UFADEVCONF

Переход от REST API к GraphQL на примере реальных проектов

Антон Морев, Wormsoft

Антон Морев

Wormsoft

- Приложения
- Сайты
- Системы автоматизации (CRM, ERP)
- Маркетинг
- Сопровождение проектов

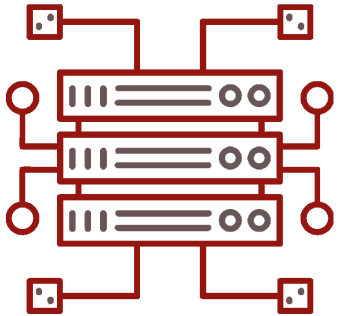
IT-Директор

- Оптимизация процессов разработки
- Review ключевых проектов

Взаимодействие с сервером

КЛИЕНТ

API



СЕРВЕР

DATABASE

CACHE

OTHER API

QUEUE

…

ETC

Стандартное решение на REST

UFADEVCONF

GET /goods

POST /order

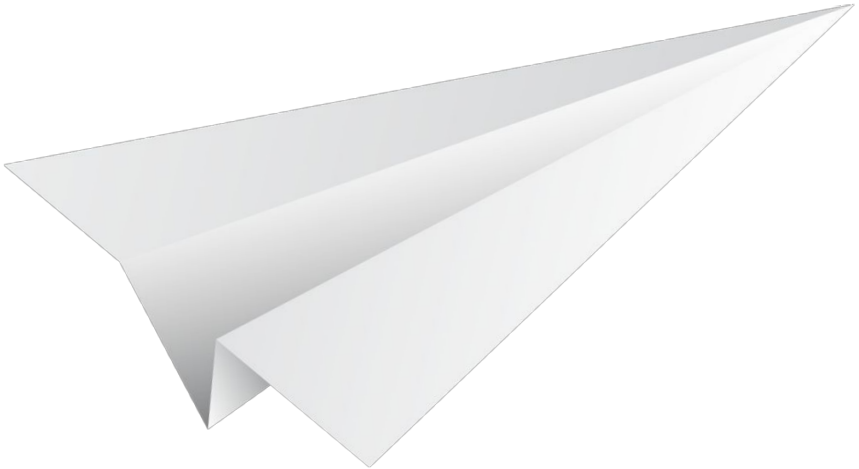
UFADEVCONF

ДОКУМЕНТАЦИЯ

№1

БЕЛЫЙ ЛИСТ

UFADEVCONF



“Я не хочу писать документацию,
я хочу писать код”

© “Senior” Developer

№2

МУДРЕЦ

UFADEVCONF

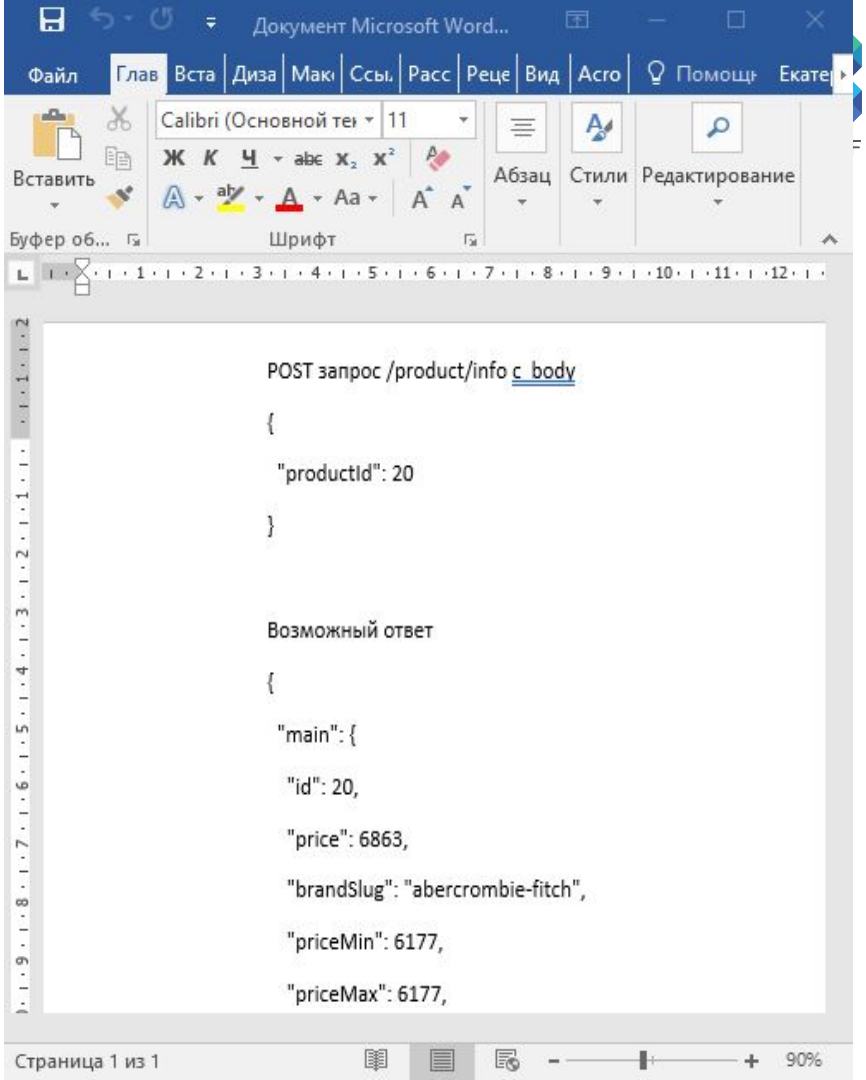
в каком поле у нас текущий статус?

спроси у Ивана - он знает где у нас это

№3

WORD, EXCEL , GOOGLEDOC

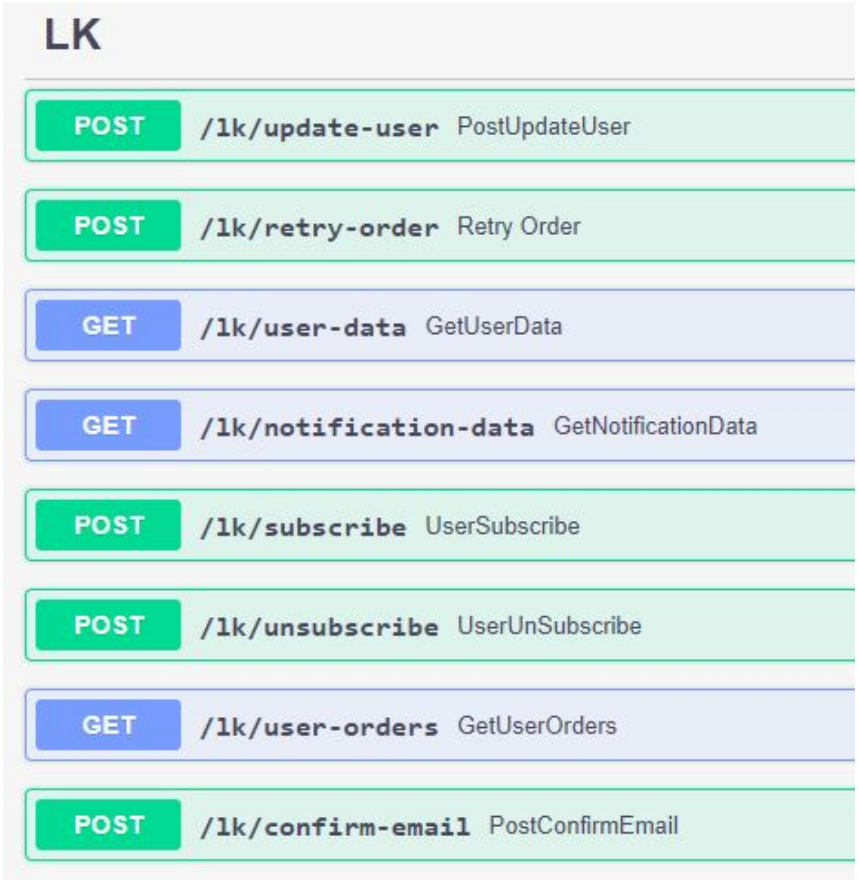
«Мы храним документацию
в docx файлах под git
со всеми JSON ответами и
запросами»



