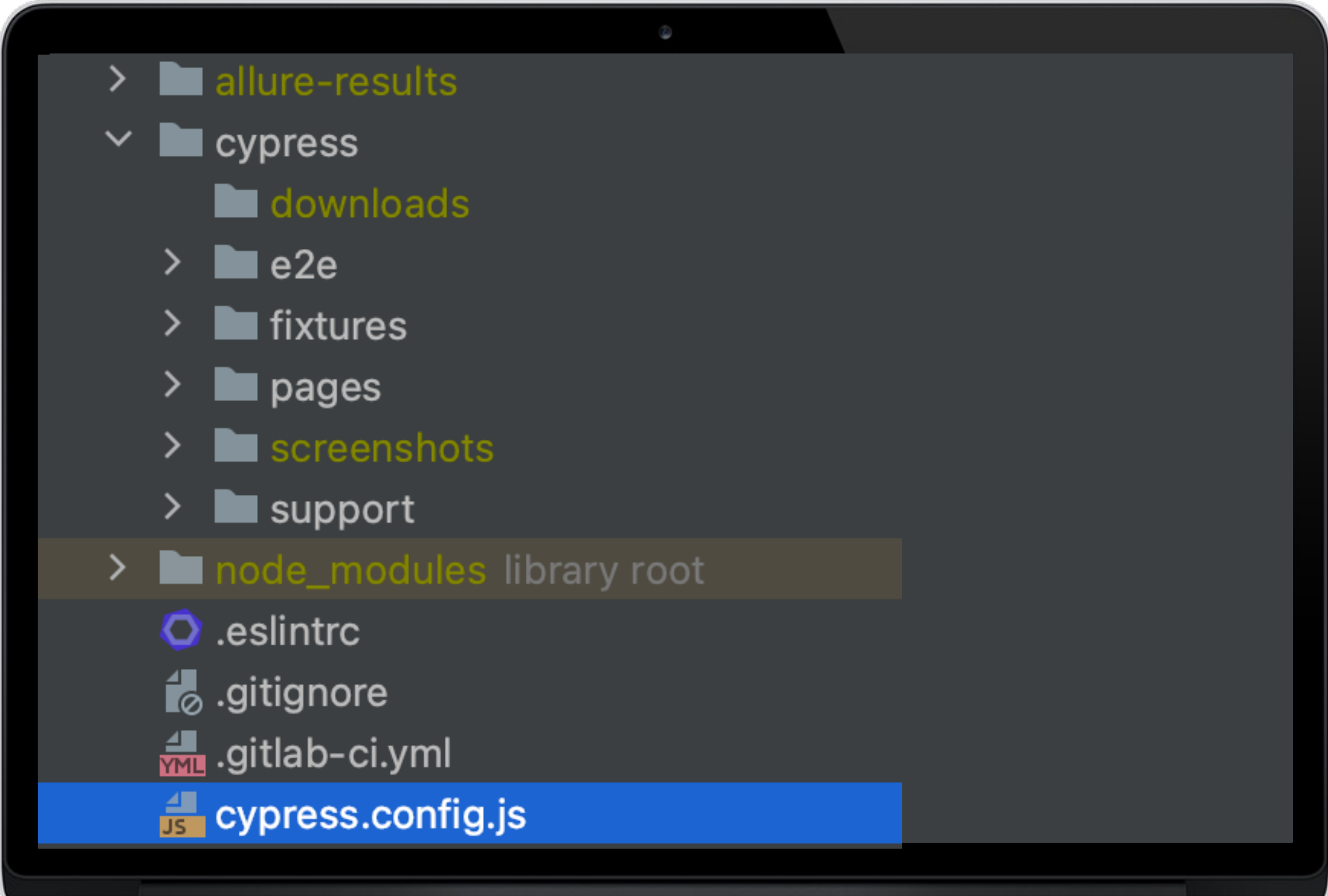
> |

Переход к проекту на Cypress



Структура проекта

Cypress.config.js –
основной файл по
настройке проекта



```
> allure-results
v cypress
  > downloads
  > e2e
  > fixtures
  > pages
  > screenshots
  > support
  > node_modules library root
.eslintrc
.gitignore
.gitlab-ci.yml
cypress.config.js
```

Настройки Cypress



cypress.config.js

```
module.exports = defineConfig({  
  projectId: '5kaWEB',  
  chromeWebSecurity: false,  
  pageLoadTimeout: 10000,  
  defaultCommandTimeout: 5000,  
  watchForFileChanges: false,  
  blockHosts: ['*analytics.x5.ru', '*analytics-5.x5.ru'],  
  ...  
});
```

- ✓ ChromeWebSecurity для работы с iframe-ами из различных источников
- ✓ Блокируем хосты с аналитикой

Настройки Cypress



cypress.config.js

```
env: {  
  allureResultsPath: './allure-results',  
  allure: true,  
  allureLogCypress: false,  
  allureClearSkippedTests: true  
},  
e2e: {  
  retries: {  
    runMode: 1,  
    openMode: 0  
  }  
},
```

- ✓ Стандартные настройки для аллюр плагина
- ✓ Будет 1 перезапуск при падении теста в headless режиме
- ✓ 0 перезапусков в режиме отладки

Настройки Cypress для автоматизации МП «Пятёрочка»

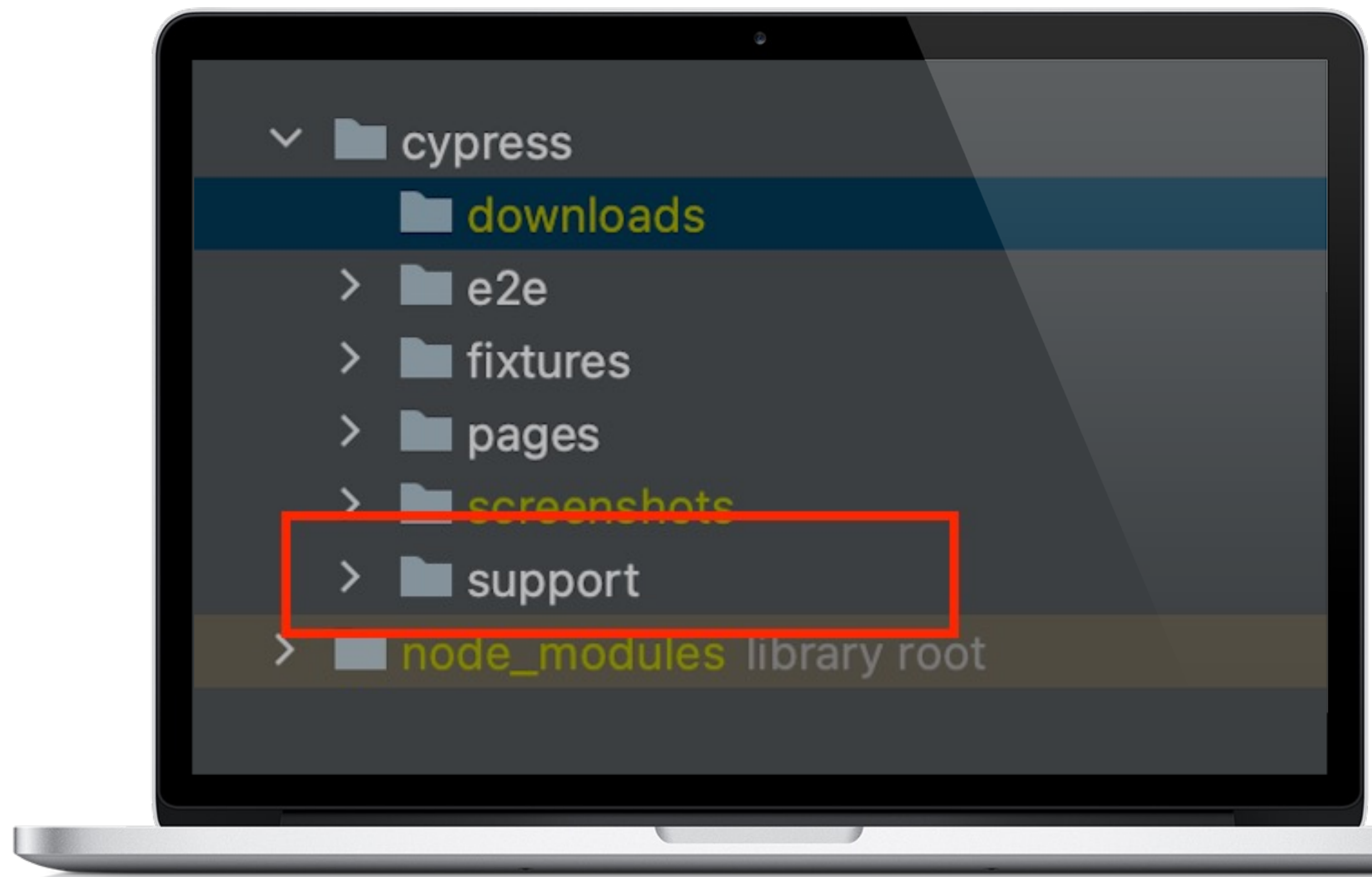


cypress.config.js

```
video: false,  
downloadsFolder: 'cypress/downloads',  
viewportWidth: 428,  
viewportHeight: 926,  
baseUrl: 'https://stage.pyaterochka.ru/',  
setupNodeEvents(on, config) {  
  on('task', {downloadFile})  
  AllureWriter(on, config)  
  require('@cypress/grep/src/plugin')(config)  
  return config  
}
```

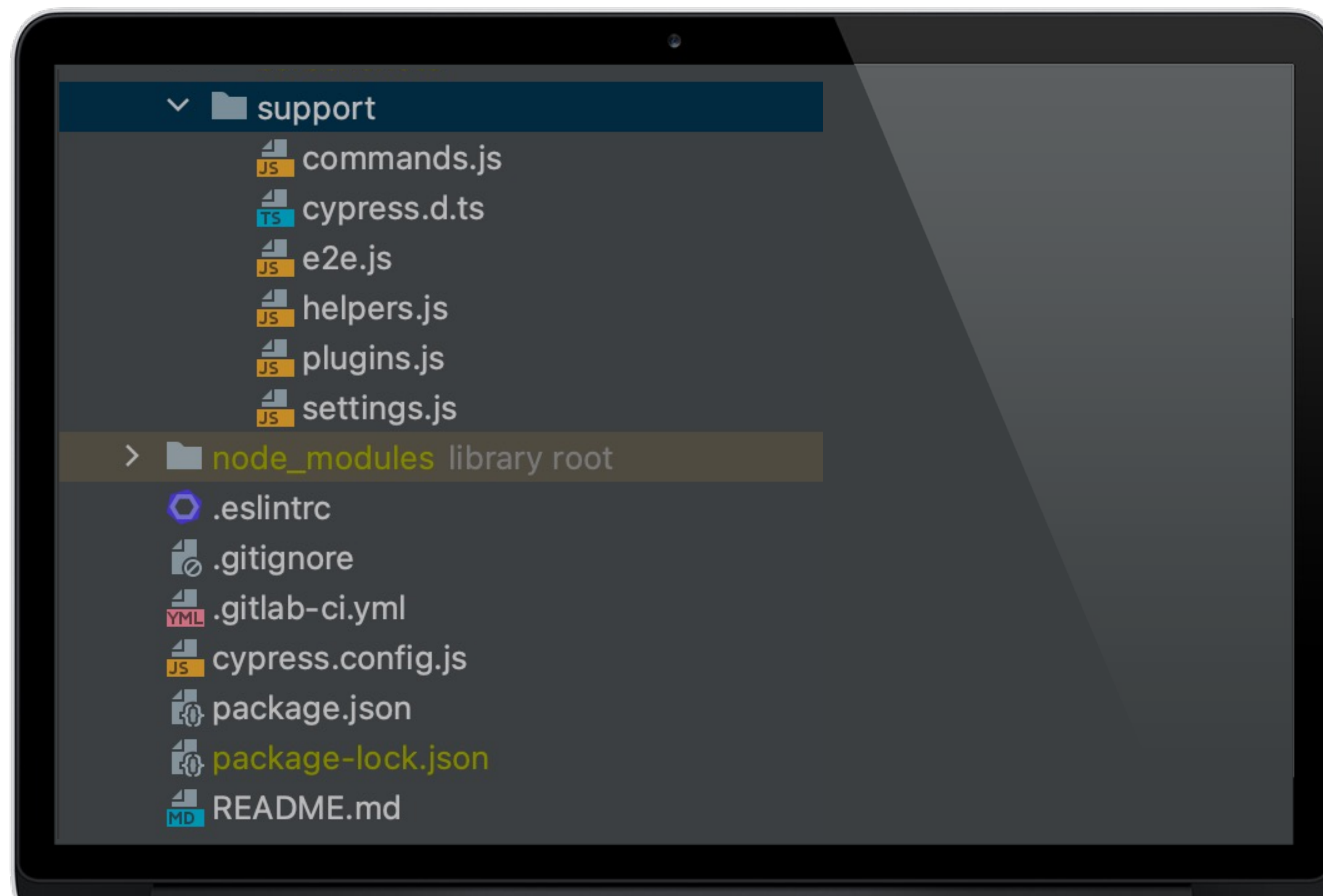
- ✓ Как показал опыт, запись видео используется крайне редко
- ✓ Размеры экрана по умолчанию

Структура проекта



Структура проекта

Support – храним
плагины, настройки и т.п.



Структура проекта

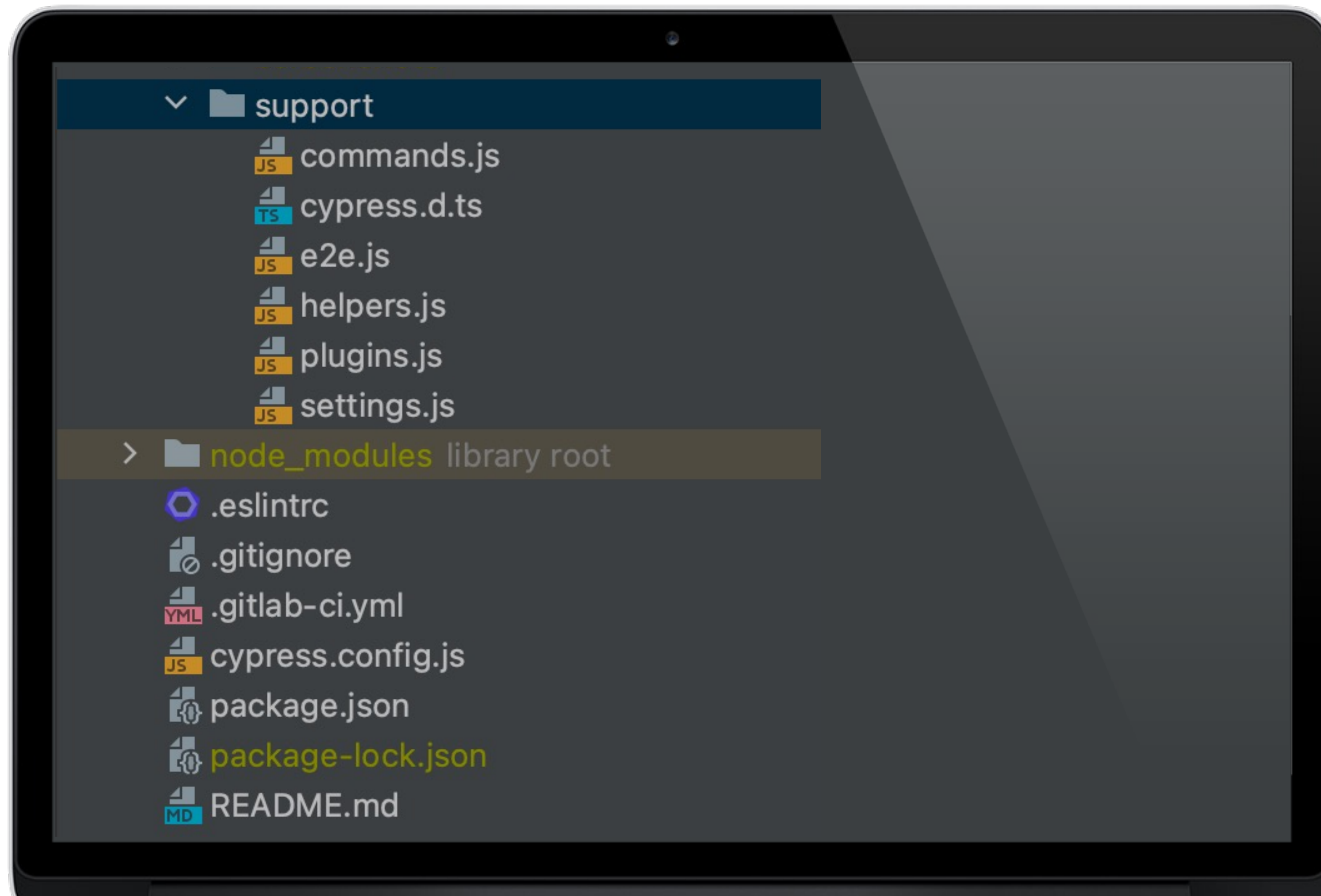
cypress.d.ts

```
TS cypress.d.ts x
25  /**
26   * Авторизация
27   * @param {string} number - номер телефона
28   * @example
29   * cy.login(number)
30   */
31  login(number: string): Chainable
```

```
Cypress.Chainable.login(
  number: string): Chainable
```

Авторизация
Example:
cy.login(number)

Params: number – номер телефона
cypress/support/cypress.d.ts



Авторизация



Как устроен наш гибрид

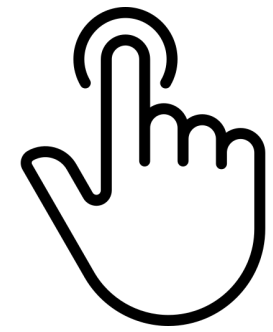


WEB

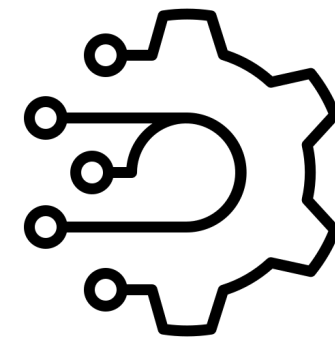
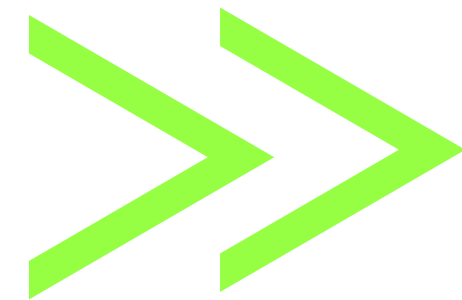


NATIVE

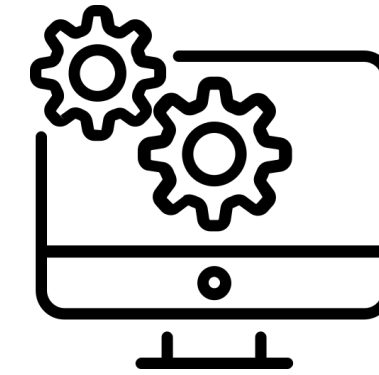
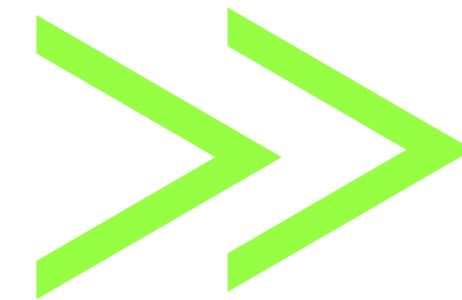
Авторизация



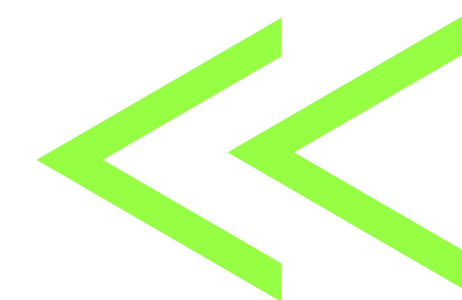
Нажимаем на кнопку «Войти»



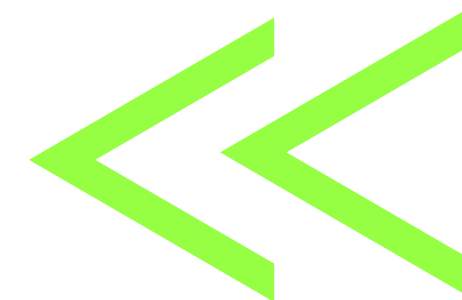
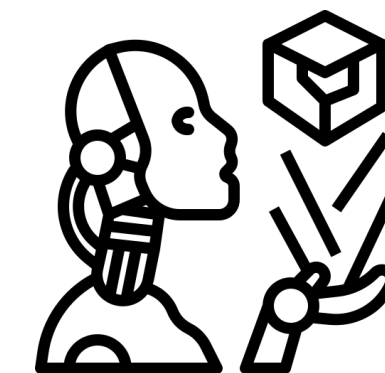
Запрос к нативной части