№4

Инструменты для документации (swagger и тд)

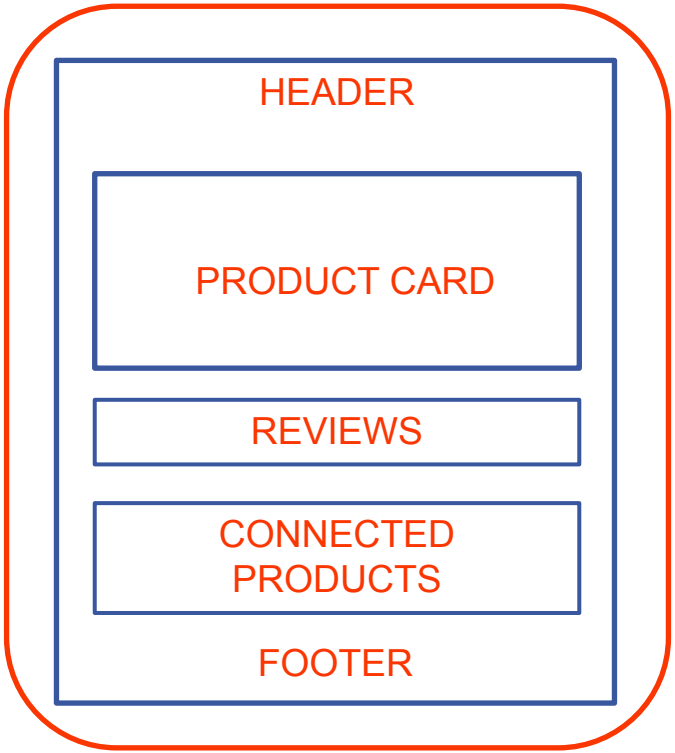
UFADEVCONF



UFADEVCONF

РАЗГРАНИЧЕНИЕ ДАННЫХ

Количество запросов



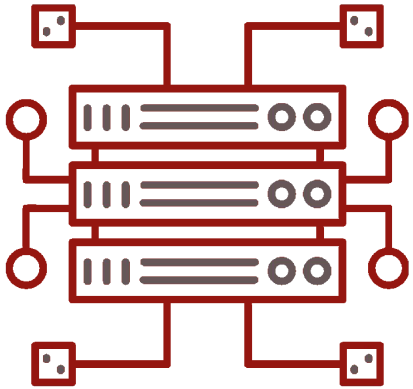
getHeaderData

getFooterData

getProductCardData

getReviewsData

getDeliveryData



UFADEVCONF

КАТАЛОГ ТОВАРОВ

КОРЗИНА

ТОВАР

ОТЗЫВЫ

ИСТОРИЯ ЗАКАЗОВ

КАРТОЧКА ТОВАРА

UFADEVCONF

GET /allRoundProduct?id=123


```
{
  "id": 123,
  "title": "Bentley For Men",
  "price": 100,
  "description": "Лучшие духи",
  "brand": "Bentley",
  "connectedProducts": [],
  "photos": [],
  "countInBasket": 3,
  "reviews": [],
  "variants": []
}
```

UFADEVCONF

СВОЯ МОДЕЛЬ НА КАЖДУЮ СИТУАЦИЮ

UFADEVCONF

GET /product/card?id=123

```
{
  "id": 123,
  "title": "",
  "price": "",
  "description": "",
  "brand": "",
  "photos": []
}
```

UFADEVCONF

GET /product/connected?id=123

```
[
  {
    "id": 123,
    "title": "",
    "price": 100,
    "photo": {}
  }
]
```

UFADEVCONF

GET /product/list

```
[
  {
    "id": 123,
    "title": "",
    "photo": [],
    "rating": 3,
    "price": 100
  }
]
```

UFADEVCONF

ИЕРАРХИЧЕСКАЯ МОДЕЛЬ

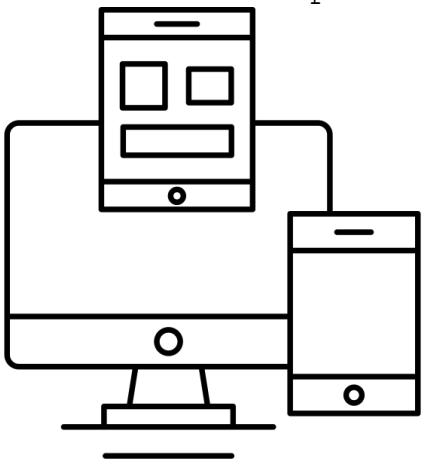
GET /product/full?id=123

```
{
  "commonInfo": {},
  "cardInfo": {},
  "listInfo": {},
  "basketInfo": {}
}
```

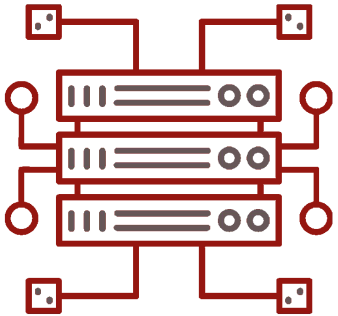
Интернет-магазин. Решения на REST

UFADEVCONF

product/full?id=1&**fields=cardInfo,basketInfo**



ЯЗЫК ЗАПРОСОВ



API

только **cardInfo** и **basketInfo**

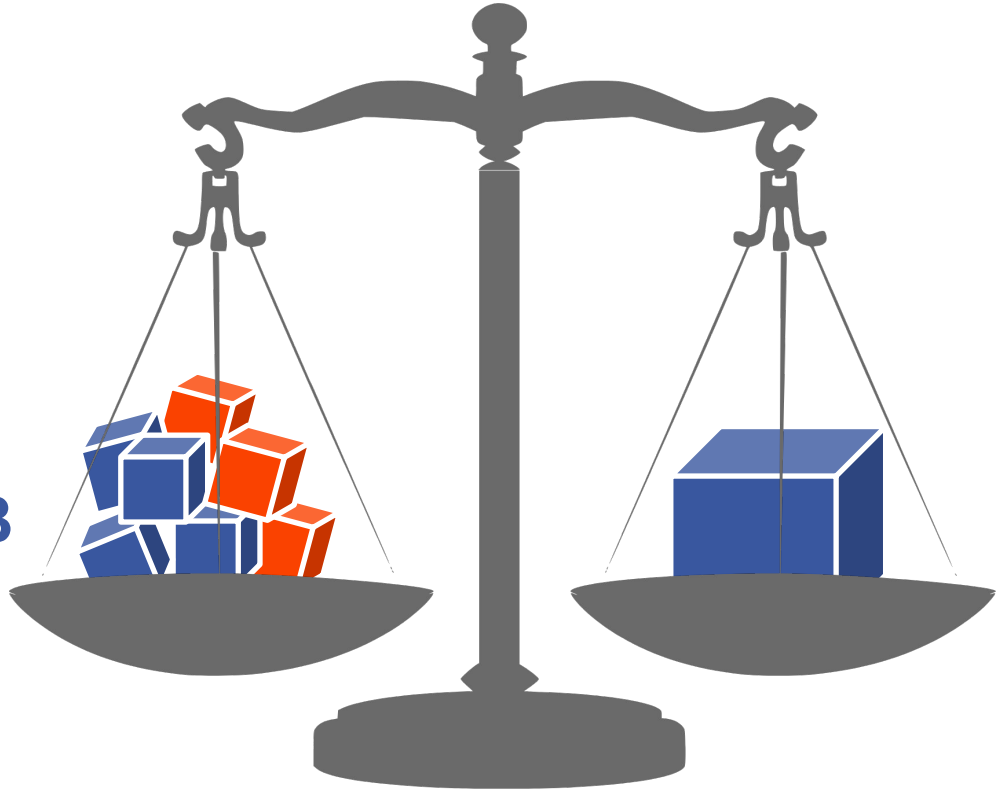
UFADEVCONF

```
getModel?id=1&fields=title,
description,price

getModel?id=1&fields=title,price,
connectedProducts.title,
connectedProducts.description

getModel?id=1&fields=title,description,
rating,price, reviews.user.name,reviews.
user.id,reviews.date,reviews.text,
photos.src,photos.alt,
connectedProducts.title,
connectedProducts.price,
connectedProducts.rating,
dicsount.rate,discount.amount
```

Интернет-магазин. Решения на REST

UFADEVCONF

МНОГО ЗАПРОСОВ (удобно)



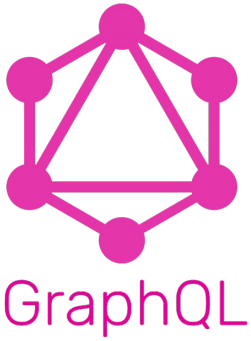
ОДИН ЗАПРОС (неудобно)

Интернет магазин. Желаемое решение

Автоматическая документация

Можно выбирать что именно мы хотим получить

Несколько сущностей в одном запросе



РЕАЛИЗАЦИЯ НА GRAPHQL

UFADEVCONF

ОПИСАНИЕ СХЕМЫ

```
type Product {
  id: ID!
  title: String
  description: String
}

type Query  {
    product (id: ID!): Product
}
```

UFADEVCONF

ЗАПРОС

```
query {
    product (id: 123)                    {
        title
        description
    }
}
```

UFADEVCONF

ОТВЕТ

```
{
    "product": {
        "title": "Духи",
        "description":"Описание товара..."
    }
}
```

UFADEVCONF

ВАРИАНТЫ ТОВАРА

```
type Product {
    id: ID
    title: String
    description: String
    variants: [ProductVariant]
}
```

UFADEVCONF

ПОЛУЧЕНИЯ ПОЛЯ VARIANTS

```
query {
    product (id: 123) {
        title
        variants {
            id
            title
        }
    }
}
```

UFADEVCONF

ОТВЕТ

```
{
  "title": "Chanel Chance Eau Tendre",
  "variants": [
    {
      "id": 1,
      "title": "Подарок, 200мл"
    },
    {
      "id": 2,
      "title": "Пробник, 1.5мл "
    }
  ]
}
```

UFADEVCONF

ИНТЕРНЕТ-МАГАЗИН РЕШЕНИЯ НА GRAPHQL

```
type ProductVariant {
    id: ID
    title: String
    price: Float
    priceDiscount: Float
    image: ImageModel
}
```

UFADEVCONF

ЗАПРОС ДЛЯ СТРАНИЦЫ

```
query {
  header {
    #header fields
  }
  product (id: 123) {
    #product fields
  }
  footer {
    #footer fields
  }
}
```

product: Product

Товар

TYPE DETAILS

```
type Product {
  id: ID
  title: String
  description: String
  variants: [ProductVariant]
}
```

variants: [ProductVariant]

Вариации товара

TYPE DETAILS

```
type ProductVariant {
  id: ID
  title: String
  price: Float
  priceDiscount: Float
  image: ImageModel
}
```

title: String

Название вариации

TYPE DETAILS

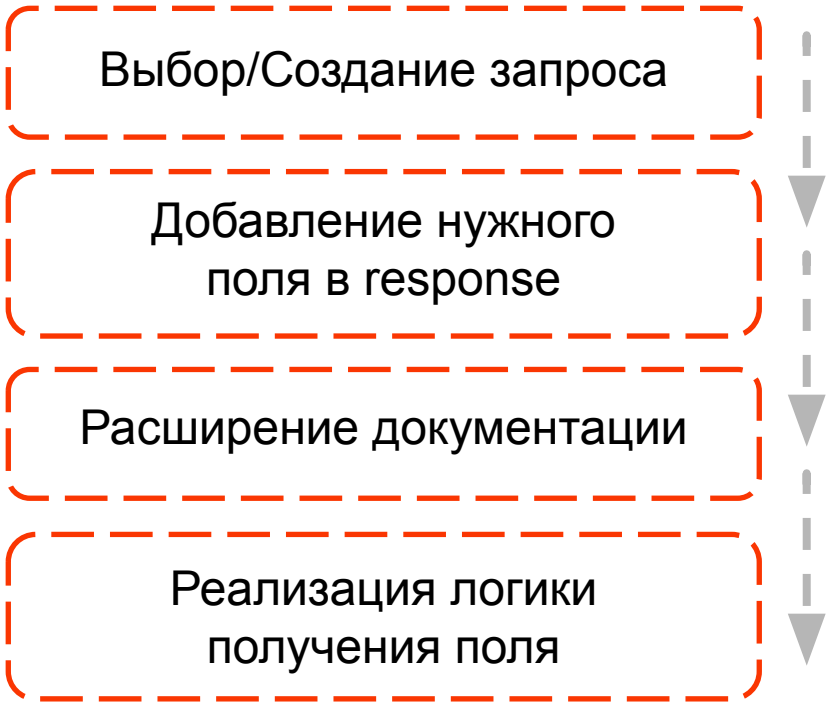
The `String` scalar type represents textual data, represented as UTF-8 character sequences. The String type is most often used by GraphQL to represent free-form human-readable text.

```
scalar String
```