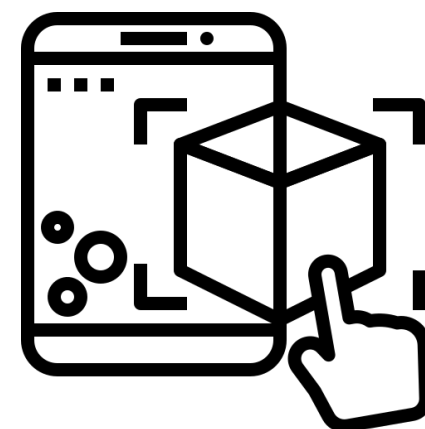
Натив обращается к сервису авторизации



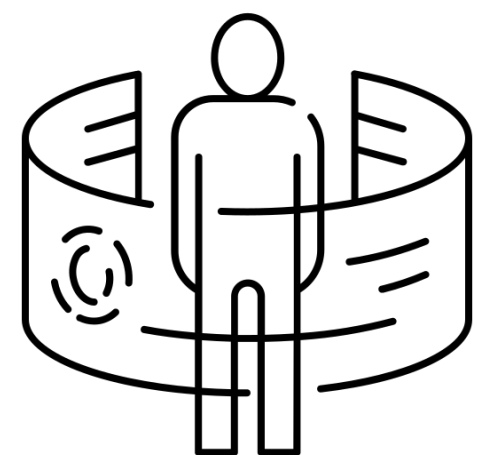
Натив получает токен



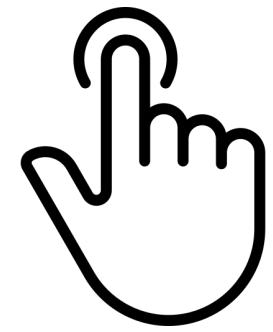
Передача токена в веб



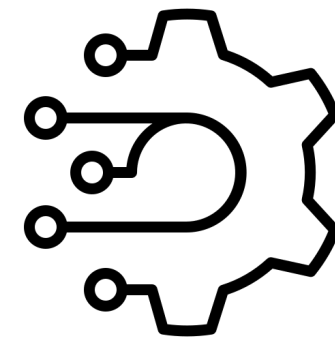
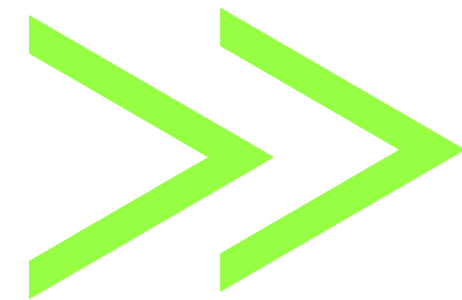
Авторизован



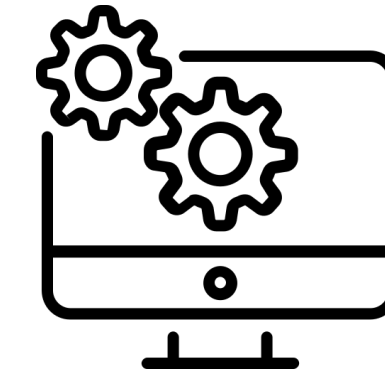
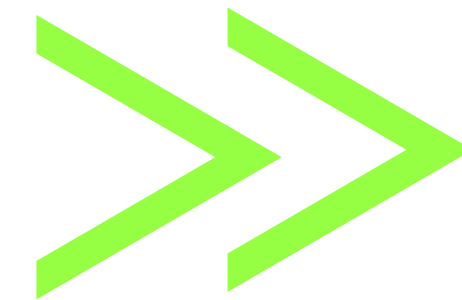
Авторизация



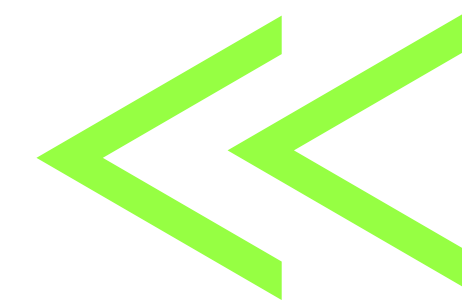
Нажимаем на кнопку «Войти»



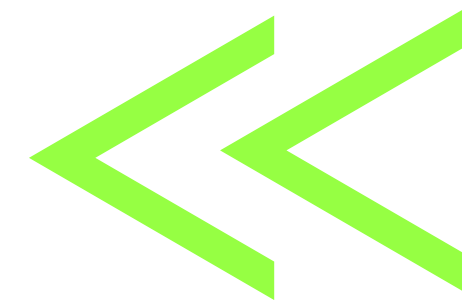
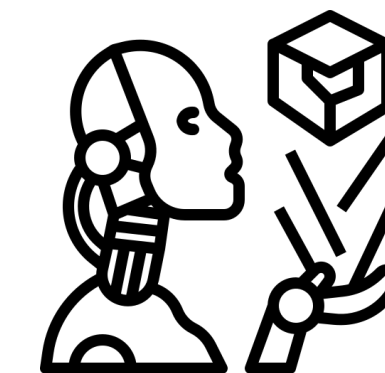
Запрос к нативной части



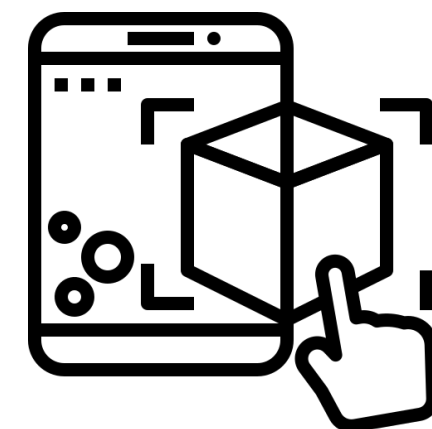
Натив обращается к сервису авторизации



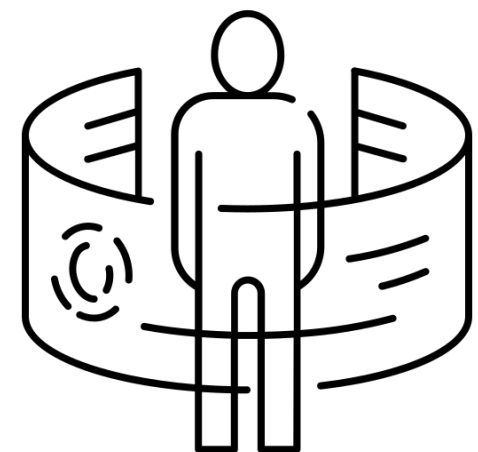
Натив получает токен



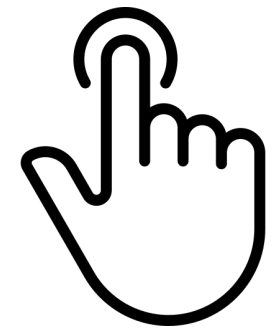
Передача токена в веб



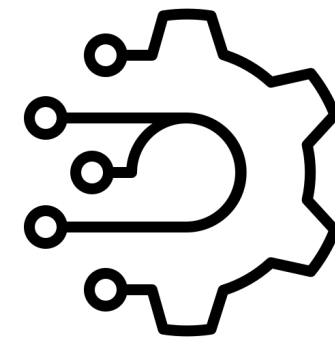
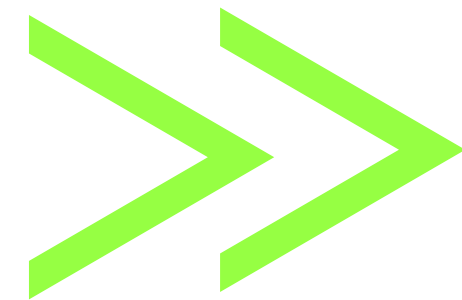
Авторизован



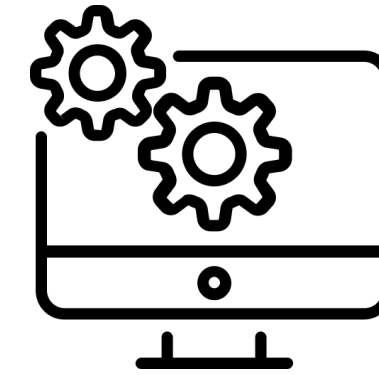
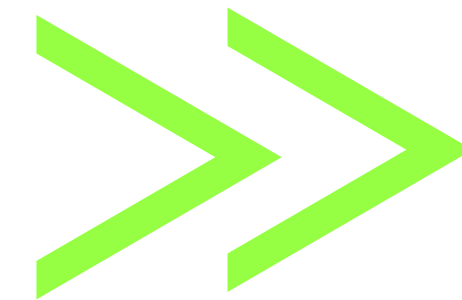
Авторизация



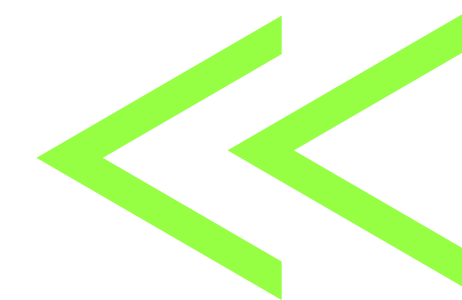
Нажимаем на кнопку «Войти»



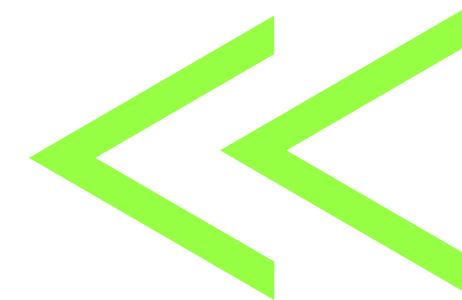
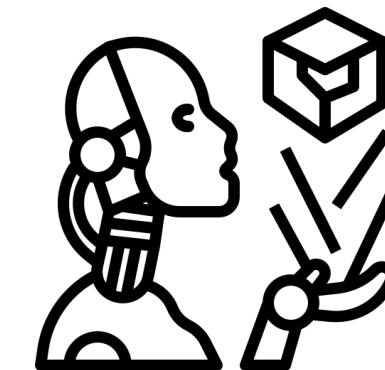
Запрос к нативной части



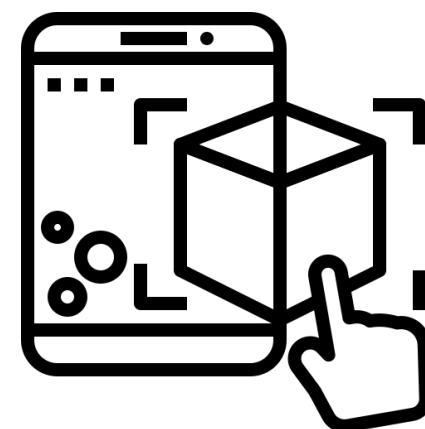
Натив обращается к сервису авторизации



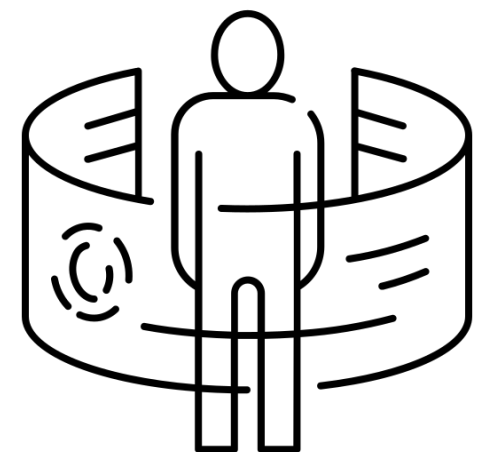
Натив получает токен



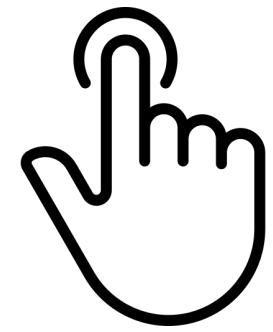
Передача токена в веб



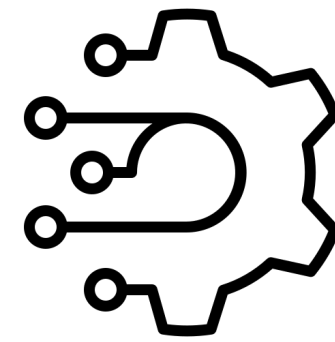
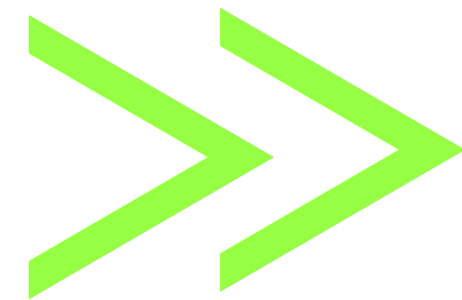
Авторизован



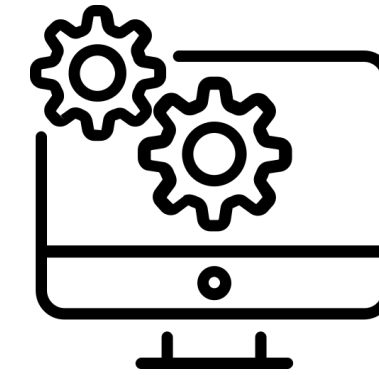
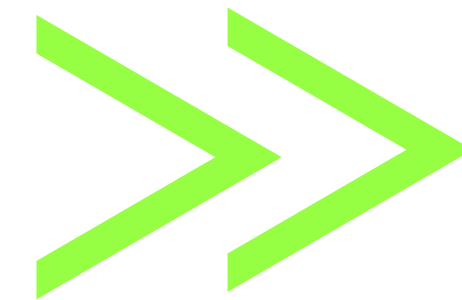
Авторизация



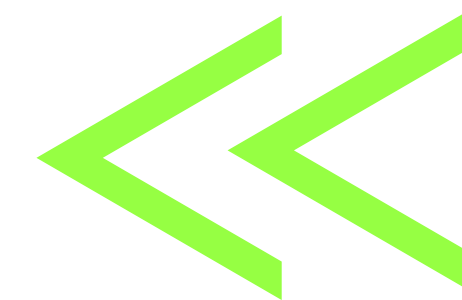
Нажимаем на кнопку «Войти»



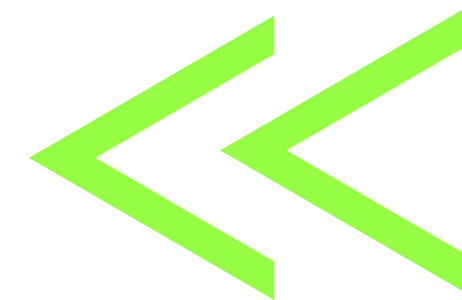
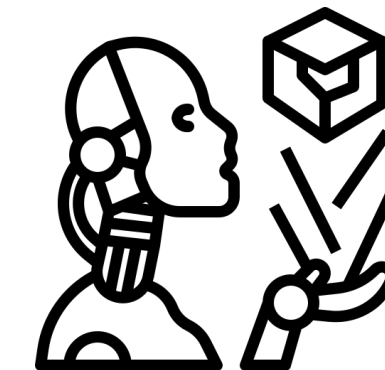
Запрос к нативной части



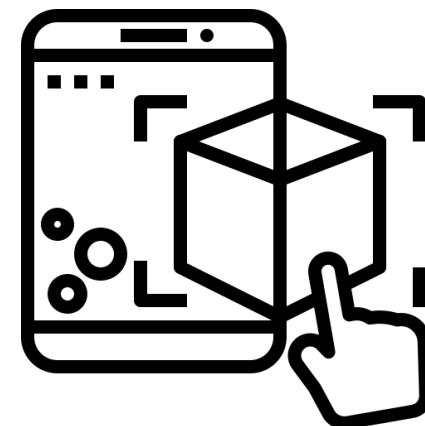
Натив обращается к сервису авторизации



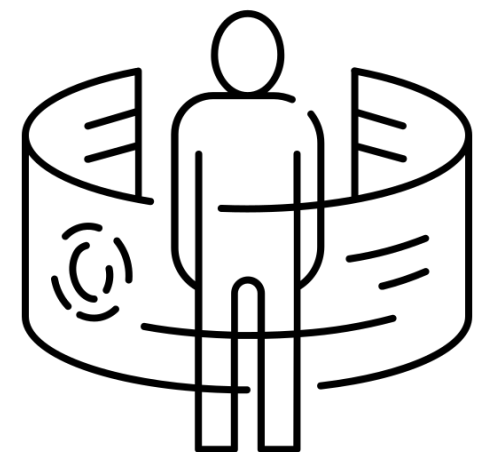
Натив получает токен



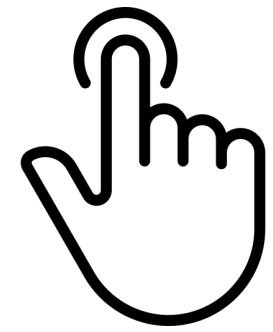
Передача токена в веб



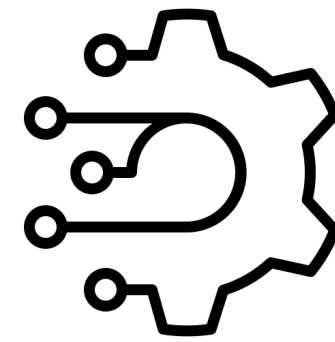
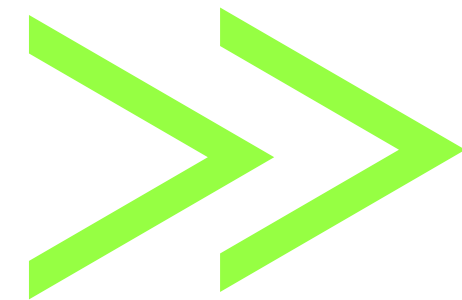
Авторизован



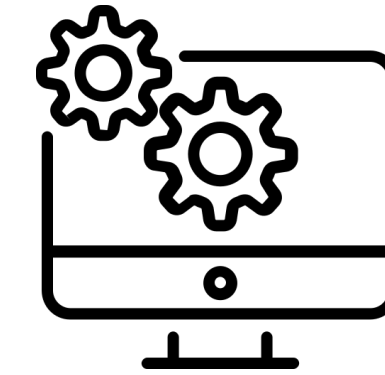
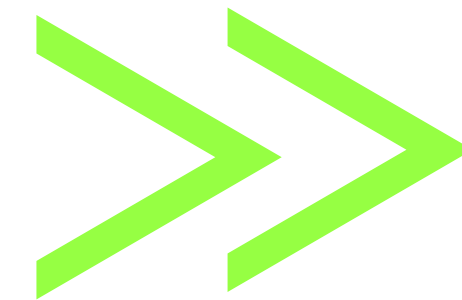
Авторизация



Нажимаем на кнопку «Войти»



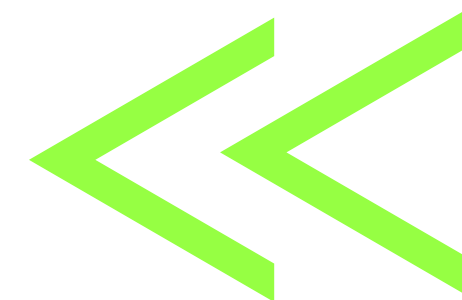
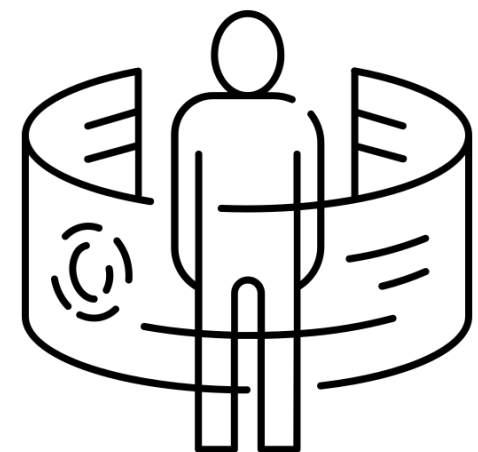
Запрос к нативной части



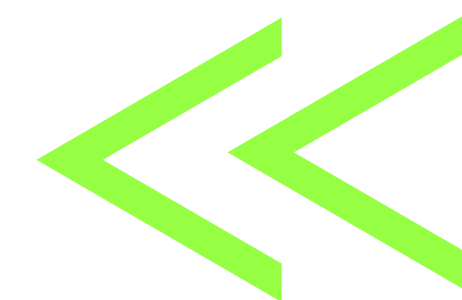
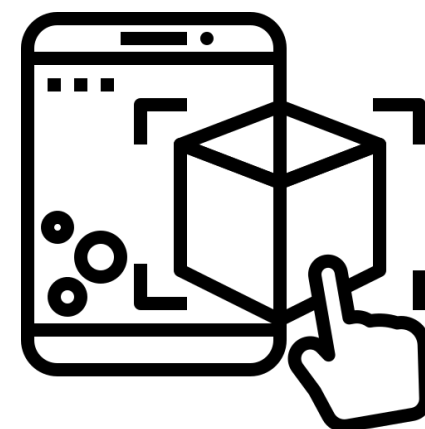
Натив обращается к сервису авторизации



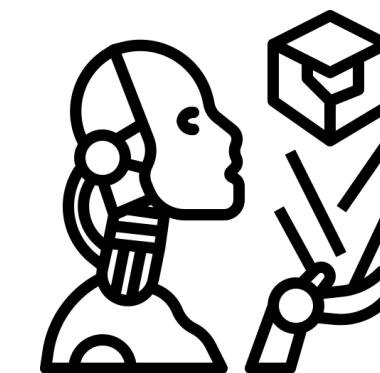
Авторизован



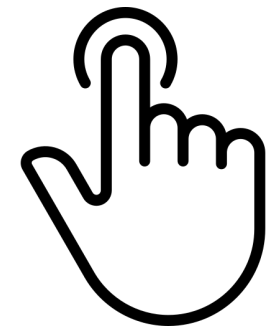
Передача токена в веб



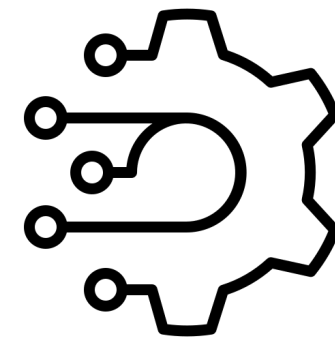
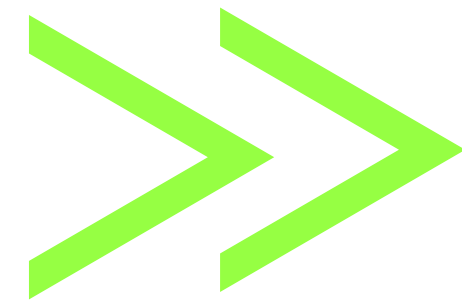
Натив получает токен