Процесс внедрения нового свойства в сущность товара

UFADEVCONF

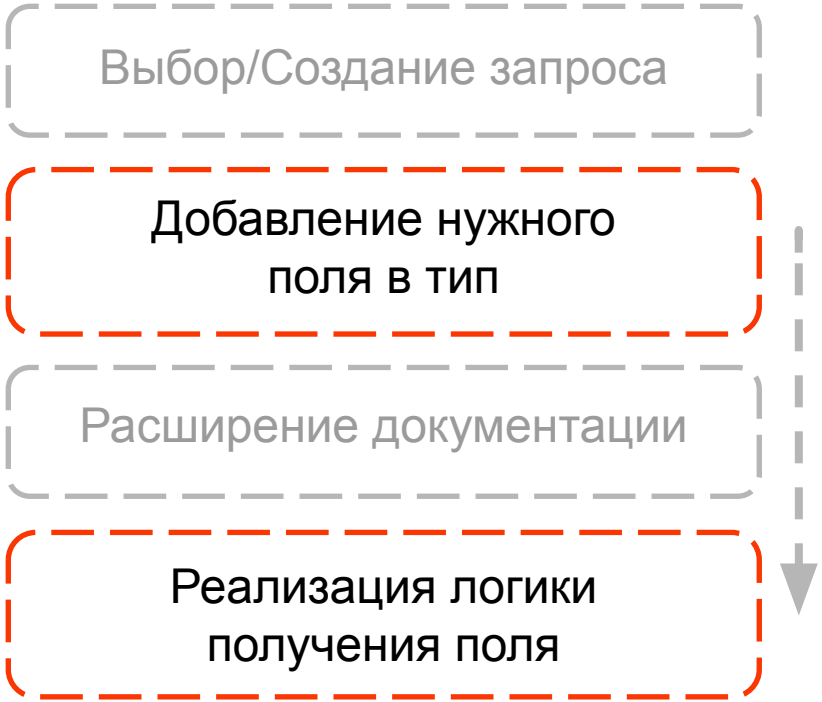
BACKEND ДО



Процесс внедрения нового свойства в сущность товара

UFADEVCONF

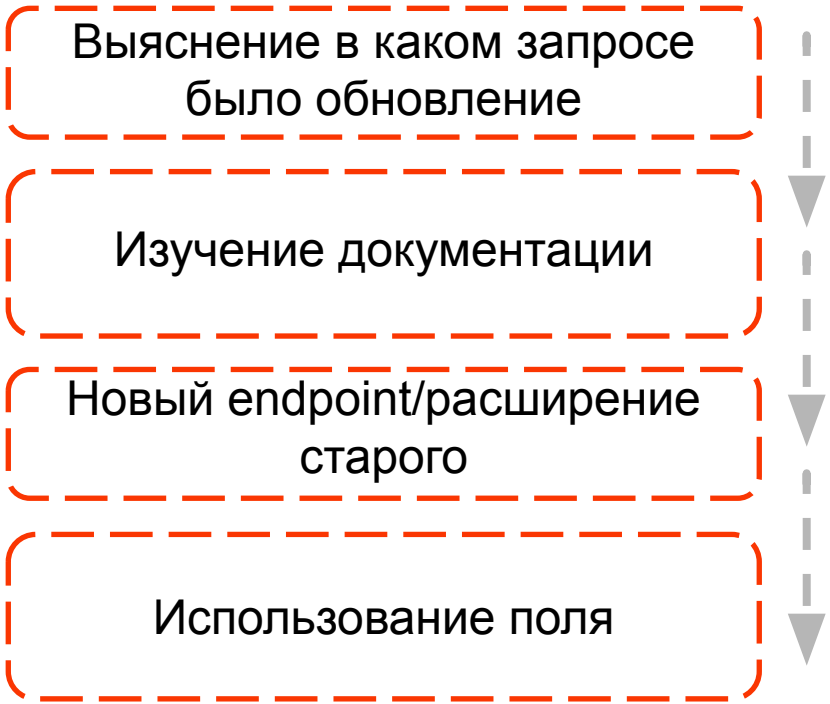
BACKEND ПОСЛЕ



Процесс внедрения нового свойства в сущность товара

UFADEVCONF

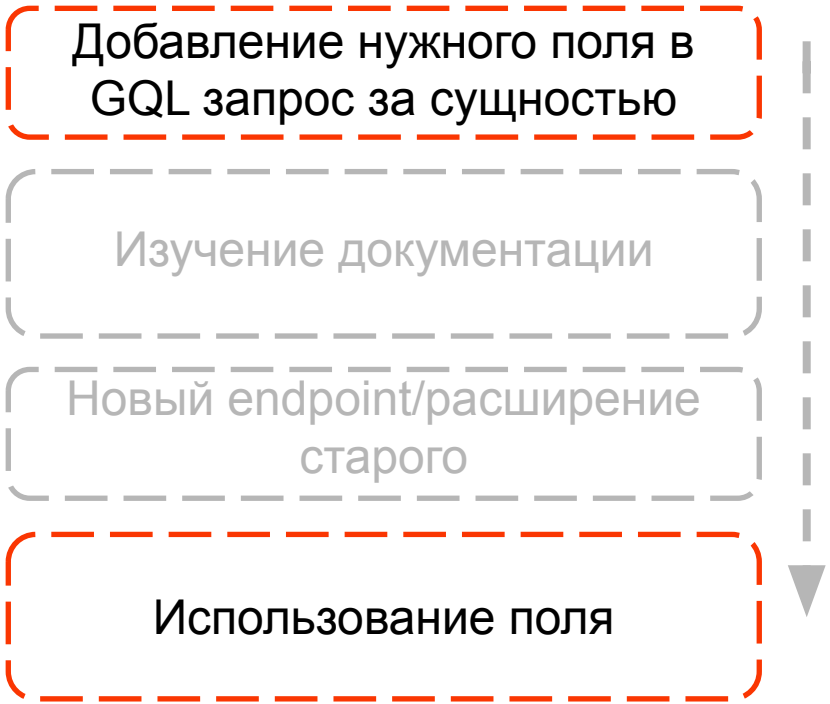
FRONTEND ДО



UFADEVCONF

Процесс внедрения нового свойства в сущность товара

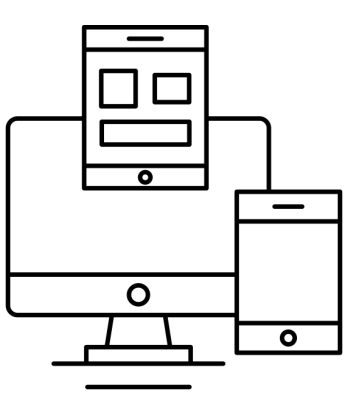
FRONTEND ПОСЛЕ



UFADEVCONF

ВНЕДРЕНИЕ GRAPHQL РАЗРАБОТКА С МИКРОСЕРВИСАМИ

Банк криптовалюты



Интерфейс

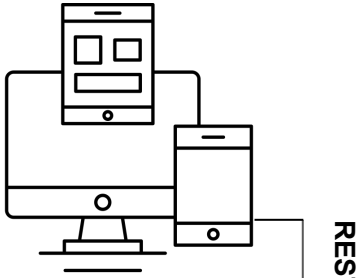
ЯДРО СИСТЕМЫ

КРИПТОВАЛЮТНЫЕ БИРЖИ

WALLET RPC

UFADEVCONF

ИНТЕРФЕЙС



REST

PHP Yii2

API Module

Email module

Payments module

Deposit module

Auth module

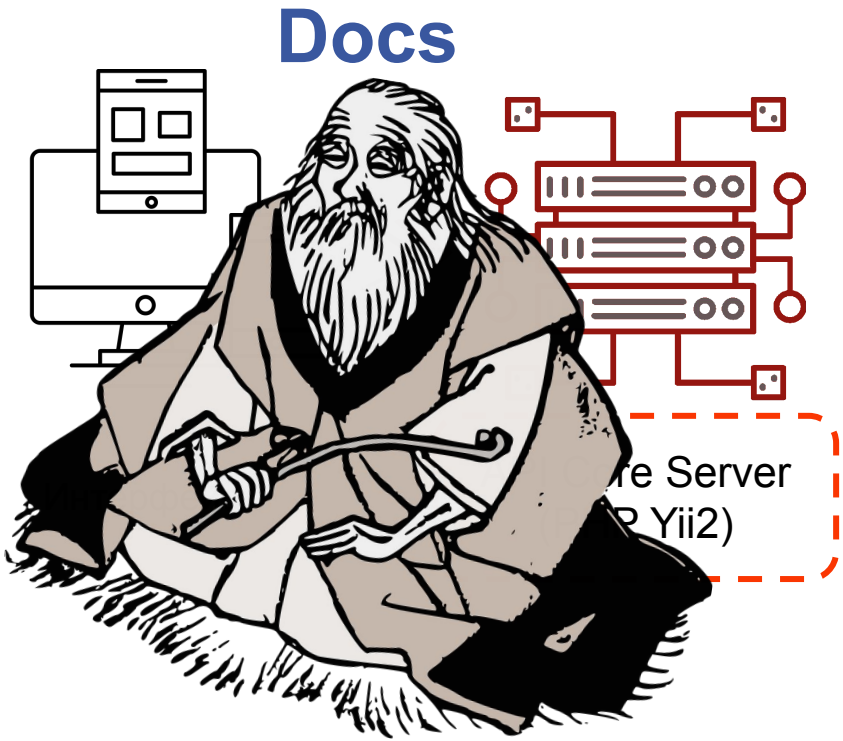
Wallet RPC (C++)

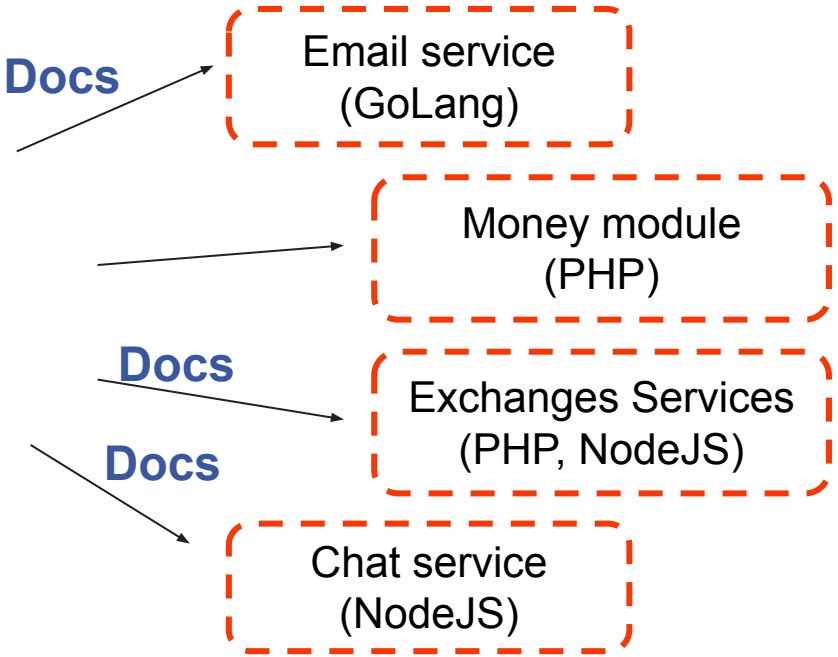
Exchange API

UFADEVCONF

```php
class TransactionsController extends Controller
{
    public function actionIndex($page = 1, $type = null)
    {
        $userId = Yii::$app->user->id;

        return $this->transactionsRepository
            ->getTransactions($userId, $page, $type);
    }
}
```

GET	/transactions	Получения списка транзакций текущего пользователя
POST	/money/withdraw-request	Начала процесса вывода денег. Отправка кода подтверждения


Docs
re Server
P Yii2)


Docs
Email service
(GoLang)
Money module
(PHP)
Docs
Exchanges Services
(PHP, NodeJS)
Docs
Chat service
(NodeJS)

UFADEVCONF

```
'resolve' => function($value, $args, $context, ResolveInfo $info) {
    //Получение любой информации
}
```

ПРЯМОЙ ЗАПРОС В БД

```
'resolve' => function($value, $args, $context, ResolveInfo $info) {
    return DB::getInstance()->select('SELECT * from products');
},
```

СОБСТВЕННЫЙ РЕПОЗИТОРИЙ

```
'resolve' => function($value, $args, $context, ResolveInfo $info) {
    return $context->productRepository->getProducts($args['page']);
}
```

UFADEVCONF

```
GET /user/payments/list
```

```
query {
    user {
        payments {
            list {
                id
                amount
            }
        }
    }
}
```

UFADEVCONF

```
GET /user/payments/list
GET /user/info
GET /user/system/status
GET /user/notifications/list
```

```
query {
    user {
        payments {
            list {
                id
                amount
            }
        }
        info {
            name
            money
        }
        system {
            status
        }
        notifications {
            list {
                id
                text
            }
        }
    }
}
```

UFADEVCONF

ВНЕДРЕНИЕ GRAPHQL ОПТИМИЗАЦИЯ РАЗРАБОТКИ

Внедрение GraphQL. Админ-панель

UFADEVCONF

SITE

- сбор данных с API сервера
- отображение данных
- Frontend логика

API Server

- сущности
- бизнес логика
- обработка заявок
- система оповещений

Admin Panel

- управление примитивными CRUD сущностями
- управления бизнес логикой (формы)

Внедрение GraphQL. Админ-панель

UFADEVCONF

Описание сущности

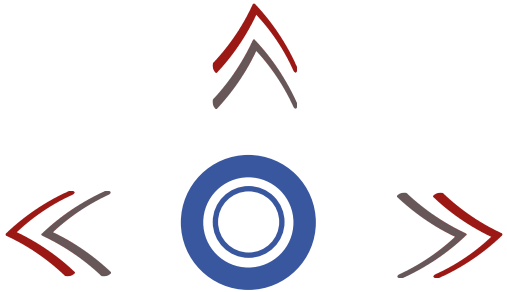
автоформирование

- GraphQL схема
- resolvers для каждого поля каждой сущности
- интерфейс админ-панели

UFADEVCONF

```
{
 “type”: “input_text”,
 “label”: “Название”
 “name”: “title”
}
```

Название
Кропоткинская



```
metro {
   list {
     items {
       title
       }
    }
 }
```

UFADEVCONF

Офис 18м²

ул. Большая Полянка, д.2/10, стр. 1
м. Полянка, м. Третьяковская, м. Кропоткинская

Офис для 4-6 человек, полностью готовый к работе.

100 000 РУБ. МЕСЯЦ

```
query {
  offices {
    title
    photoSrc
    address
    metro {
      list {
        title
      }
    }
    shortDescription
    price {
      month
    }
  }
}
```

Изменение в процессе добавления/ изменения примитивной сущности

UFADEVCONF

BACKEND

ДО

Разработка Логики хранения данных | Разработка логики CRUD | Разработка интерфейса управления | Добавление/ расширение запроса | Обновление документаци и API

ПОСЛЕ

Изменение в процессе добавления/ изменения примитивной сущности

UFADEVCONF

FRONTEND

ДО

- Изучение документации
- Добавление нового запроса
- Обработка данных

ПОСЛЕ

- Изучение документации
- Расширение Query
- Обработка данных

UFADEVCONF

ОБНОВЛЕНИЕ ДАННЫХ ЧЕРЕЗ GRAPHQL

GraphQL. Мутации

UFADEVCONF

```
mutation {
  office {
    update (id:123, model: {title: "newTitle"})
  }
}
```

GraphQL. Обновление данных

UFADEVCONF

```
POST /taskModule/task/mainInfo?id=123
{
    title: “title”,
    description: “description”
}
```

```
{
    taskModule {
        task (id: 123) {
            mainInfo (
                title: “title”,
                description: “description”
            )
        }
    }
}
```