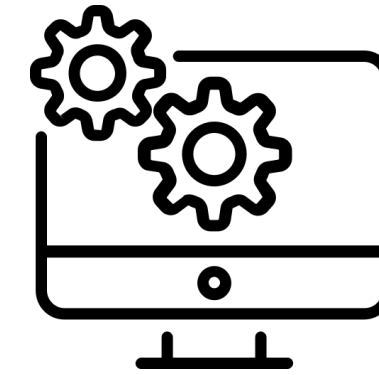
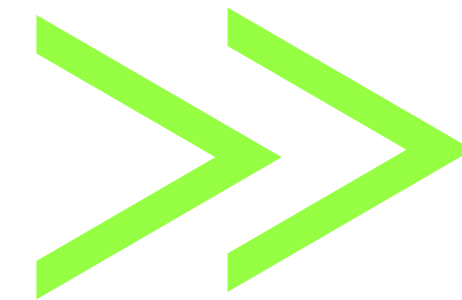
Авторизация



Нажимаем на кнопку «Войти»



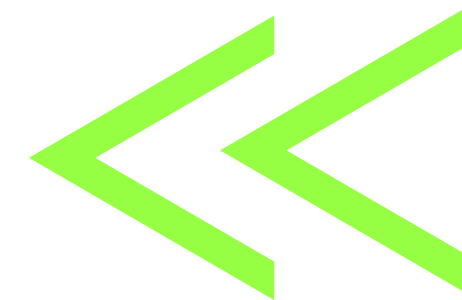
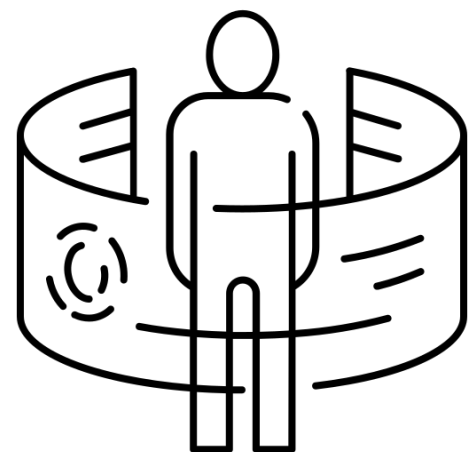
Запрос к нативной части



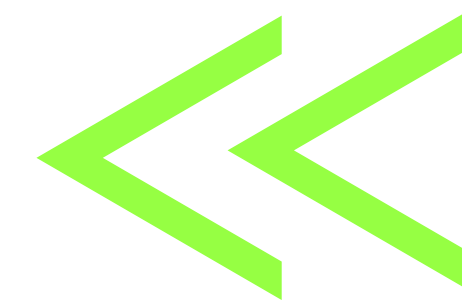
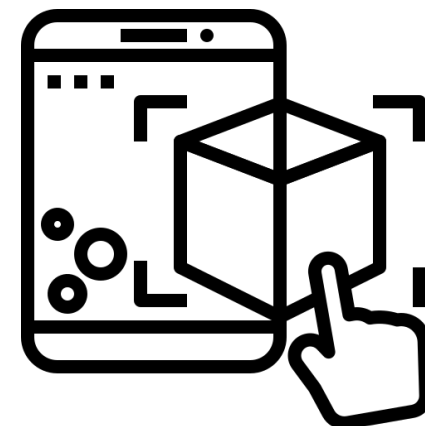
Натив обращается к сервису авторизации



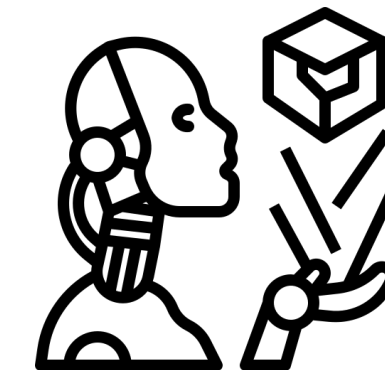
Авторизован



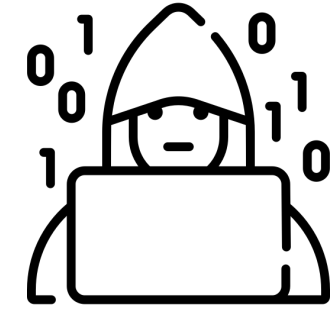
Передача токена в веб



Натив получает токен



Метод авторизации через сторонний сервис



commands.js

```
function inputMobileCode(response) {  
  const _el = document.createElement('html')  
  _el.innerHTML = response.body  
  const loginForm = _el.getElementsByTagName('input')  
  return cy.request({  
    method: 'POST',  
    headers: {},  
    url: baseUrl,  
    body: {  
      username: number,  
      code: code,  
    },  
  })  
}
```

Метод авторизации через сторонний сервис



commands.js

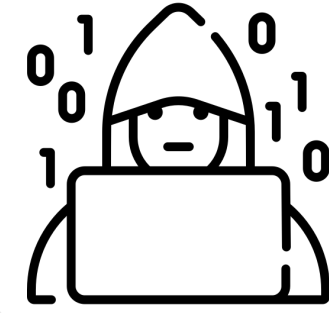
```
function inputMobileCode(response) {  
  const _el = document.createElement('html')  
  _el.innerHTML = response.body  
  const loginForm = _el.getElementsByTagName('input')  
  return cy.request({  
    method: 'POST',  
    headers: {},  
    url: baseUrl,  
    body: {  
      username: number,  
      code: code,  
    },  
  })  
}
```



commands.js

```
function getToken(response) {  
  const _el = document.createElement('html')  
  _el.innerHTML = response.body  
  const regex = /code=(.*)/gm  
  let code = regex.exec(response.redirects[0])[1]  
  return cy.request({  
    form: true,  
    method: 'POST',  
    url: `https://${baseUrl}/token`,  
    body: {  
      grant_type: 'authorization_code',  
      redirect_uri: 'https://pytorochka.5ka.ru/',  
      code,  
    },  
  })  
}
```

Получение токена



commands.js

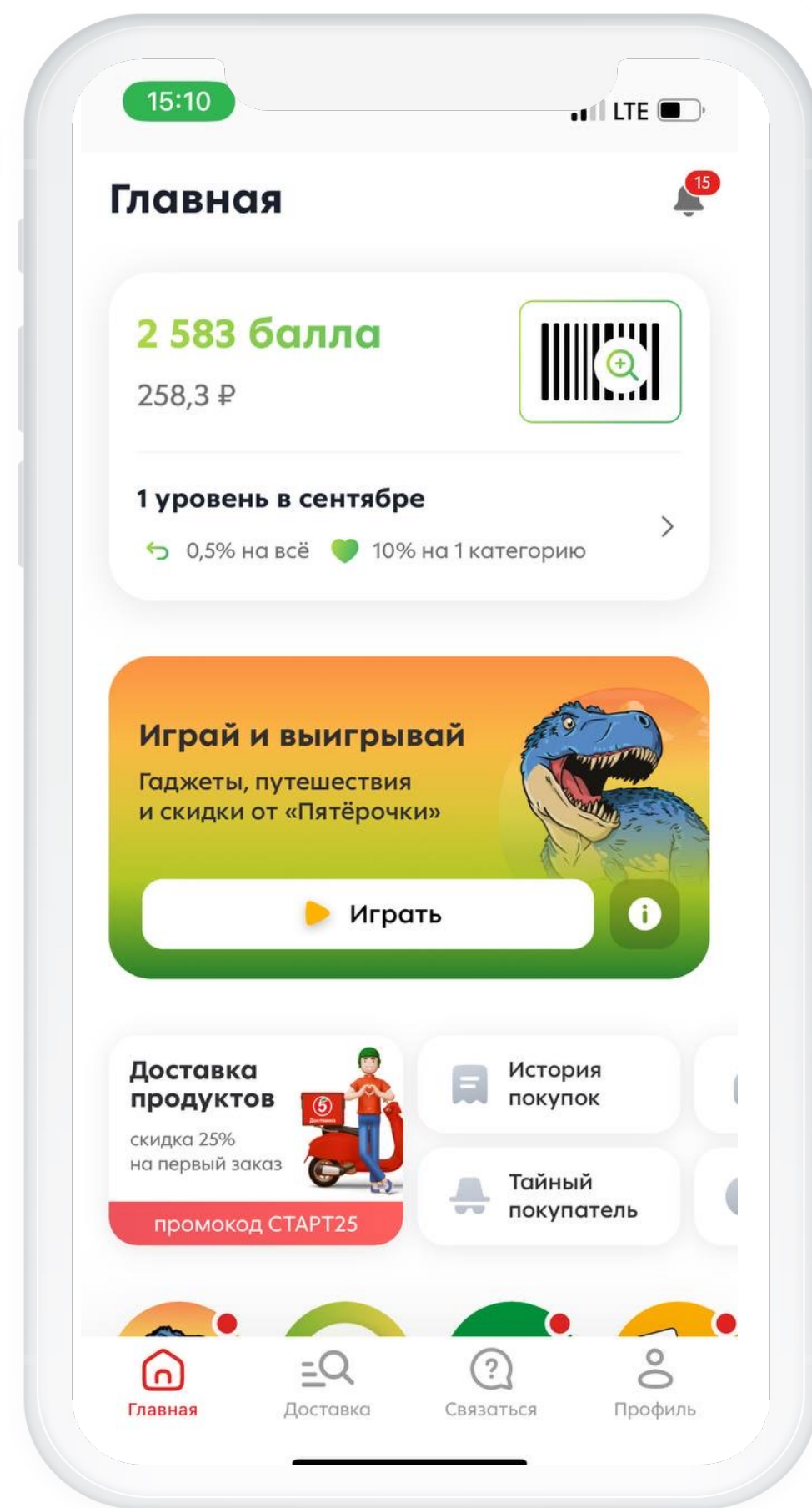
```
cy.request(loginPageRequest)
  .then(inputMobileNumber)
  .then(inputMobileCode)
  .then(getToken)
  .then((response) => {
    cy.setCookie('auth-token', response.body['token'])
  })
...

```

Получение токенов через API
для тестовых сред

- ✓ Токен получаем через API
- ✓ Складываем его в cookie, откуда его заберет web-приложение

Токен хранится в cookies



get →

Cookies

token: 1234567890

← 1234567890

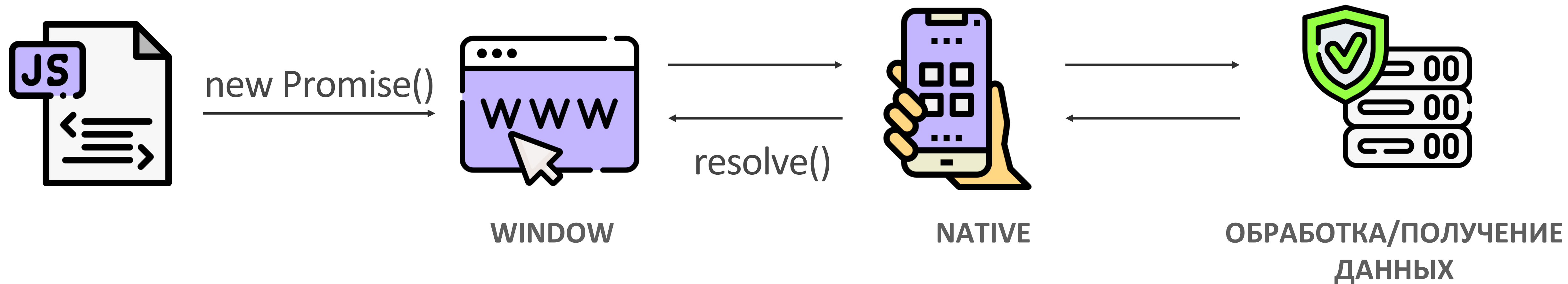
Как тестировать остальной функционал?

- ✓ Зависящий от операционной системы (iOS, Android)
- ✓ Зависящий от версии приложения
- ✓ Зависящий от нативных функций



Как работает JS API?

Все общается через promise и объект window



Настройки Cypress для автоматизации МП «Пятёрочка»



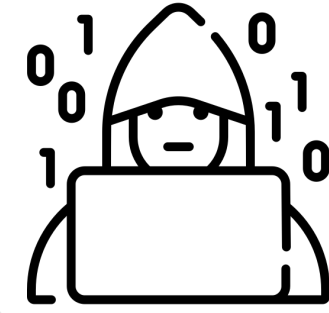
commands.js

```
cy.visit(path, {  
  onBeforeLoad: win => {  
    Object.defineProperty(win.navigator, 'userAgent', {  
      value: `Mozilla/5.0 (iPhone; CPU iPhone OS  
${osVersion} like Mac OS X) AppleWebKit/605.1.15` +  
      `(KHTML, like Gecko) Version/15.0 Mobile/15E148  
app-version/${appVersion} Safari/605.1.15`  
    })  
  })  
})
```

Эмулируем переход на страницу
мобильным приложением iOS

- ✓ Перед загрузкой страницы
подменяем user agent
- ✓ Указываем версию ОС
- ✓ Указываем версию приложения

Настройки Cypress для автоматизации МП «Пятёрочка»



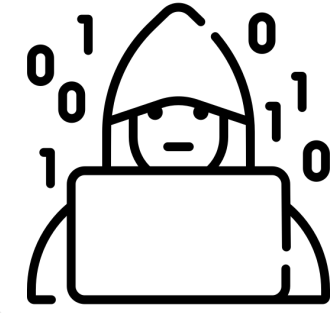
commands.js

```
cy.visit(path, {  
  onBeforeLoad: win => {  
    Object.defineProperty(win.navigator, 'userAgent', {  
      value: `Mozilla/5.0 (Linux; Android ${osVersion})  
AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0` +  
      `Chrome/88.0.4324.93 five-  
browser/${appVersion} Mobile Safari/537.36`  
    })  
  })  
})
```

Эмулируем переход на страницу
мобильным приложением Android

- ✓ Перед загрузкой страницы
подменяем user agent
- ✓ Указываем версию ОС
- ✓ Указываем версию приложения

Настройки Cypress для автоматизации МП «Пятёрочка»



commands.js

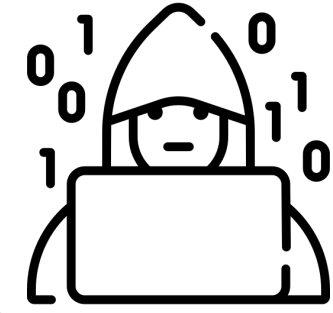
```
window.webkit = {  
  messageHandlers: {  
    el5: {  
      postMessage: () => {  
        setTimeout(() => {  
          win.__EL5_EVENT__(  
            JSON.stringify({name: 'ACCESS_TOKEN', payload:  
token?.value}))  
          }, 500)  
        }  
      }  
    }  
  }  
}
```

...

Мокаем JS API для iOS

- ✓ В интерфейс **webkit** для iOS добавляем свою функцию
- ✓ Подкидываем в нее токен

Настройки Cypress для автоматизации МП «Пятёрочка»



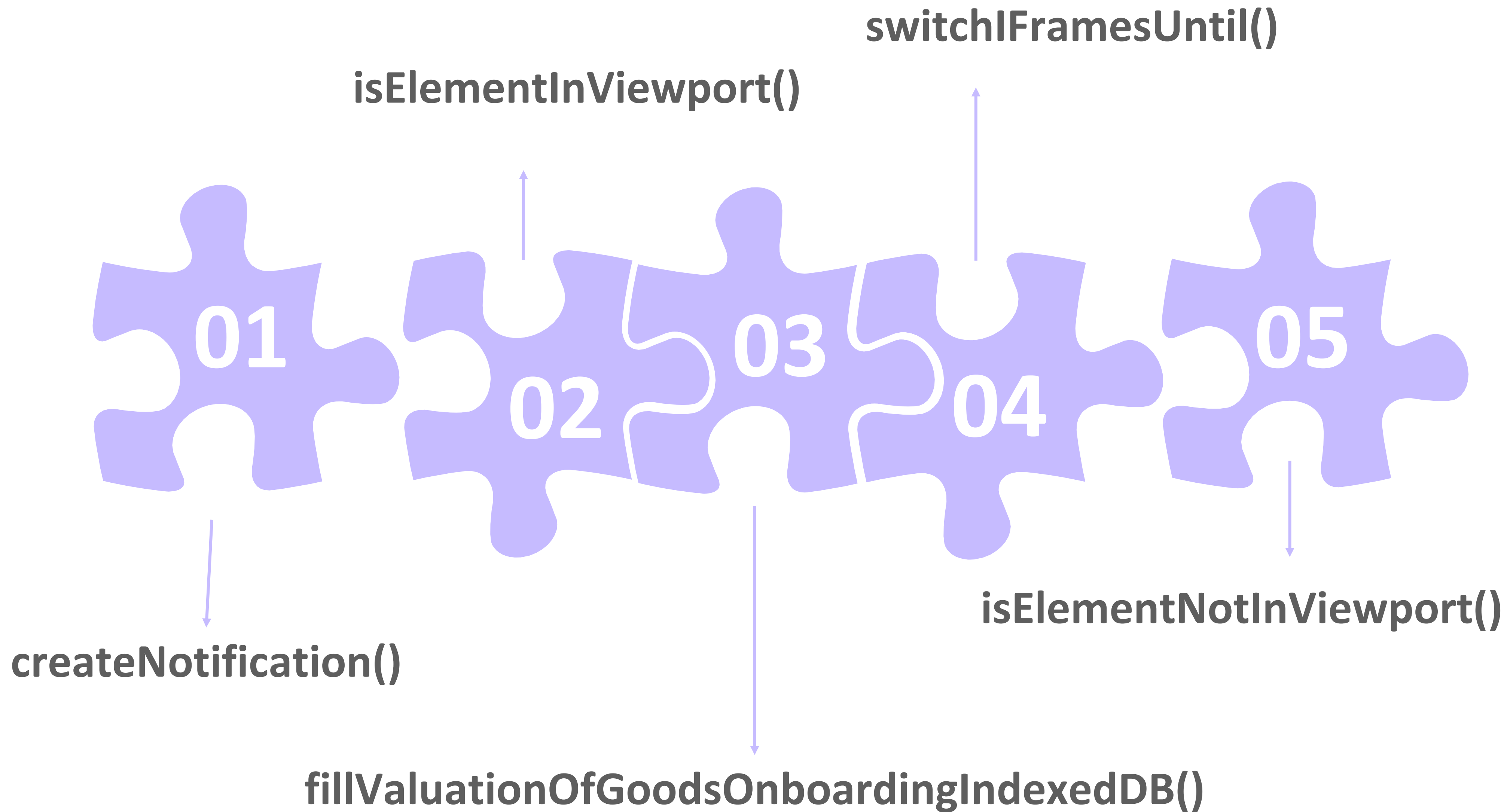
commands.js

```
window.AndroidInterface = {  
  getAccessToken: () => {  
    setTimeout(() => {  
      win.__EL5_TRIGGER_EVENT__(  
        JSON.stringify({name: 'ACCESS_TOKEN', payload:  
token?.value}))  
      }, 500)  
    }  
  }  
}...
```

Мокаем JS API для Android

- ✓ В интерфейс **AndroidInterface** для Android добавляем свою функцию
- ✓ Подкидываем в нее токен


Функции для функциональных тестировщиков



Выбрать 1 вариант или сложить пазл?



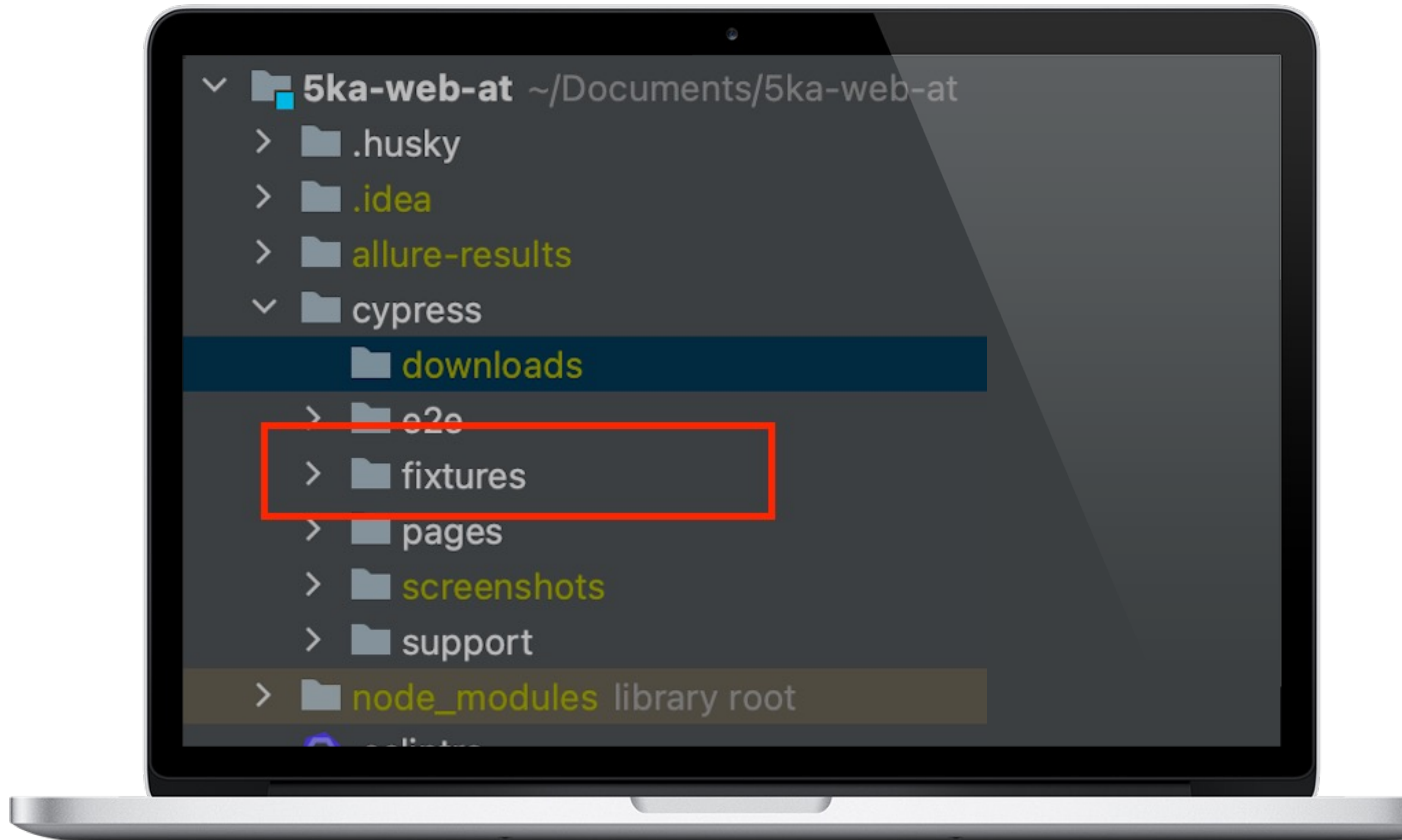
Пример функции и ее использования

 Использование
тестировщиком в тесте

 `cy.fillValuationOfGoodsOnboardingIndexedDB(true, false)`

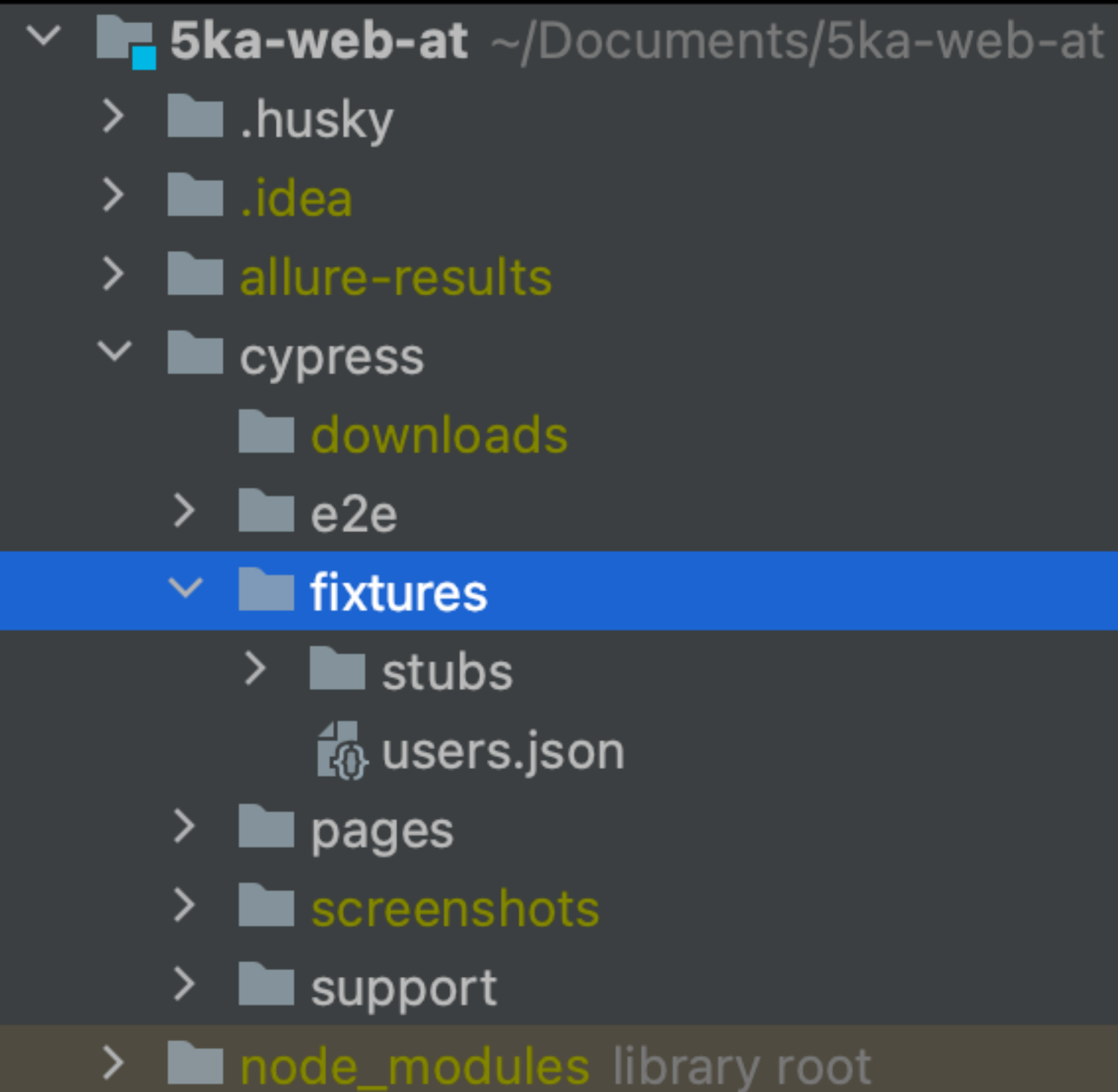
```
Cypress.Commands.add('fillValuationOfGoodsOnboardingIndexedDB',
(productHistoryTabs, valuationOfGoods) =>
new Cypress.Promise(async (resolve, reject) => {
  const request = indexedDB.open('fiveAppDatabase')
  request.onerror = function() {
    reject('Ошибка подключения к fiveAppDatabase')
  }
  request.onsuccess = function(event) {
    if ([...event.target.result.objectStoreNames].includes('onboardings')) {
      const db = event.target.result
      const transaction = db.transaction(['onboardings'], 'readwrite')
      const objectStore = transaction.objectStore('onboardings')
      const updatedDate = +new Date()
      const putRequest = objectStore.put({
        scope: 'valuation-of-goods',
        tours: {
          productHistoryTabs: productHistoryTabs,
          valuationOfGoods: valuationOfGoods
        },
        updatedDate: updatedDate
      })
      putRequest.onsuccess = function() {
        resolve()
      }
      putRequest.onerror = function() {
        reject('Ошибка записи в objectStore "onboardings"')
      }
    } else {
      resolve()
    }
  }
})
```

Структура проекта



Структура проекта

Fixtures - json файлы для
моков/стабов



```
5ka-web-at ~/Documents/5ka-web-at
├── .husky
├── .idea
├── allure-results
├── cypress
│   ├── downloads
│   ├── e2e
│   └── fixtures
│       ├── stubs
│       └── users.json
│   ├── pages
│   ├── screenshots
│   └── support
└── node_modules library root
```

Моки



Из файла

Спец.су.js

```
су.intercept('POST', '**/api/cardsService/card',  
{ fixture: 'stubs/emptyCard.json' })
```

Расположение в проекте



Моки

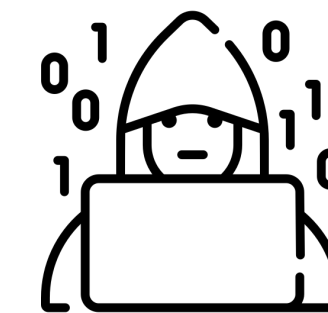


Из файла

Спец.су.js

```
cy.intercept('POST', '**/api/cardsService/card',
{ fixture: 'stubs/emptyCard.json' })
```

Расположение в проекте



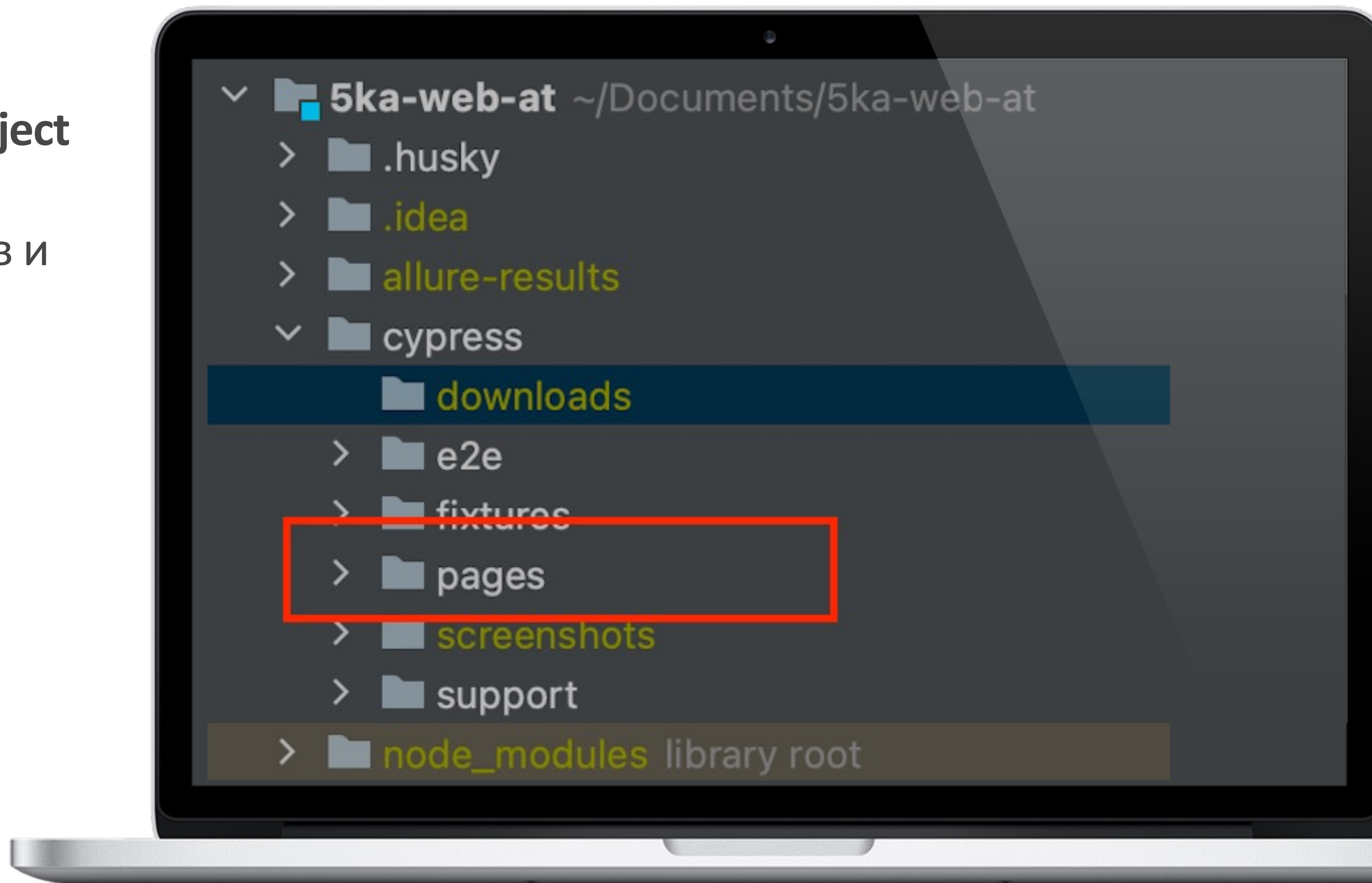
На лету

Спец.су.js

```
cy.intercept('GET',
 '**/api/balanceOperations/v3/balance/**',
 (req) => {
   req.reply((res) => {
     res.body.balance = 1234.56
     res.send(res.body)
   })
 })
```

Структура проекта

На проекте мы используем **Page Object Model** для хранения констант, селекторов и функций



Структура page object model

- ✓ Для начинающих тестировщиков достаточно вынести в **pageObject** все элементы страницы

```
class MainPage {
  navigate = () => cy.visit('/')

  buttonShowBarcode = () => cy.get('[data-qa="loyalty-cards-widget"]',
    {timeout:20000})
  levelDescription = () => cy.get('[data-qa="level-description"]')
  levelDescriptionArrow = () => cy.get('[data-qa="level-description_back-icon"]',
    {timeout: 10000})
  loyaltyCardDetails = () => cy.get('[data-qa="loyalty-card_details"]')
  stockCarousel = () => cy.get('[data-qa="stock-carousel"] > div > div')
  bigBanner = () => cy.get('[class*="Banner-module__btn"]')
  mysteryShopper = () => cy.get('[data-qa="slider-block_mystery-shopper"]')
  mainBanner = {
    text: () => cy.contains('скидка на первый заказ'),
    promo: () => cy.contains('промокод'),
    link: () => cy.get('[href*="/app/"]'),
    image: () => cy.get('[src*=".png"]')
  }
  bannerImg = () => cy.get('div[class*=Cards-Stock-Stock-module__img]')
  levelAlertBanner = () => cy.get('[data-qa="level-widget_alert-banner"]')
}
export const mainPage = new MainPage()
```

Структура page object model

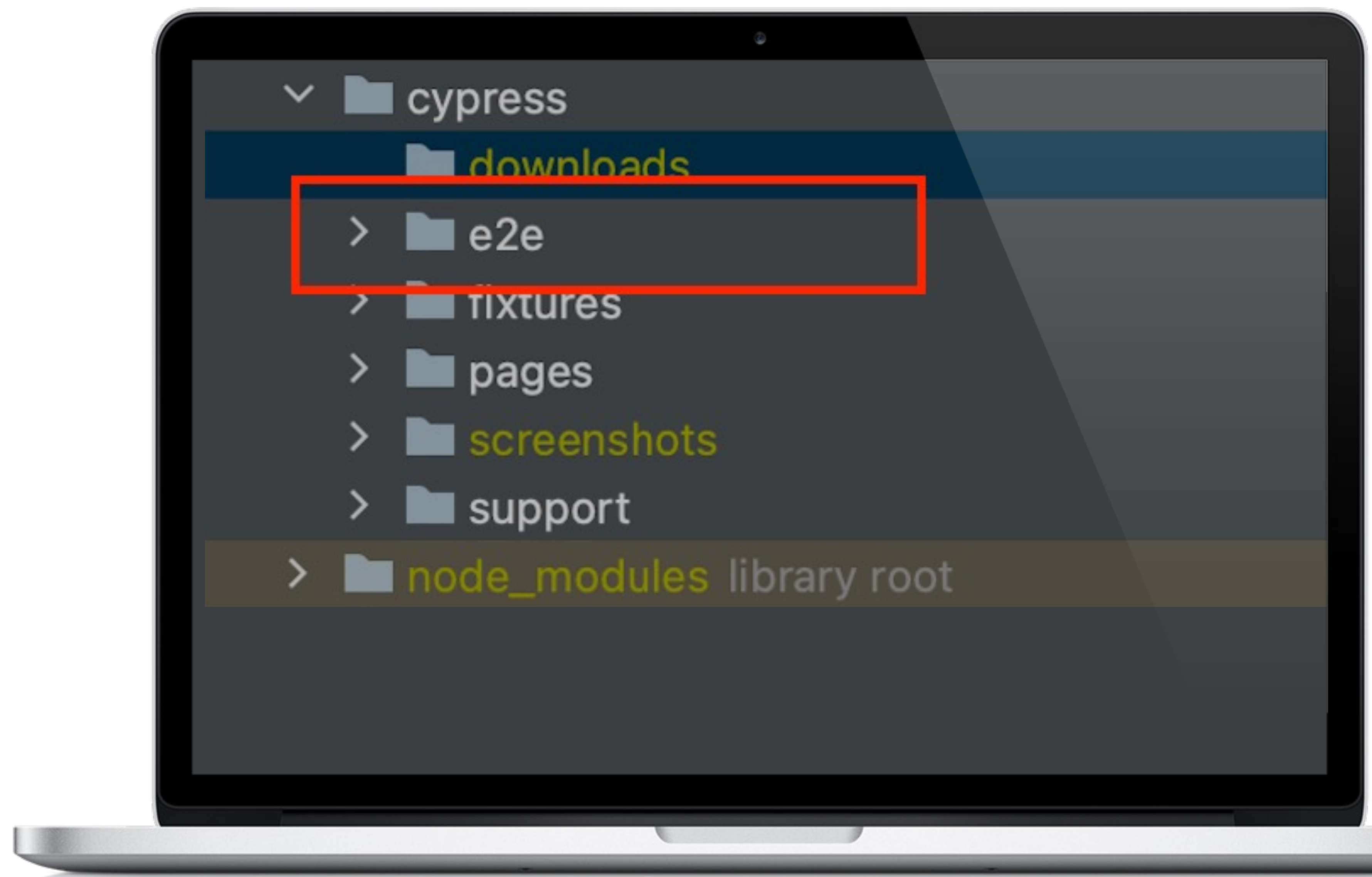
- ✓ А также функции, которые можно вызвать на текущей странице

```
class MyCardsPage {
  navigate = () => cy.visit('/cards')

  checkAnotherCardsPriority = () => {
    const cardNumbers = []
    this.anotherCardsNumbers()
      .each((el) => {
        cardNumbers.push(el.text())
      })
      .then(() => {
        cardNumbers.forEach((number, i, numbers) => {
          if (i > 0) assert(number > numbers[i - 1], 'Последовательность отображения')
        })
      })
  }
}

export const MyCardsPage = new MyCardsPage()
```

Структура проекта



Пишем тест



```
import {Page} from '../..../mainPage '  
import users from '../..../fixtures/u.json'  
import {bannerPage} from '../..../pages/Homepage/bannersPage'  
  
describe('', () => {  
  
  before(() =>{  
    cy.allure()  
      .feature('Главная') })  
  
    beforeEach(() => {  
      cy.allure().link('Y', 'X')  
    })  
  
    it('Название', () => { })  
  
  })  
})
```

Для начинающих автотестировщиков

- ✓ Импорты страниц и фикстур
- ✓ Заготовки для аллюр плагина
в соответствии с ТМС

Пишем тест



```
import {mainPage} from '.././././mainPage'  
import users from '.././././fixtures/users.json'  
import {bannerPage} from '.././././pages/Homepage/bannersPage'
```

```
describe('', () => {
```

```
  it('Test', () => {
```

```
    cy.allure().step('Открыть раздел "Главная"')
```

```
    mainPage.navigate()
```

```
    cy.allure().step('Нажать на баннер')
```

```
    mainPage.banner.click()
```

```
    cy.allure().step('Проверить, что заголовок отображается')
```

```
    bannersPage.title.checkVisibility()
```

```
  })
```

```
})
```

Тестировщик

- ✓ Обращается к элементам страницы по ее названию
- ✓ На каждом из шагов теста понимает, на какой странице сейчас находится, поэтому с легкостью может редактировать существующие сценарии

Пример запуска и отладки



Напишем простой тест

```
it( title: 'Проверка перехода по клику на баннер', config: () => {  
  cy.allure().step( name: 'Перейти на главную')  
  MainPage.navigate()  
  
  cy.allure().step( name: 'Нажать на большой баннер')  
  MainPage.bigBanner().click()  
  
  cy.allure().step( name: 'Проверить, что тайтл отображается и имеет текст "Посуда за наклейки"')  
  BigBannersPage.title().should( chainer: 'be.visible').and( chainer: 'have.text', value: 'Посуда за наклейки')  
})
```

Запуск в режиме отладки

test.cy.js 926ms

▼ Главная

- ✓ Проверка перехода по клику на баннер
- ▼ TEST BODY

```
visit /  
2 get [data-content-name="main_page-main_banner"]  
3 -click  
  (new url) https://stage.pyaterochka.ru/liberhaus-info  
4 get [data-qa="common_header"]  
5 -assert expected <header.src-layouts-DefaultLayout-  
DefaultLayout-module_headerWrapper--rqjju.css-12595yo> to be  
  visible  
6 -assert expected <header.src-layouts-DefaultLayout-  
DefaultLayout-module_headerWrapper--rqjju.css-12595yo> to have  
  text Посуда за наклейки
```

Electron 106 428x926 (79%)

Главная

Доступно обновление приложения [Обновить](#)

Войдите,
чтобы получать персональные предложения и баллы за покупки

[Войти](#)

Доставка продуктов
скидка 25% на первый заказ
промокод СТАРТ25


[История покупок](#)

[Частые вопросы](#)

Реклама

Просто накопить — выгодно купить!
Скидка до 55% на посуду для приготовления Liberhaus

[Узнать больше](#)



Визуализация действий с элементом

test.cy.js 926ms

▼ Главная

- ✓ Проверка перехода по клику на баннер

▼ TEST BODY

```
1 visit /
2 get [data-content-name="main_page-main_banner"]
3 -click
  (new url) https://stage.pyaterochka.ru/liberhaus-info
4 get [data-qa="common_header"]
5 -assert expected <header.src-layouts-DefaultLayout-DefaultLayout-module__headerWrapper--rqjju.css-12595yo> to be visible
6 -assert expected <header.src-layouts-DefaultLayout-DefaultLayout-module__headerWrapper--rqjju.css-12595yo> to have text Посуда за наклейки
```

Electron 106 428x926 (79%)

Главная

Доступно обновление приложения [Обновить](#)

Войдите,
чтобы получать персональные предложения и баллы за покупки

[Войти](#)

Доставка продуктов
скидка 25% на первый заказ
промокод СТАРТ25

[История покупок](#)

[Частые вопросы](#)

Реклама

Просто накопить — выгодно купить!
Скидка до 55% на посуду для приготовления Liberhaus

[Узнать больше](#)

Визуализация действий с элементом

The screenshot displays the Cypress test runner interface. On the left, the test script in `test.cy.js` is shown with the following steps:

```
1 visit /
2 get [data-content-name="main_page-main_banner"]
3 -click
  (new url) https://stage.pyaterochka.ru/liberhaus-info
4 get [data-qa="common_header"]
5 -assert expected <header.src-layouts-DefaultLayout-DefaultLayout-module__headerWrapper--rqjju.css-12595yo> to be visible
   -assert expected <header.src-layouts-DefaultLayout-DefaultLayout-module__headerWrapper--rqjju.css-12595yo> to have text Посуда за наклейки
```

The right side of the image shows a visualized browser action on a website banner for "Посуда за наклейки" (Dishes for stickers). The banner features a cooking scene with a pan and a pot, a "До 55%" discount tag, and the text "Просто накопить — выгодно купить!" (Simply accumulate — it's profitable to buy!). The test runner highlights the element being interacted with.