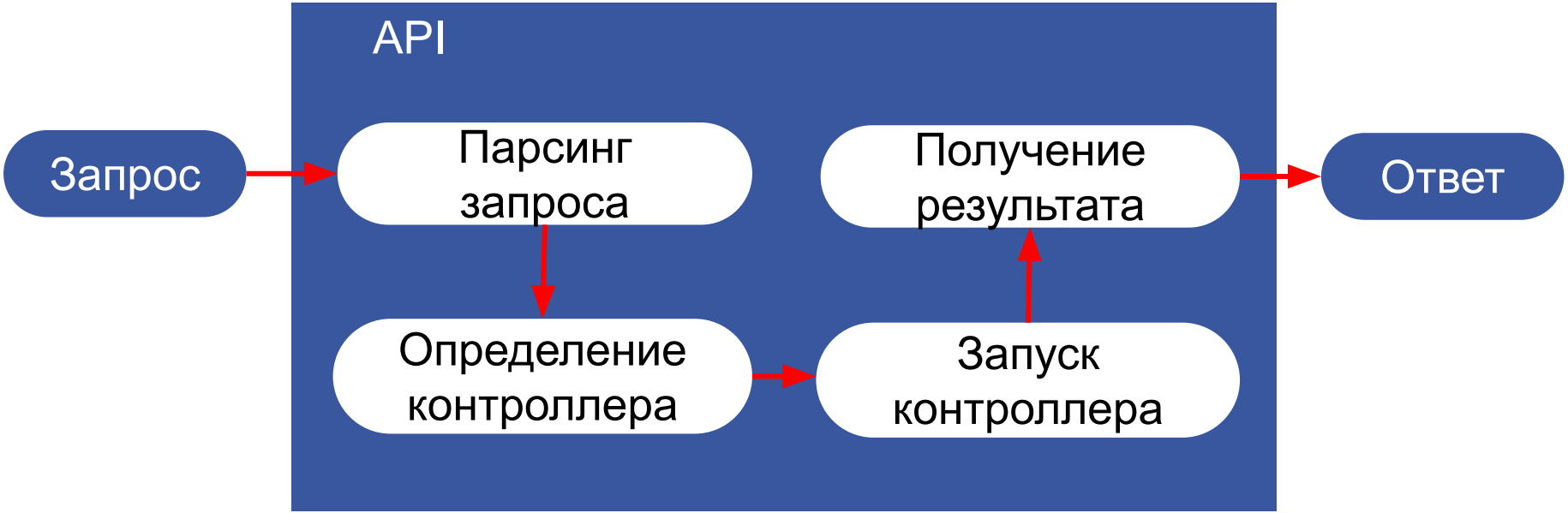
UFADEVCONF

UFADEVCONF

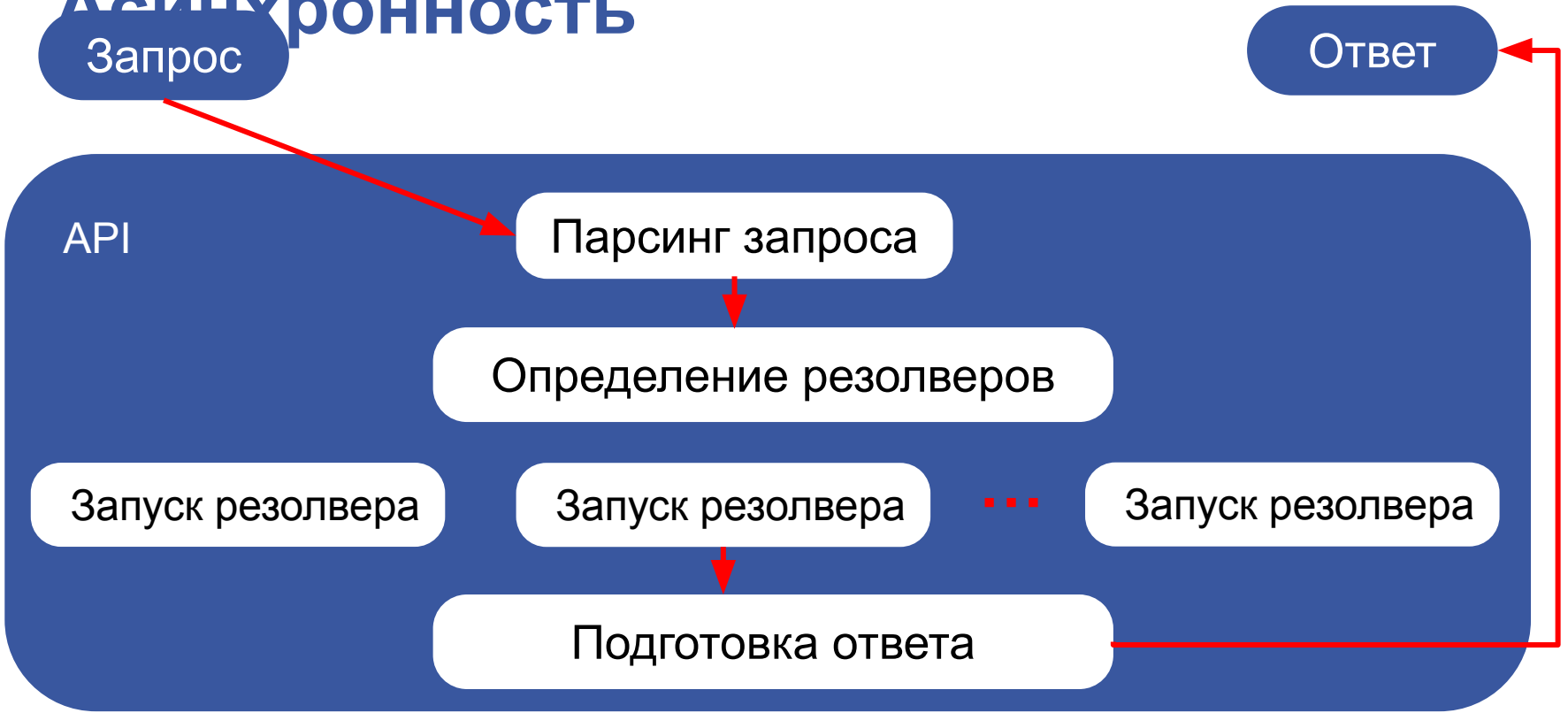
ОСОБЕННОСТИ GRAPHQL

Обычный запрос в REST

UFADEVCONF



Как работать с GraphQL. Асинхронность



За чем следить в GraphQL?

UFADEVCONF

- N+1
- Сложность запросов
- Сложность контроля
- Кэширование
- Асинхронность
- Порог вхождения

Особенности GraphQL. N+1

UFADEVCONF

```
query {
  task {
    list {
      id
      title
    }
  }
}
```

Особенности GraphQL. N+1

UFADEVCONF

```
query {
  task {
    list {
      id
      title
      subtask {
        title
      }
    }
  }
}
```

Особенности GraphQL. N+1

UFADEVCONF

```
'resolve' => function ($root) {
        Loader::add($root->id);
        return new Deferred(
            function () use ($root) {
                Loader::prepareData();
                return Loader::get($root->id);
            }
        );
    },
]
```

Как работать с GraphQL. Сложность запроса

```
task {
  subTask {
    subTask {
      subTask {
        subTask {
          subTask {
            id
          }
        }
      }
    }
  }
}
```

UFADEVCONF

UFADEVCONF

ОГРАНИЧЕНИЕ ГЛУБИНЫ ЗАПРОСА

Query Depth < 4

```
1. task {
 2. subTask {
   3. subTask {
     4. subTask {
       5. subTask {
         6. id
       }
     }
    }
  }
}
```