At the bottom, the "Developer Tools" console is open, showing the following log entry:

```
Command: assert
Actual:   Посуда за наклейки
Expected: Посуда за наклейки
Subject:
  > jQuery.fn.init [header.src-layouts-DefaultLayout-DefaultLayout-module__headerWrapper--rqjju.css-12595yo]
Message: expected <header.src-layouts-DefaultLayout-DefaultLayout-module__headerWrapper--rqjju.css-12595yo> to have text Посуда за наклейки
```

В консоли все круто и подробно отображается!



Линтер

.eslintrc

- ✓ Важен при работе большого количества человек на проекте
- ✓ Начинающим очень легко поддерживать стиль и не допускать ошибок

```
{
  "env": {
    "es2021": true,
    "node": true
  },
  "extends": ["plugin:cypress/recommended", "plugin:mocha/recommended"],
    "plugins": ["cypress", "mocha"],
    "overrides": [],
    "parserOptions": {
      "ecmaVersion": "latest",
      "sourceType": "module"
    },

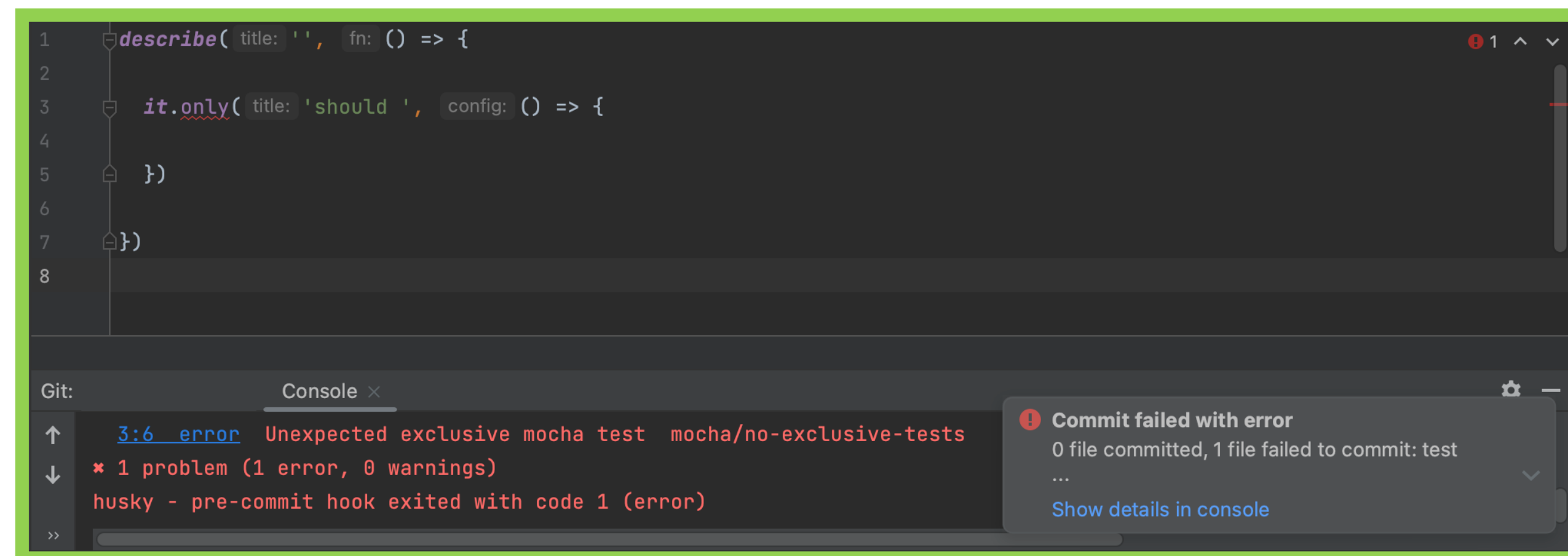
  "rules": {
    "newline-per-chained-call": "off",
    "no-unused-vars": "error",
    "indent": ["error", 2],
    "quotes": ["error", "single"],
    "semi": ["error", "never"],
    "comma-spacing": "error",
    "space-in-parens": "error",
    "max-len": ["error", { "code": 120, "comments": 120, "ignoreUrls": true }],
    "keyword-spacing": "error",
    "no-trailing-spaces": ["error", { "skipBlankLines": true }],
    ...
  }
}
```

Pre-commit hook



package.json

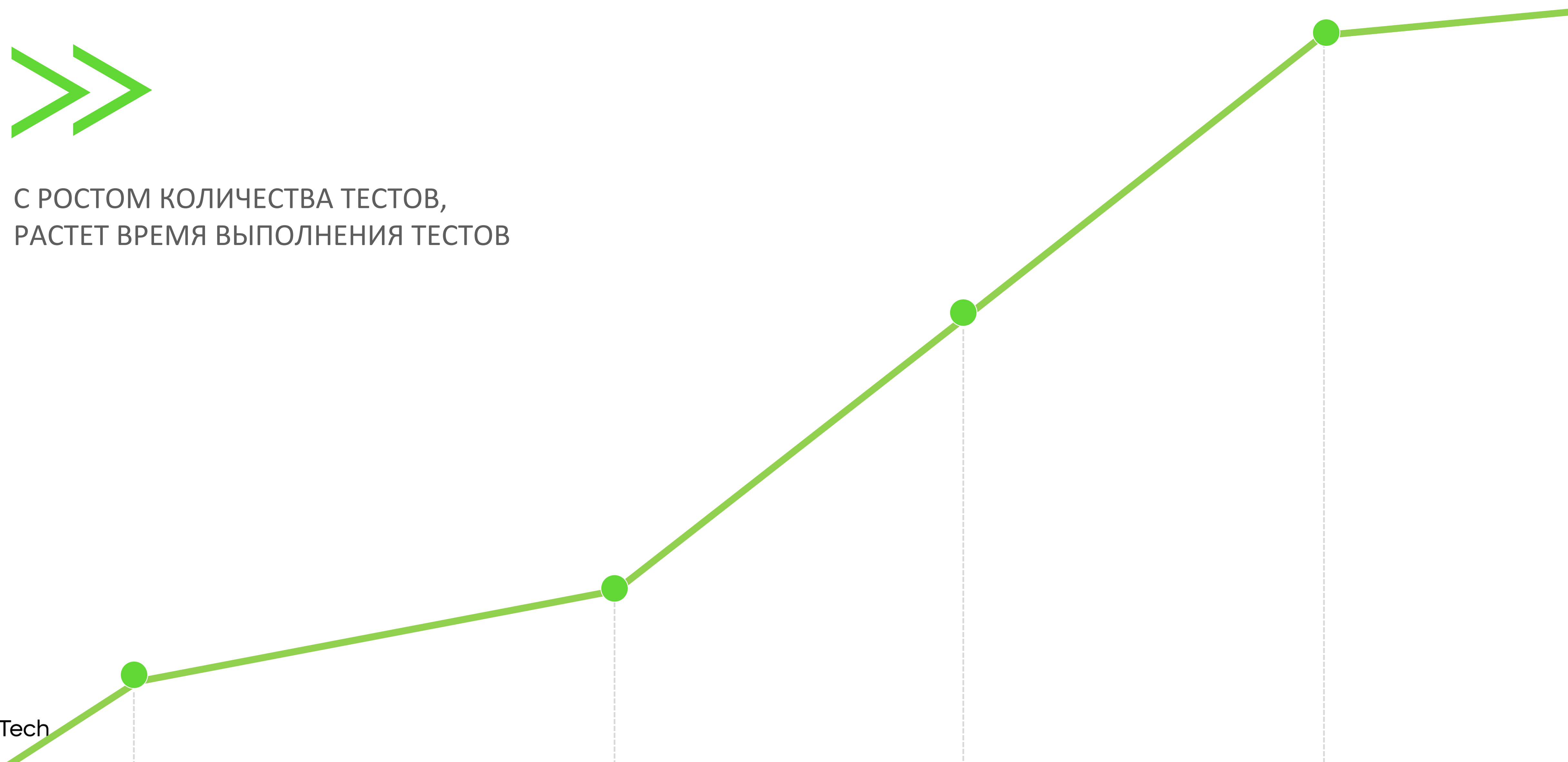
```
"scripts": {  
  "prepare": "husky install",  
},  
"lint-staged": {  
  "**.js": [  
    "node_modules/.bin/eslint --max-warnings=0",  
    "git add"  
  ]  
}
```



Рост вовлеченности и автоматизации



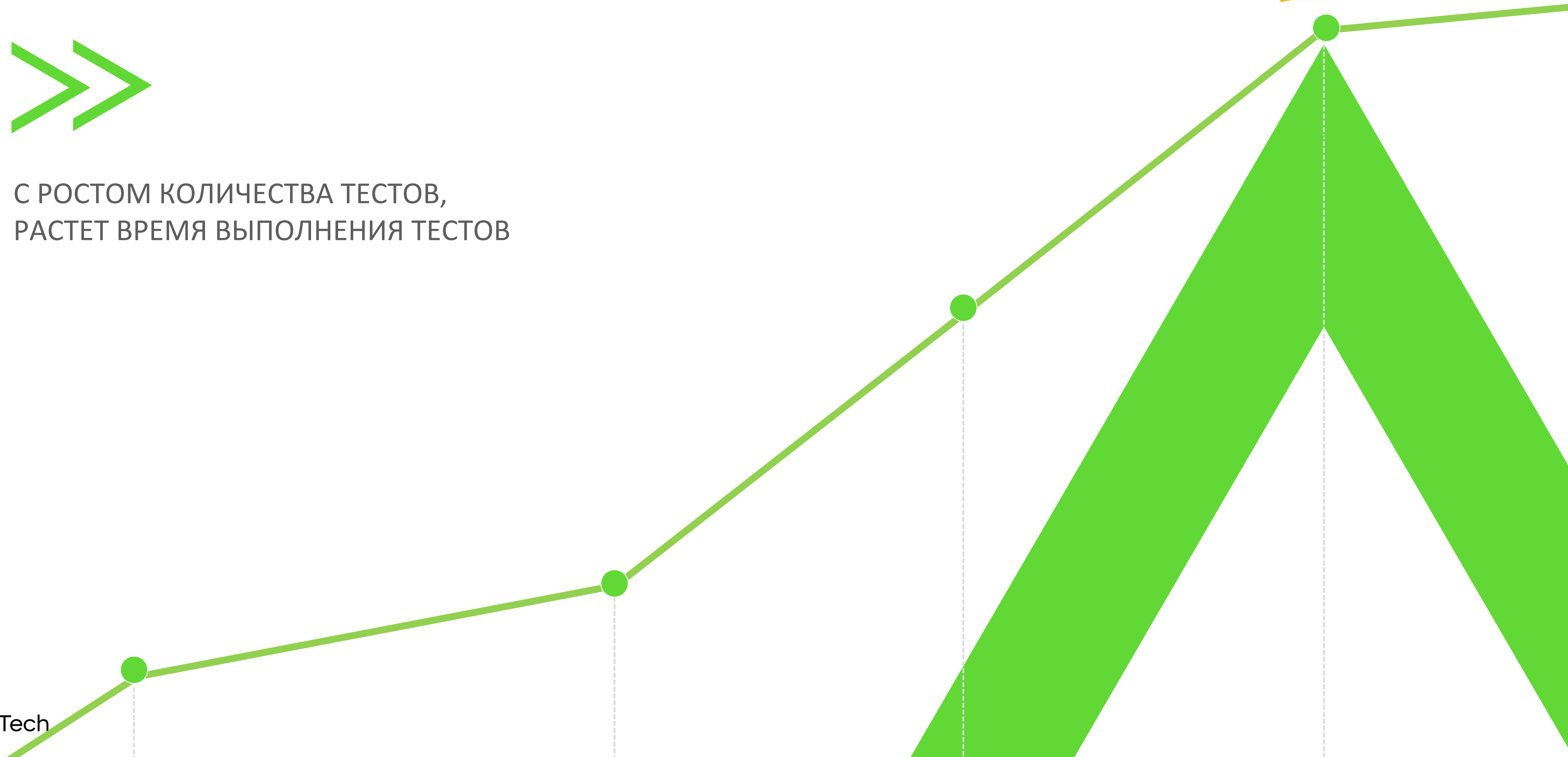
С РОСТОМ КОЛИЧЕСТВА ТЕСТОВ,
РАСТЕТ ВРЕМЯ ВЫПОЛНЕНИЯ ТЕСТОВ



Рост вовлеченности и автоматизации



С РОСТОМ КОЛИЧЕСТВА ТЕСТОВ,
РАСТЕТ ВРЕМЯ ВЫПОЛНЕНИЯ ТЕСТОВ



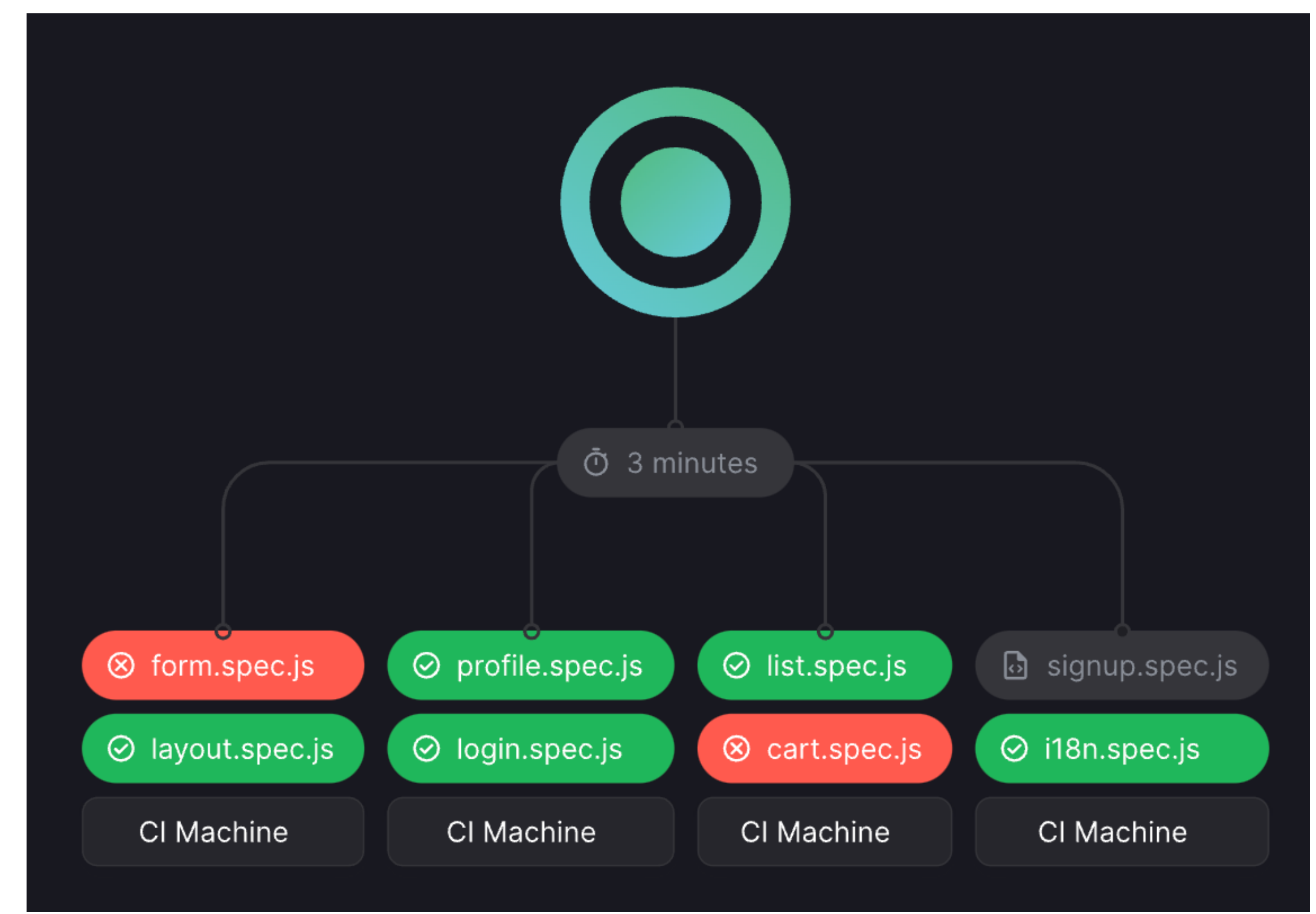
Параллелизация



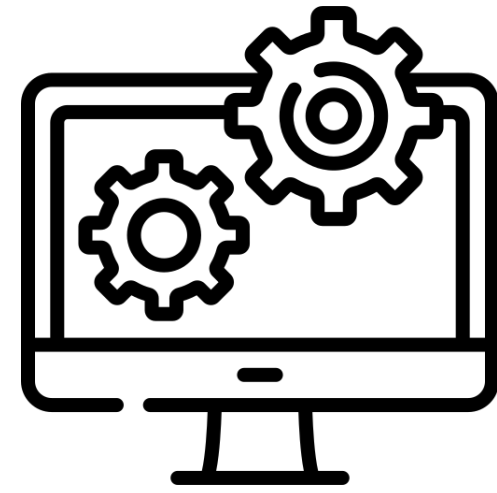
.gitlab-ci.yml

```

test-job:
  allow_failure: true
  stage: test
  parallel: 4
  script:
    - npm i
    - allurectl watch -- npx cy2 run --record --
      key 5WEB --parallel --ci-build-id
      $CI_PIPELINE_ID
  
```

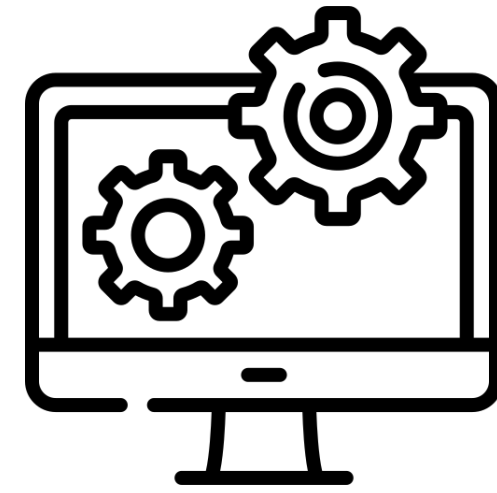


Sorry Cypress



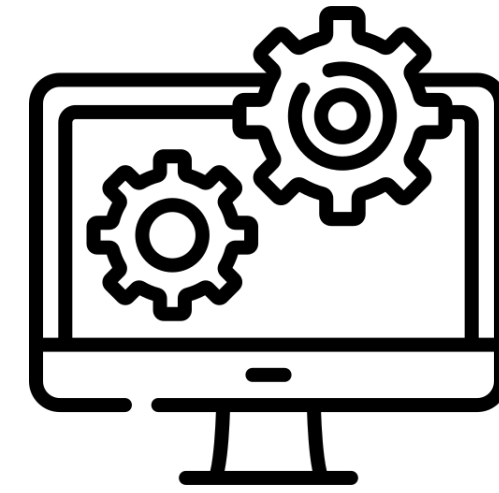
CI machine 1

form.spec.js
layout.spec.js
profile.spec.js
login.spec.js
list.spec.js
cart.spec.js
signup.spec.js
logout.spec.js



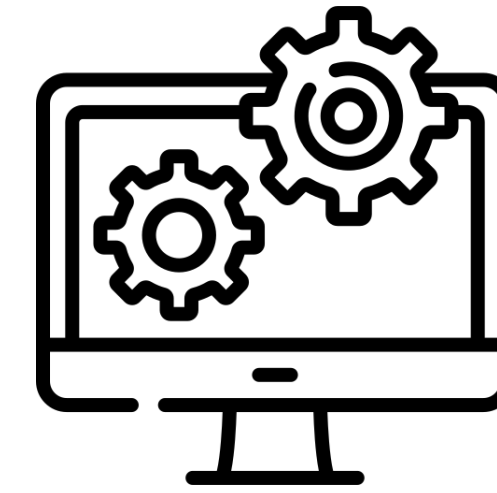
CI machine 2

form.spec.js
layout.spec.js
profile.spec.js
login.spec.js
list.spec.js
cart.spec.js
signup.spec.js
logout.spec.js



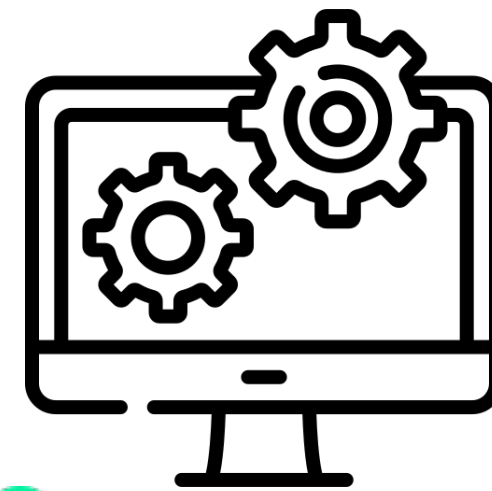
CI machine 3

form.spec.js
layout.spec.js
profile.spec.js
login.spec.js
list.spec.js
cart.spec.js
signup.spec.js
logout.spec.js

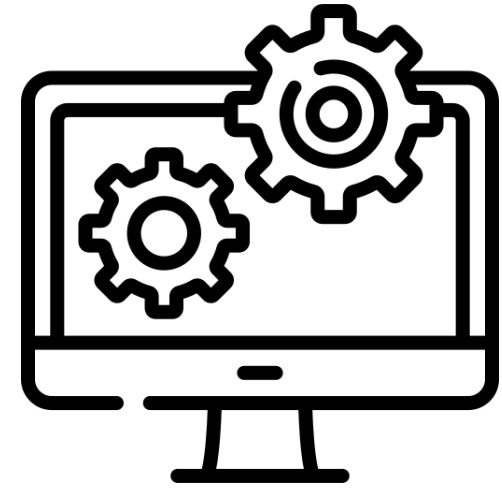


CI machine 4

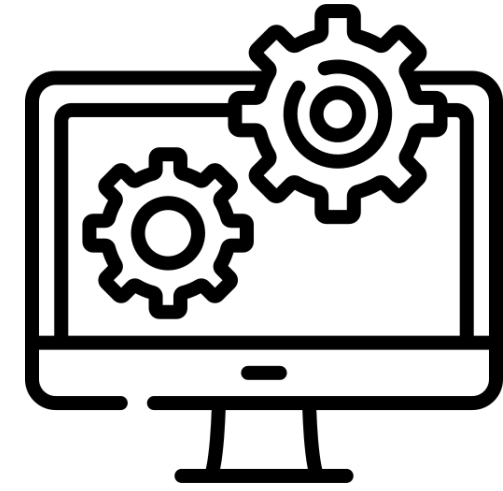
form.spec.js
layout.spec.js
profile.spec.js
login.spec.js
list.spec.js
cart.spec.js
signup.spec.js
logout.spec.js



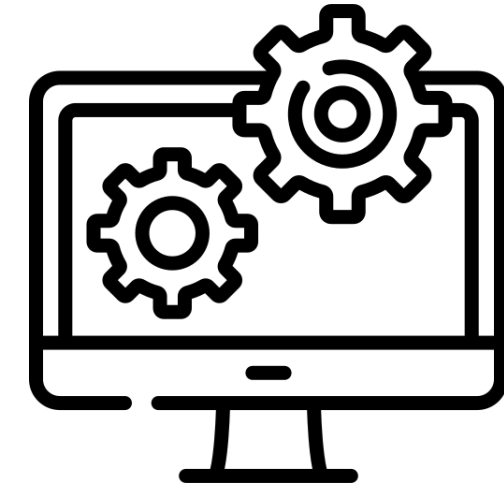
Sorry Cypress



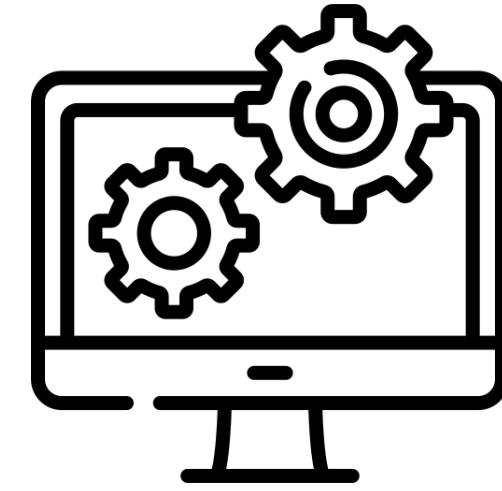
CI machine 1



CI machine 2

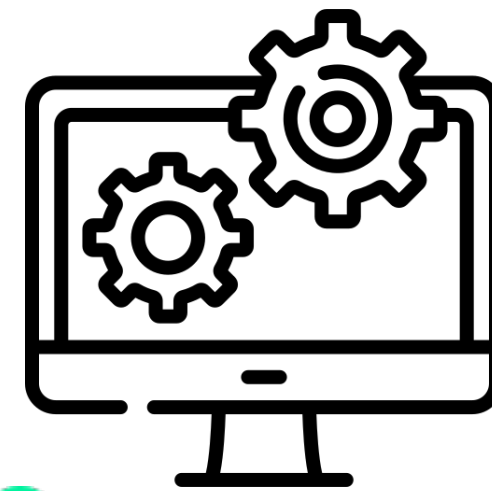
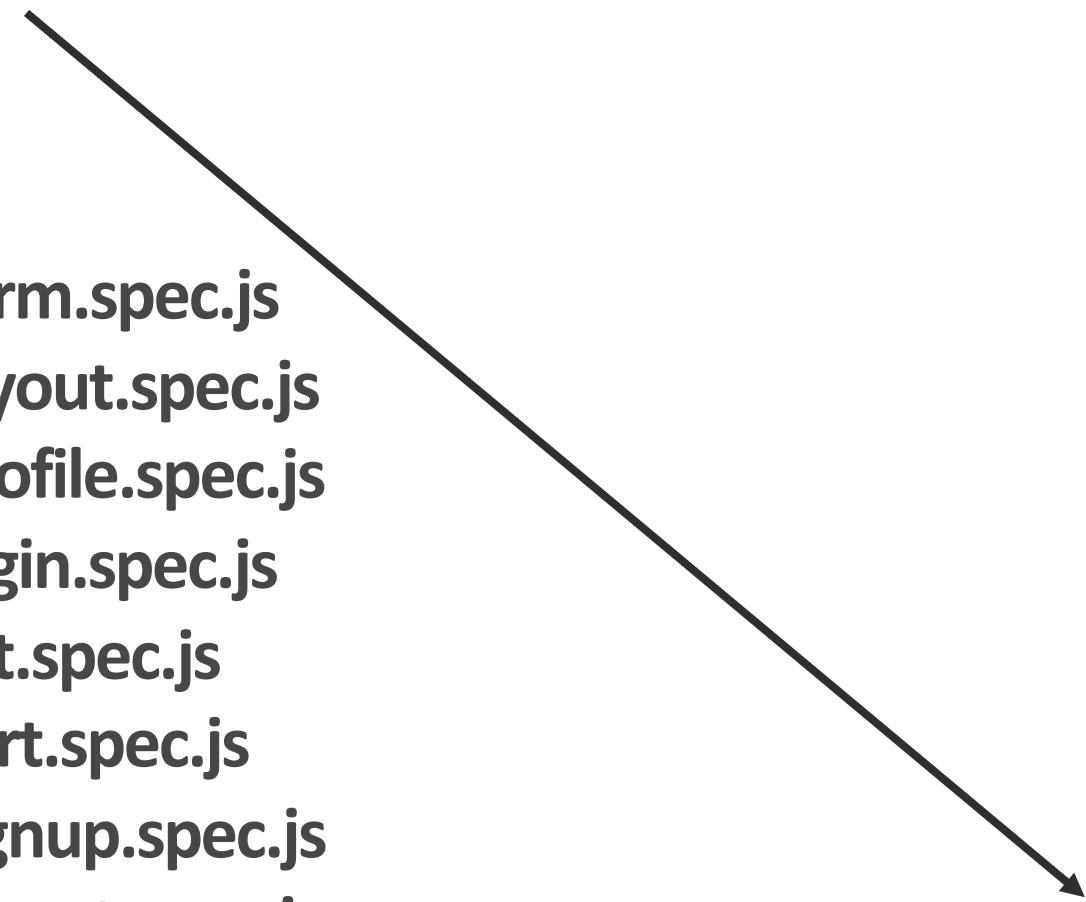


CI machine 3

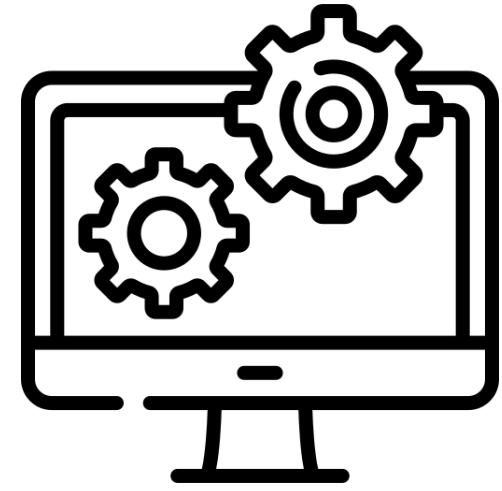


CI machine 4

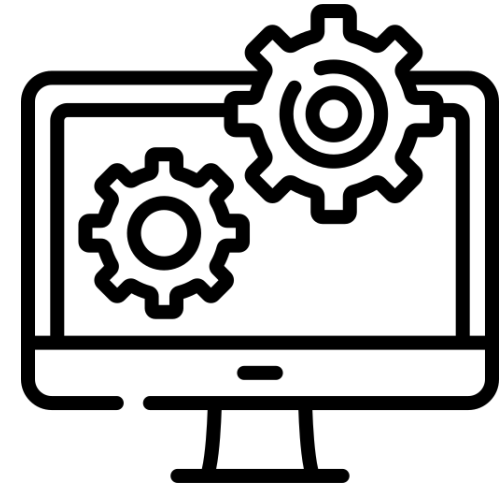
form.spec.js
layout.spec.js
profile.spec.js
login.spec.js
list.spec.js
cart.spec.js
signup.spec.js
logout.spec.js



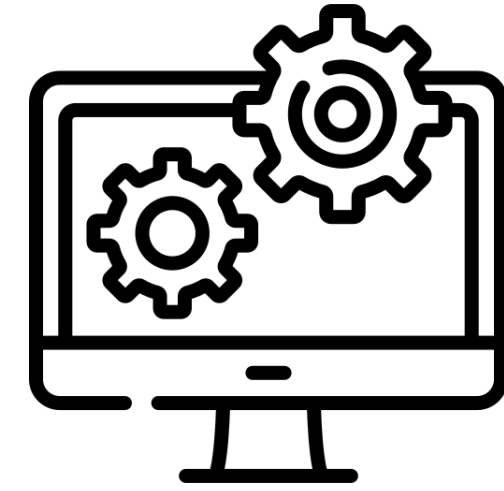
Sorry Cypress



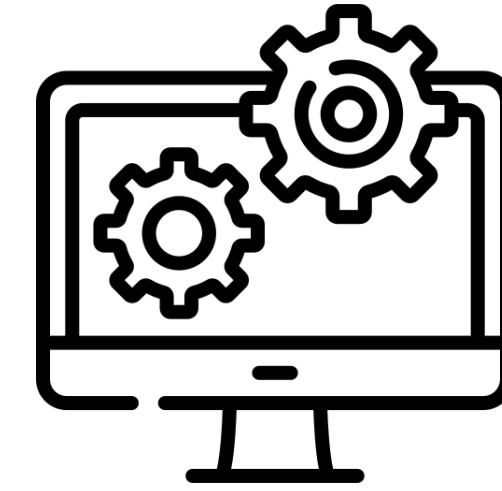
CI machine 1



CI machine 2

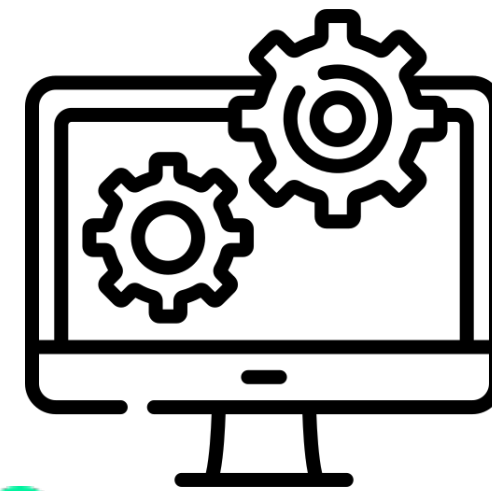


CI machine 3



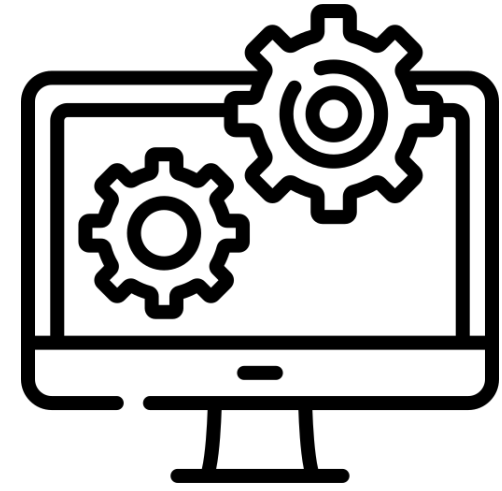
CI machine 4

form.spec.js

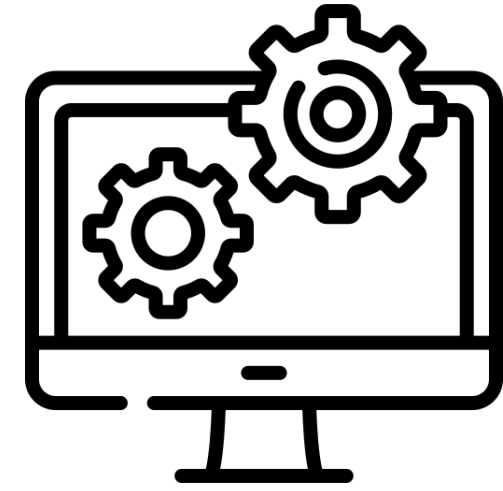


- form.spec.js
- layout.spec.js
- profile.spec.js
- login.spec.js
- list.spec.js
- cart.spec.js
- signup.spec.js
- logout.spec.js

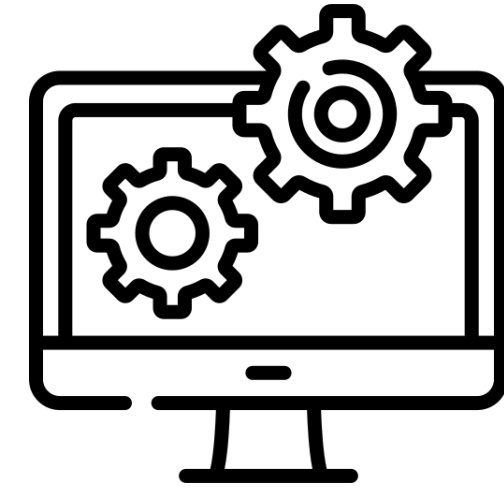
Sorry Cypress



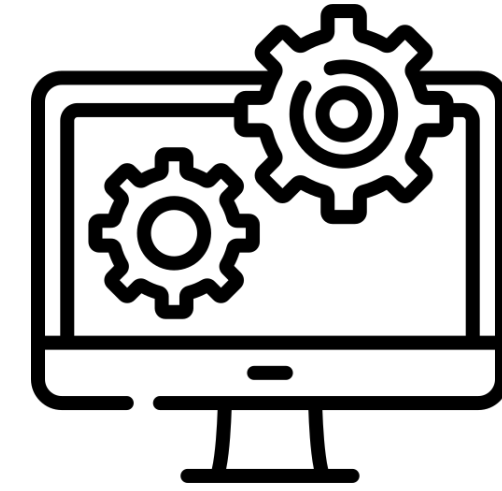
CI machine 1
form.spec.js



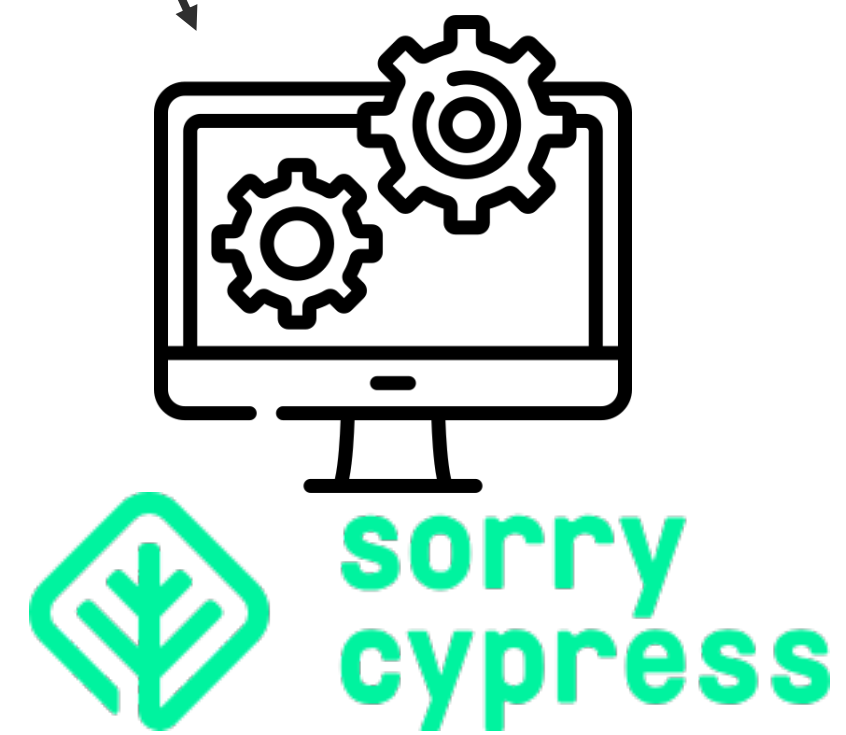
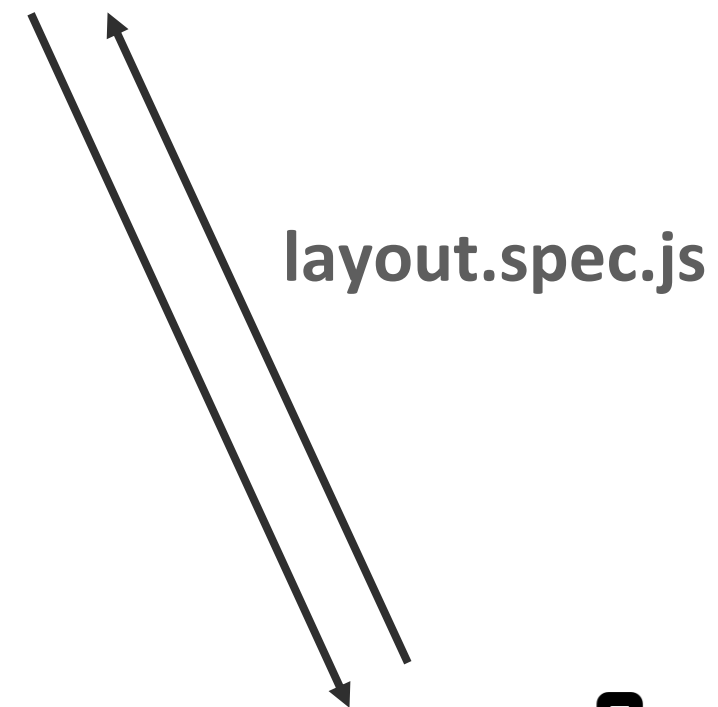
CI machine 2



CI machine 3

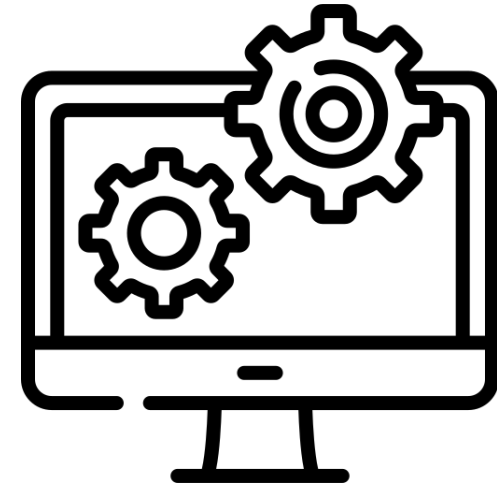


CI machine 4



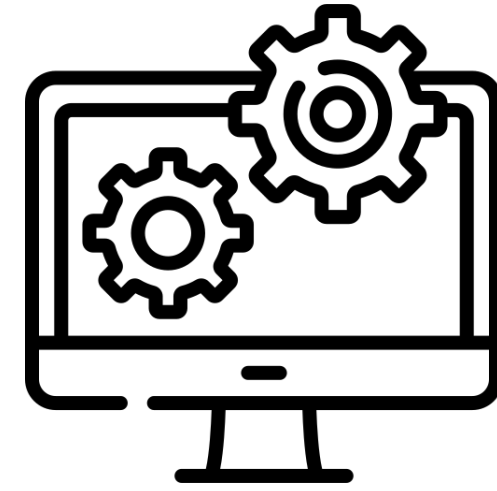
- form.spec.js
- layout.spec.js
- profile.spec.js
- login.spec.js
- list.spec.js
- cart.spec.js
- signup.spec.js
- logout.spec.js