UFADEVCONF

ОГРАНИЧЕНИЕ СЛОЖНОСТИ ЗАПРОСА

Query Complexity < 40

```
1. task {
 3*i2  subTask {
    3*i3  subTask {
      3*i4  subTask {
          3*i5  subTask {
                  3*i6  id
          }
        }
      }
   }
}
```

Как работать с GraphQL. Кэширование

UFADEVCONF

Кэширование на уровне web-сервера

Как работать с GraphQL. Кэширование

UFADEVCONF

Кэширование на уровне web-сервера

POST /graphql

NGINX

API

КЛИЕНТ

Приложение

~~Кешировать на основании body~~

Как работать с GraphQL. Кэширование

UFADEVCONF

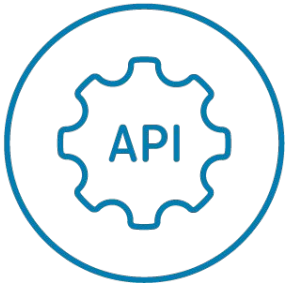
Кэширование на уровне web-сервера

Клиент

GET /api/info

cache

NGINX



Приложение

```
public function actionInfo()
{
    $query = ''; //фиксированный запрос
    return GraphQLHandler::handleQuery($query);
}
```

Внедрение GraphQL. Сложность адаптации

UFADEVCONF

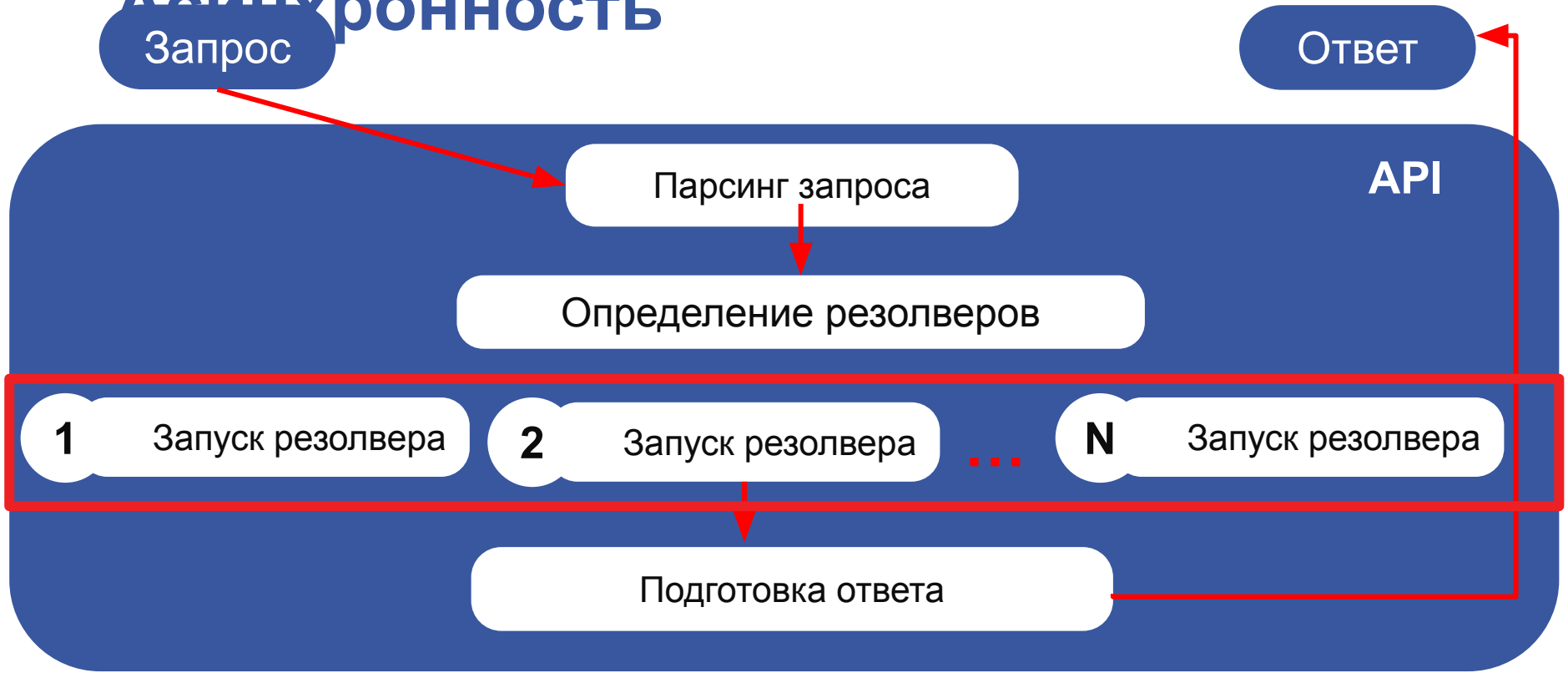
Зачем GraphQL,
если и так сойдет?

Внедрение GraphQL. Сложность адаптации

UFADEVCONF

А почему не GraphQL?

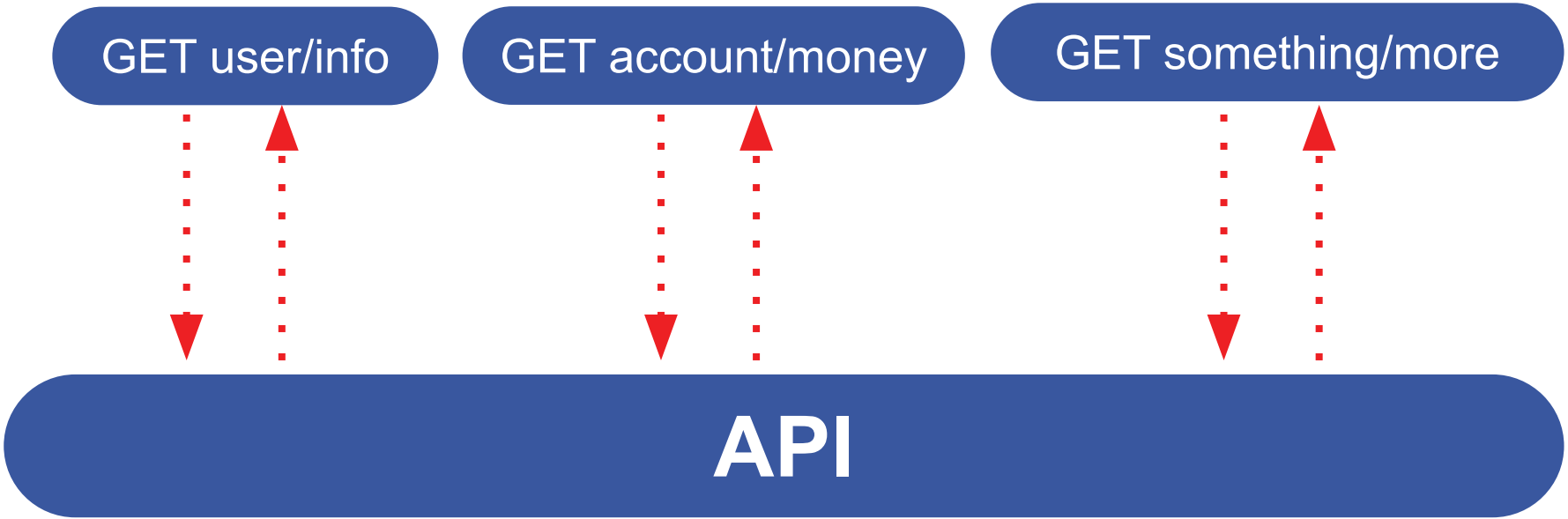
Как работать с GraphQL. Асинхронность



Как работать с GraphQL. Асинхронность

UFADEVCONF

REST



Как работать с GraphQL. Асинхронность

UFADEVCONF

Инициализация запроса ~ от 30мс (PHP)

Как работать с GraphQL. Долгие поля

```
query {
  easyData {} #30мс
  veryEasyData {} #2мс
  awesomeHeavyData {} #5000мс
}
```

UFADEVCONF

Внедрение GraphQL. Постановка вопроса

UFADEVCONF

Один тяжелый или много легких?