Sorry Cypress



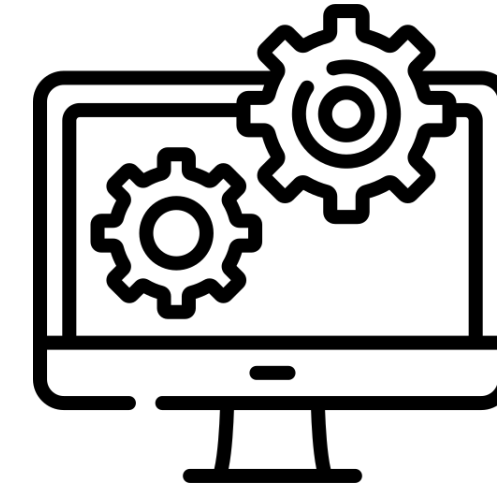
CI machine 1

form.spec.js
list.spec.js



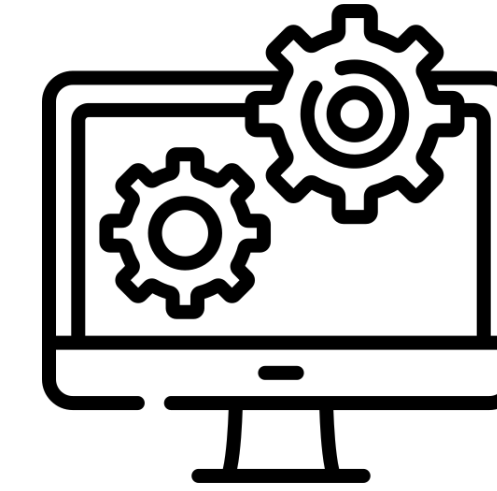
CI machine 2

layout.spec.js
cart.spec.js



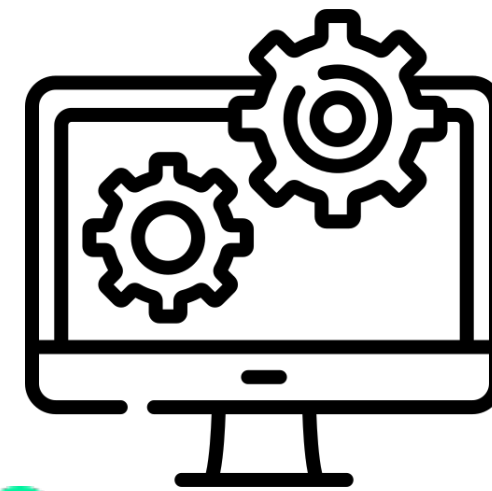
CI machine 3

profile.spec.js
signup.spec.js



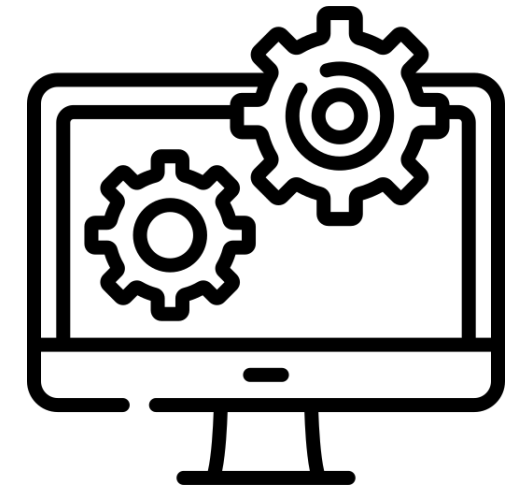
CI machine 4

login.spec.js
logout.spec.js



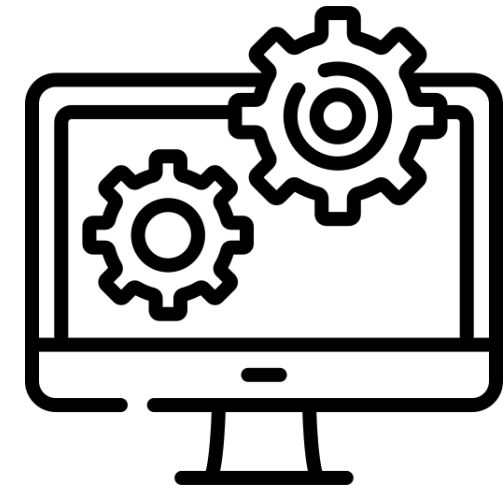
form.spec.js
layout.spec.js
profile.spec.js
login.spec.js
list.spec.js
cart.spec.js
signup.spec.js
logout.spec.js

Sorry Cypress



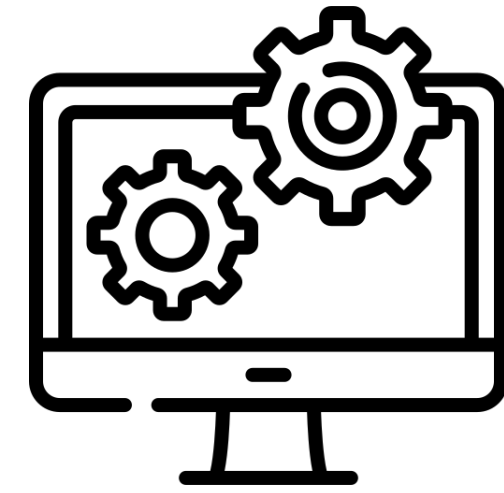
CI machine 1

form.spec.js
list.spec.js



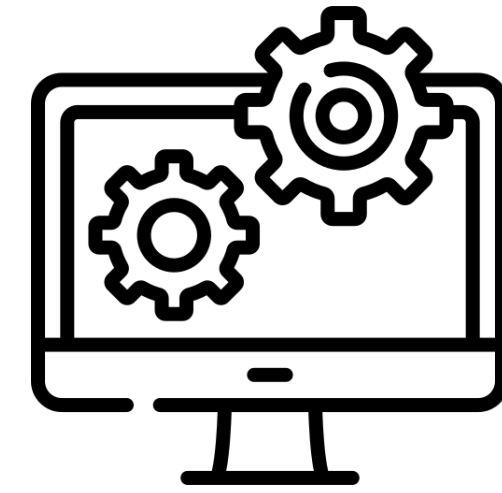
CI machine 2

layout.spec.js
cart.spec.js



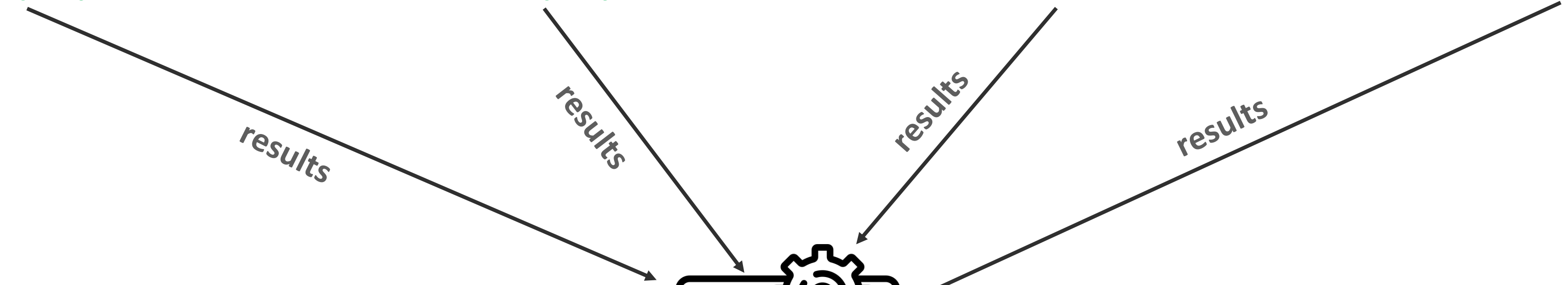
CI machine 3

profile.spec.js
signup.spec.js



CI machine 4

login.spec.js
logout.spec.js

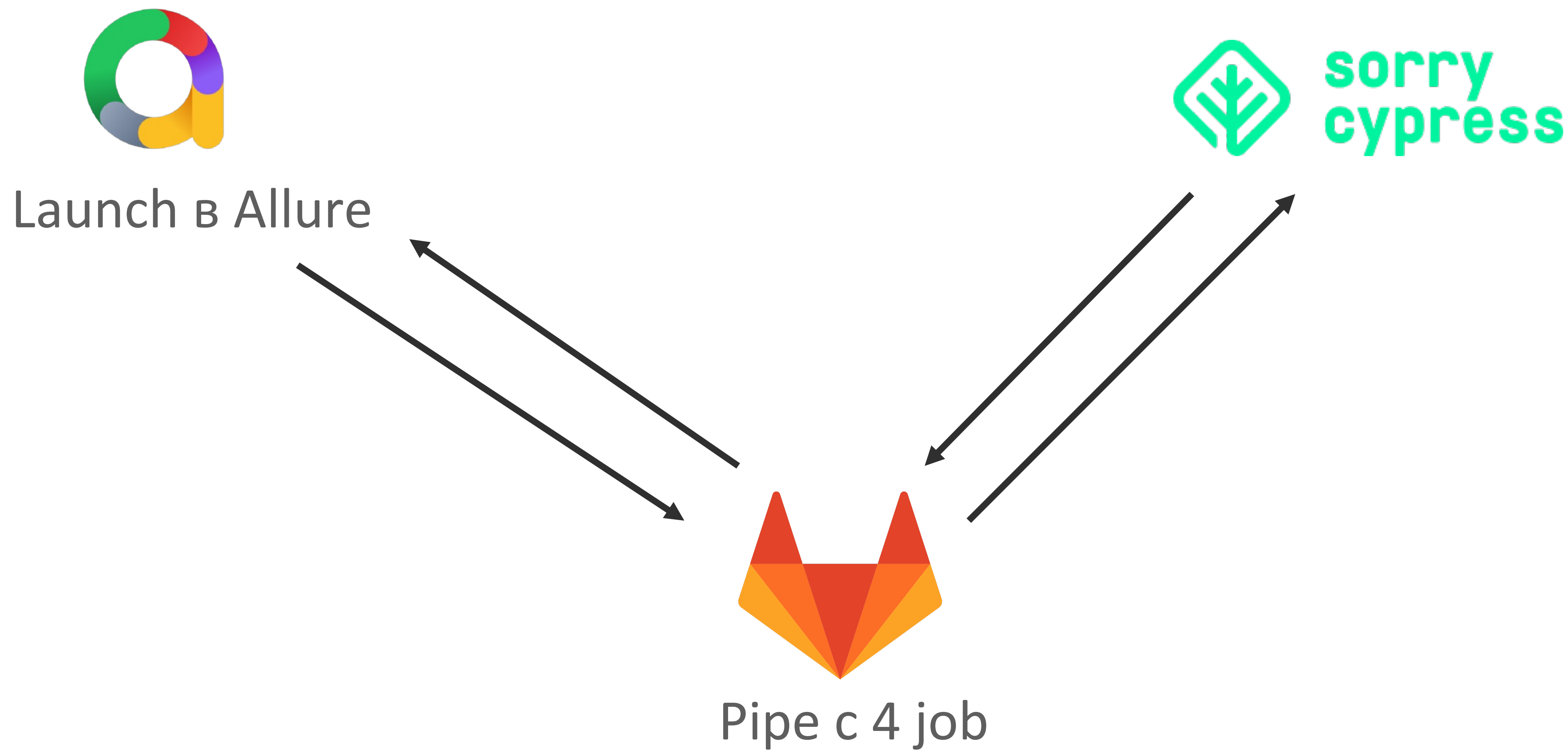


Результаты в Sorry Cypress

The screenshot displays the 'Latest runs' page for the '5kaWEB' project. The interface includes a sidebar with navigation options: Recent Builds, Projects, 5kaWEB, Latest runs, and Project settings. The main content area shows a list of build runs with the following details:

- Run 2783829:** Status: Failed (red dot). Date: Sep 27 2023, 13:04:37. Duration: 24m 15s. Progress: 52 / 52. Author: John Dow. Action: Merge branch 'LPAP-593111-carousel-fluffy' into 'master'. Test results: 384 passed, 358 failed, 20 skipped, 10 errored, 0 retried, 0 cancelled.
- Run 2781696:** Status: Failed (red dot). Date: Sep 26 2023, 19:08:19. Duration: 8m 32s. Progress: 51 / 51. Author: John Dow. Action: Merge branch 'SP-844066-NC_tests_from_visual_branch' into 'master'. Test results: 377 passed, 363 failed, 14 skipped, 10 errored, 0 retried, 0 cancelled.
- Run 2781529:** Status: Failed (red dot). Date: Sep 26 2023, 18:07:00. Duration: 15m 26s. Progress: 51 / 51. Author: John Dow. Action: Merge branch 'SP-844066-NC_tests_from_visual_branch' into 'master'. Test results: 377 passed, 344 failed, 33 skipped, 8 errored, 0 retried, 0 cancelled.
- Run 2776434:** Status: Failed (red dot). Date: Sep 25 2023, 14:25:29. Duration: 8m 44s. Progress: 51 / 51. Author: Jane Dow. Action: Merge branch 'lpap-593310-fix=test-fluffy' into 'master'. Test results: 375 passed, 358 failed, 17 skipped, 7 errored, 0 retried, 0 cancelled.
- Run 2775929:** Status: Failed (red dot). Date: Sep 25 2023, 12:34:52. Duration: 30m. Progress: 51 / 51. Status: timed out. Author: Jane Dow. Action: Merge branch 'lpap-593310-fix=test-fluffy' into 'master'. Test results: 375 passed, 344 failed, 20 skipped, 13 errored, 0 retried, 0 cancelled.

Запуск из AllureTestOps

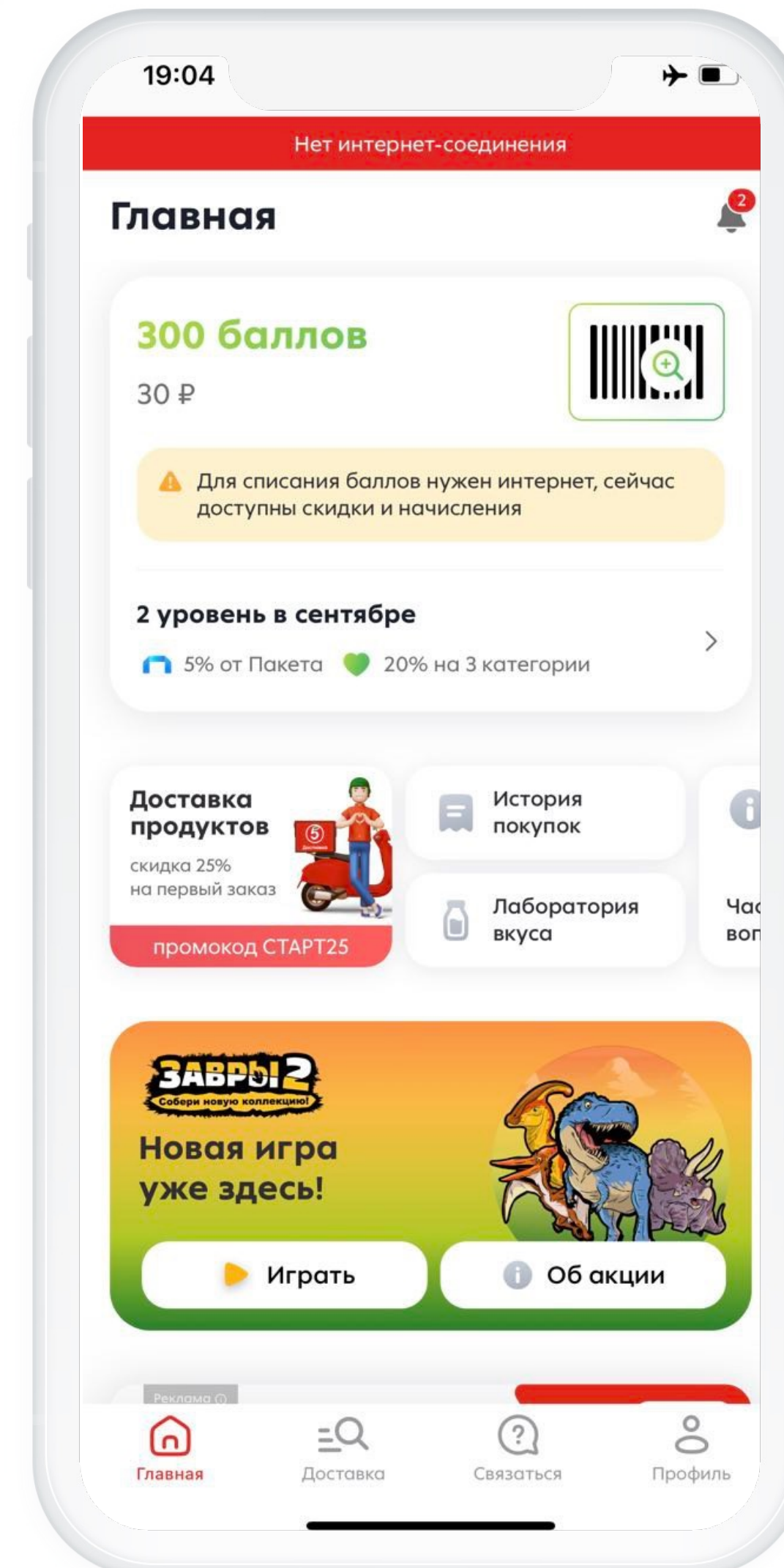


Проблемы



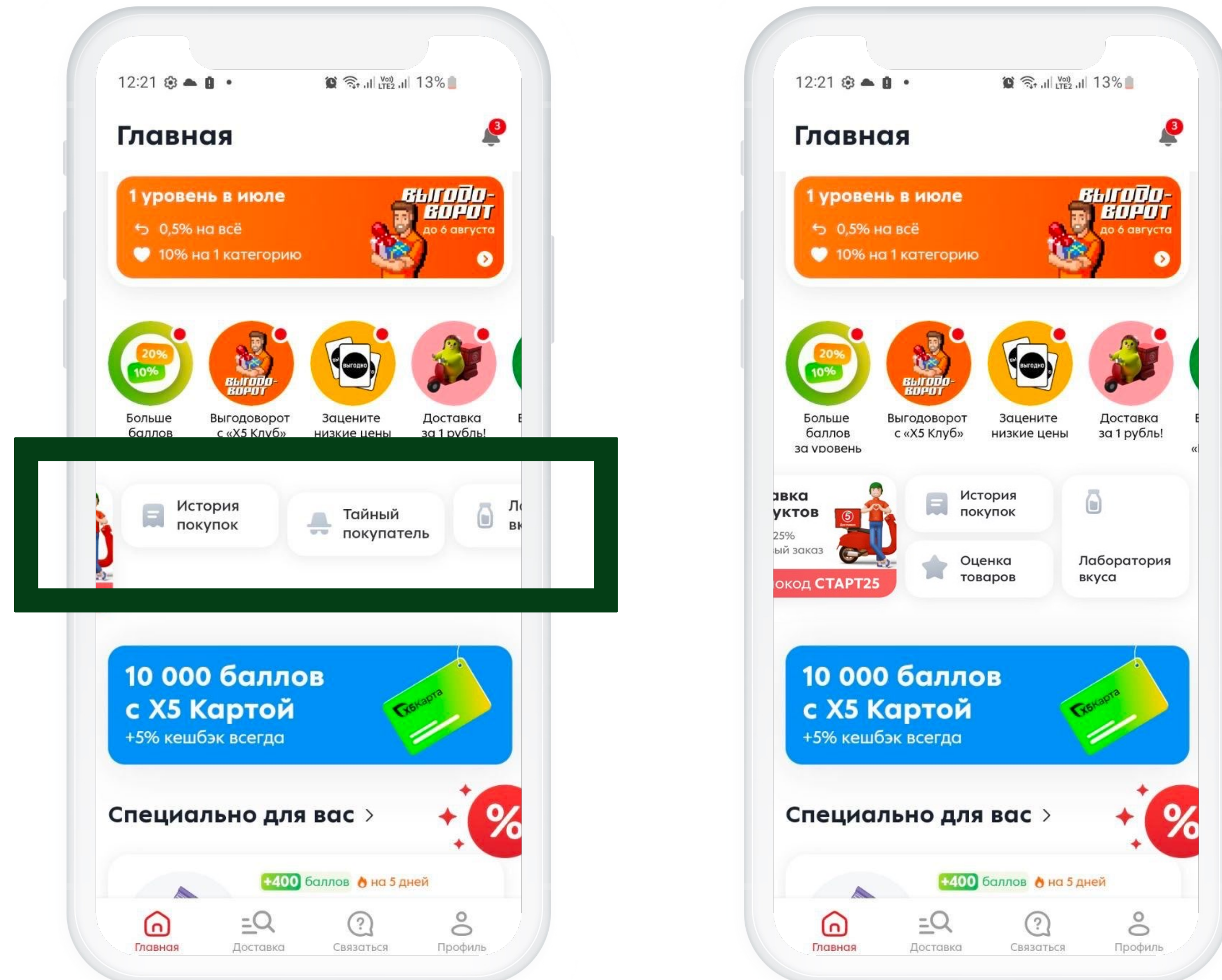
А какие минусы в такой автоматизации?

- Оффлайн не взлетел

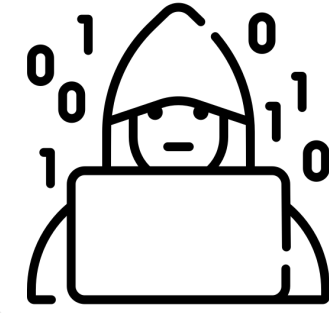


А какие минусы в такой автоматизации?

- Оффлайн не взлетел
- Отловили дефект в другом браузере



Отключение сервис воркера



settings.js

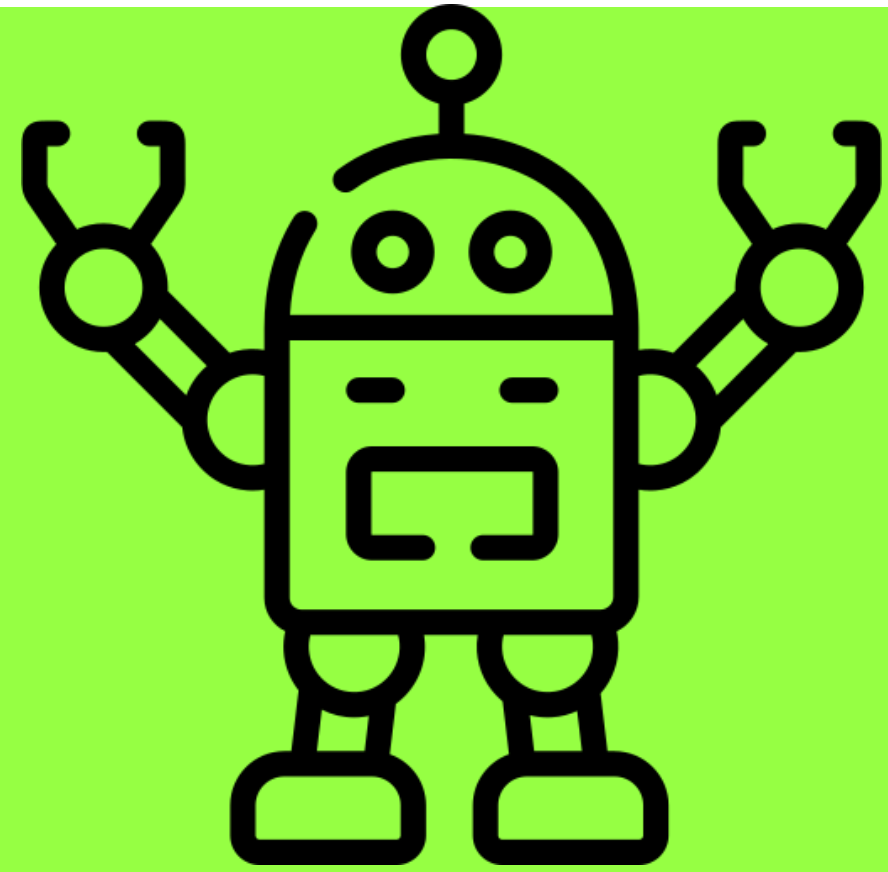
```
Cypress.on('window:before:load', () => {  
  navigator.serviceWorker.getRegistrations()  
  .then(function (registrations) {  
    for (let registration of registrations) {  
      registration.unregister().then()  
    }  
  })  
})
```

The screenshot shows the Cypress test runner interface. At the top, there are icons for a menu, a checkmark, a red 'X', and a refresh button. Below that, there's a dropdown menu and a square button. The main area shows a file named 'favoriteCategories.cy.js' with a duration of '00:09'. Underneath, there's a 'Suite' section with a 'Case' section. The 'TEST BODY' section is expanded, showing a red error message: '(uncaught exception) TypeError: Cannot read properties of undefined (reading 'getRegistrations')'.

Выводы



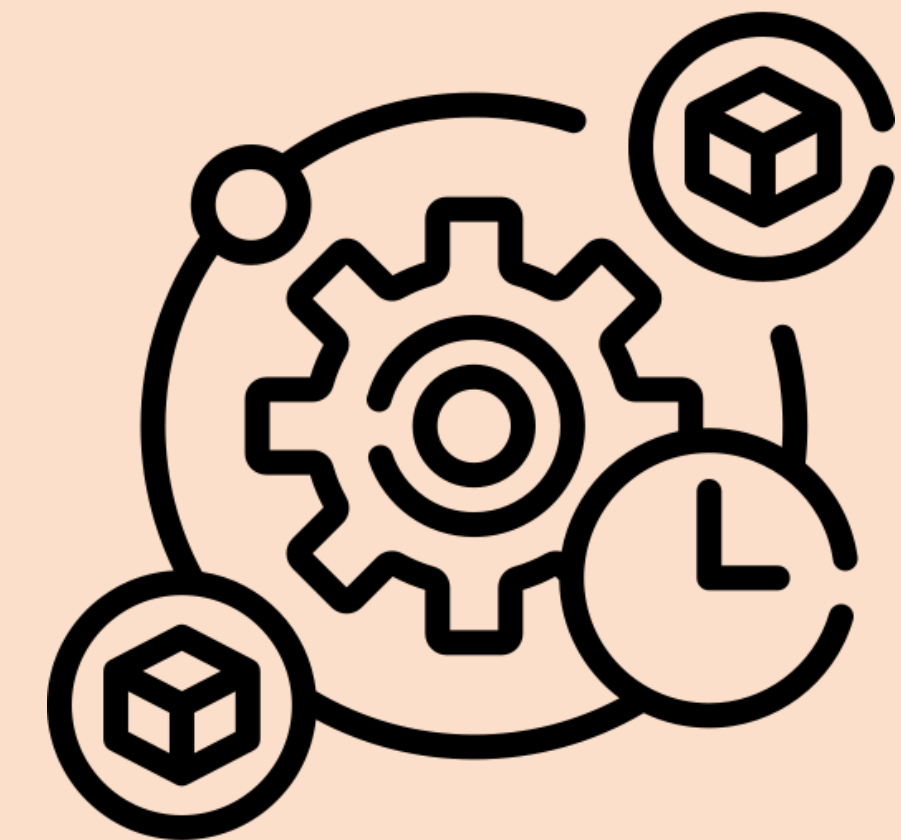
Бизнесовые выводы



Автоматизировали 70% регресса веб части гибридного мобильного приложения



Все желающие писать автотесты, начали это делать без подготовки

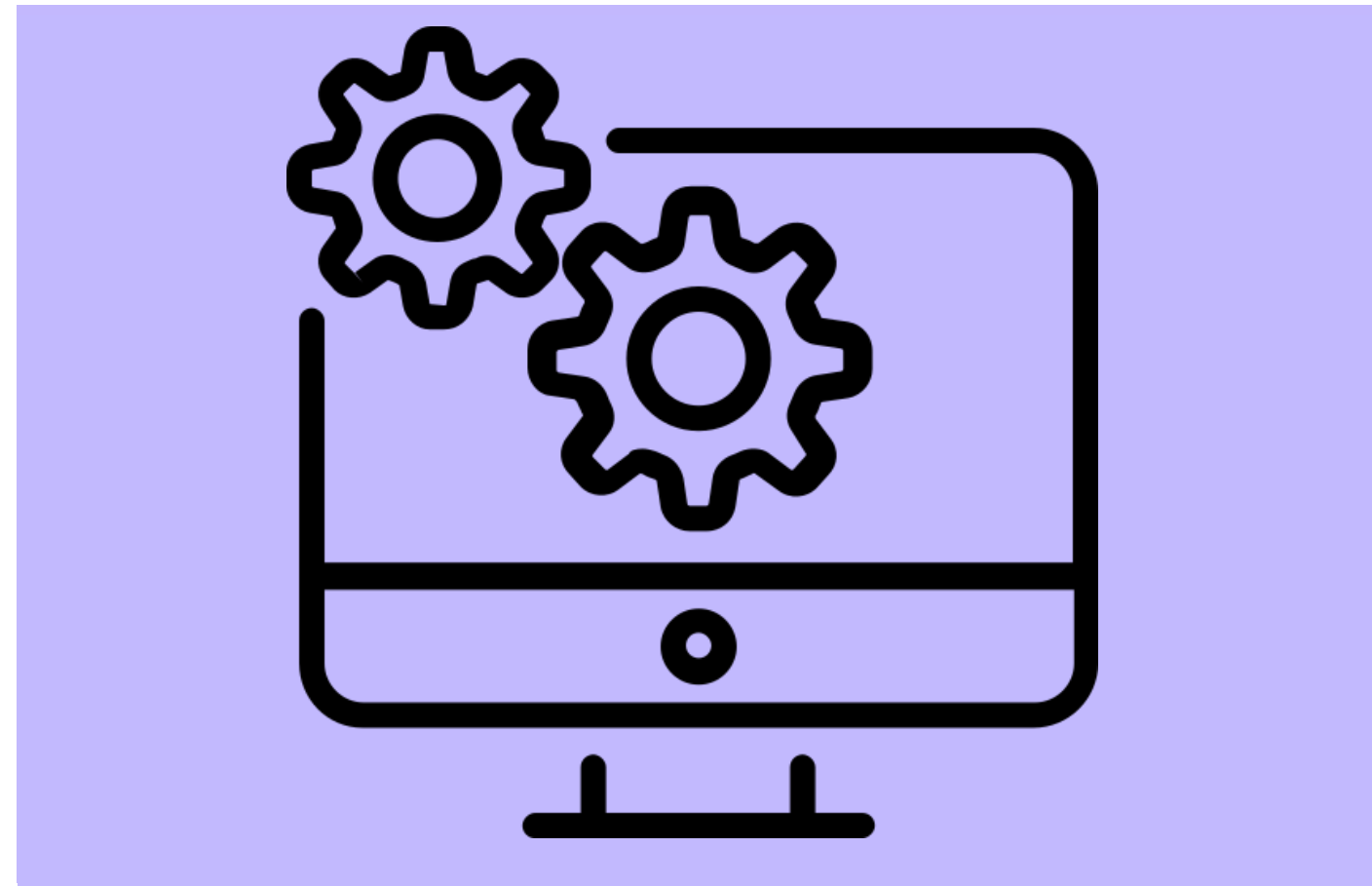


Сократил время регресса, чем увеличили стабильность и частоту релизов

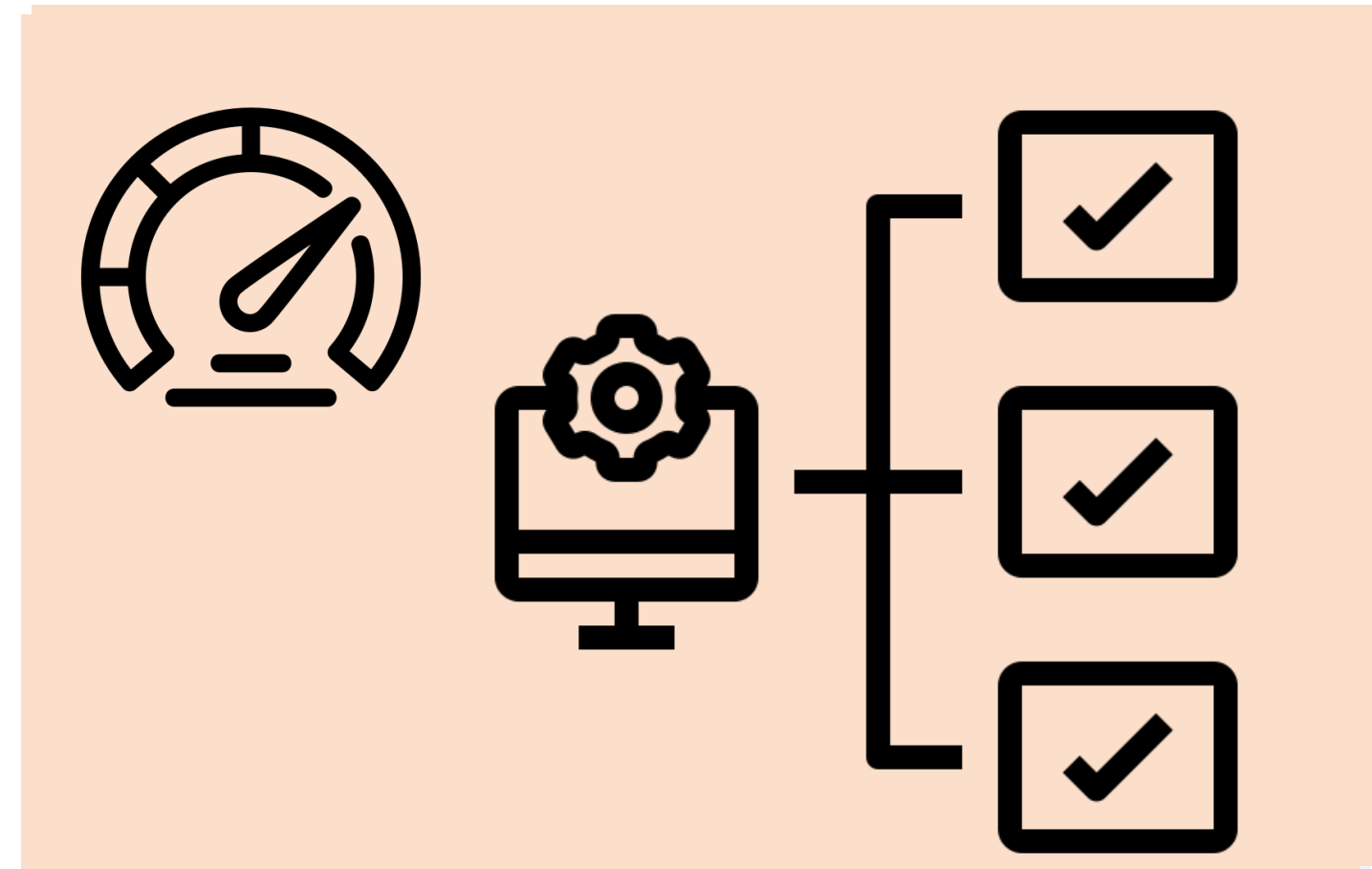
Технические выводы



Не нужно запускать эмуляторы



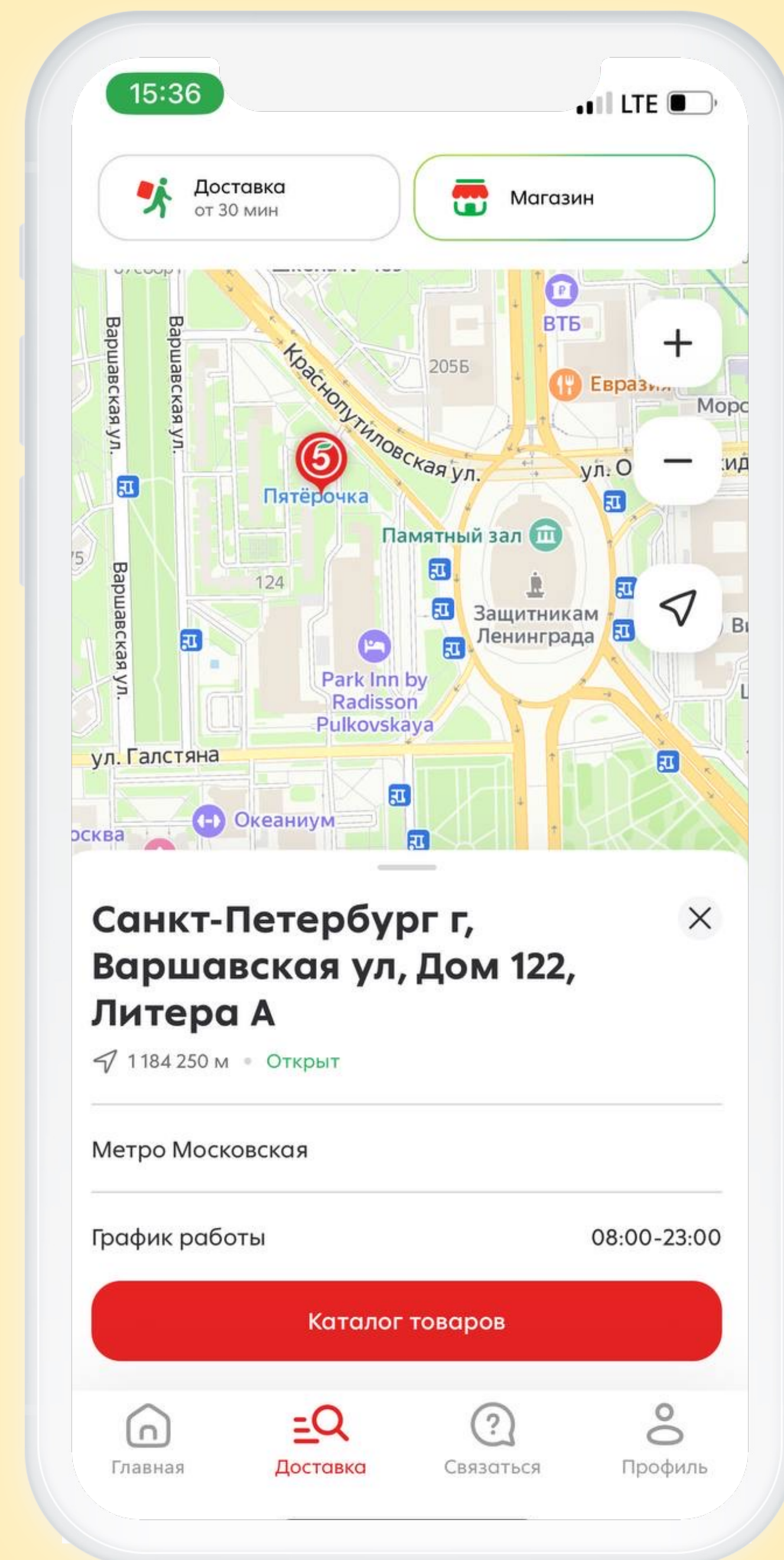
Нет привязки к операционной системе



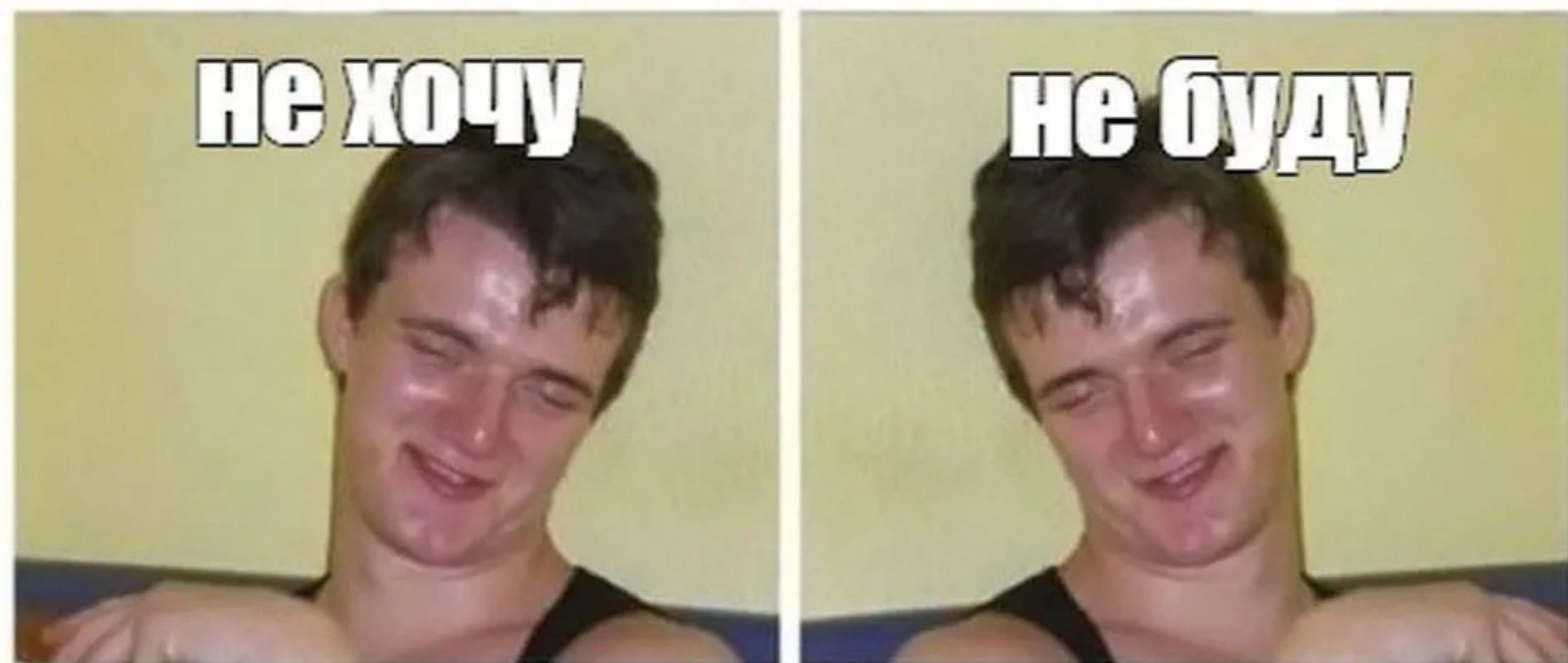
Быстрые тесты в несколько потоков без ручного вмешательства

Спасибули за соавторство и помощь с дизайном:

- ✔ **Саша Сухинин,**
менеджер направления тестирования
- ✔ **Саша Мелентьев,**
руководитель команды тестирования
- ✔ **Анна-Мария Беджанян,**
самый топовый дизайнер X5Tech
- ✔ **Алексей Альтер-Песоцкий,**
наш куратор из программного комитета



Готовы ответить на вопросы!



<https://www.x5-tech.ru/>