UFADEVCONF

ТЕСТОВЫЙ КЕЙС

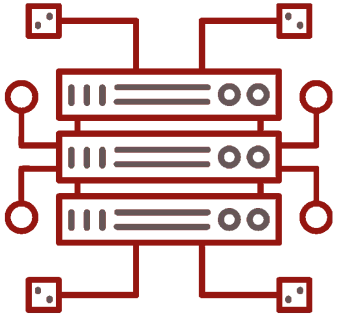
Нагрузочный тест. Инструментарий

Нагрузочный тест

UFADEVCONF

Nuxt.JS

API



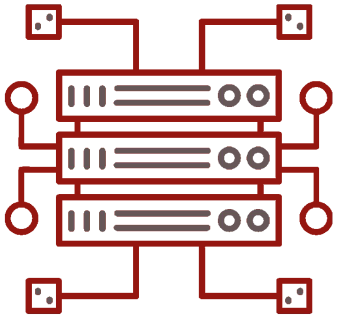
PHP-SLIM

SQLITE

Нагрузочный тест

Node.JS

API



PHP-SLIM

Redis

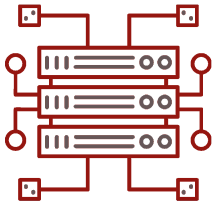
Нагрузочный тест. Инструментарий

UFADEVCONF

ubuntu

ОЗУ: 1ГБ
SSD: 30ГБ
CPU: 1

Backend



PHP 7.2
Slim Framework
webonyx/graphql-php

Frontend

Node.JS 10.16
Express
Axios HTTP Client

Нагрузочный тест. Условия

UFADEVCONF

- 30 одновременных пользователей
- Суммарное количество запросов: 200 каждый

Кейс 1. Список товаров

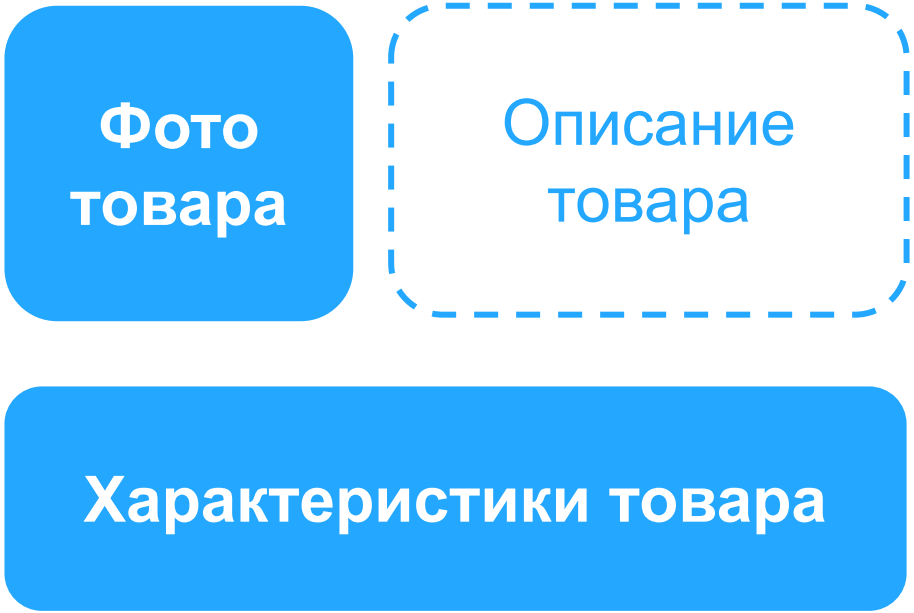
Кейс 1. Список товаров

UFADEVCONF

Среднее время ответа (мс)

Количество товаров	REST	GraphQL без оптимизации	GraphQL с оптимизацией
5	102	109	104
15	108	180	108
30	120	230	116
50	117	320	130
100	140	500	134

Кейс 2. Карточка товара без связей



Кейс 2. Карточка товара без связей

UFADEVCONF

Среднее время ответа (мс)

# Теста	REST	GraphQL
1	80	100
2	83	110
3	76	115
4	90	118
5	80	130

Кейс 3. Карточка со связями

UFADEVCONF

Кейс 3. Карточка со связями

UFADEVCONF

Среднее время ответа (мс)

Количество связанных товаров	REST	GraphQL без оптимизации	GraphQL с оптимизацией
5	220	300	150
15	240	420	148
30	232	510	145
50	273	604	155
100	310	1 100	180

Небольшой вывод

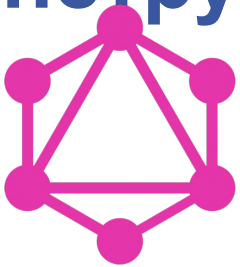
UFADEVCONF

Стоит ли подключать GraphQL в реальные проекты?

Да, но всегда помнить о:

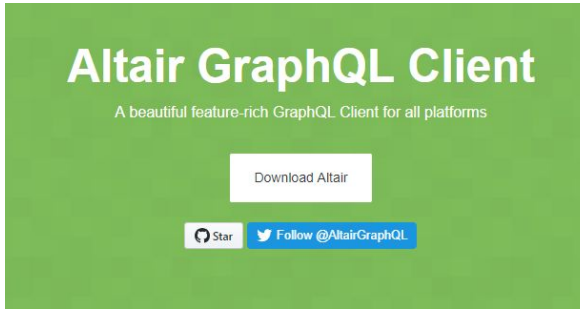
- N+1
- Сложностях кеширования
- N+1
- Сложности запроса
- N+1

Как работать с GraphQL. Инструменты



GraphiQL

- от создателей GraphQL



- Интерфейс
- Работа с подписками
- curl export
- CPU 20%

JS-GraphQL

- интеграция с кодом проекта

Как работать с GraphQL. Библиотеки

UFADEVCONF

Frontend

Vue.js — vuejs/vue-apollo

React — apollographql/react-apollo

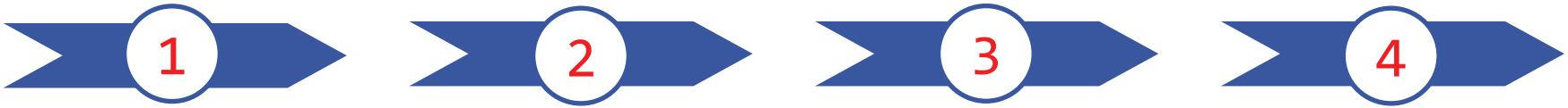
Backend

graphql-go/graphql

webonyx/graphql-php

graphql/graphql-js

Когда GraphQL усложняет процессы



1. В некоторых MVP
2. В системах отчетности и аналитики
3. В общих сервисах аутентификации
4. В командах где не знают GraphQL

Антон Морев

Благодарю за внимание!

wormsoft.ru

@amorev

GitHub amorev