

Операция «запрос – ответ». Наш подход к генерации клиентов для межсервисного взаимодействия

Сергей Солдатов
soldatov_sr@magnit.ru

О нас

Magnit Online – департамент, отвечающий за работу мобильных приложений и сайтов Магнит в сегменте b2c

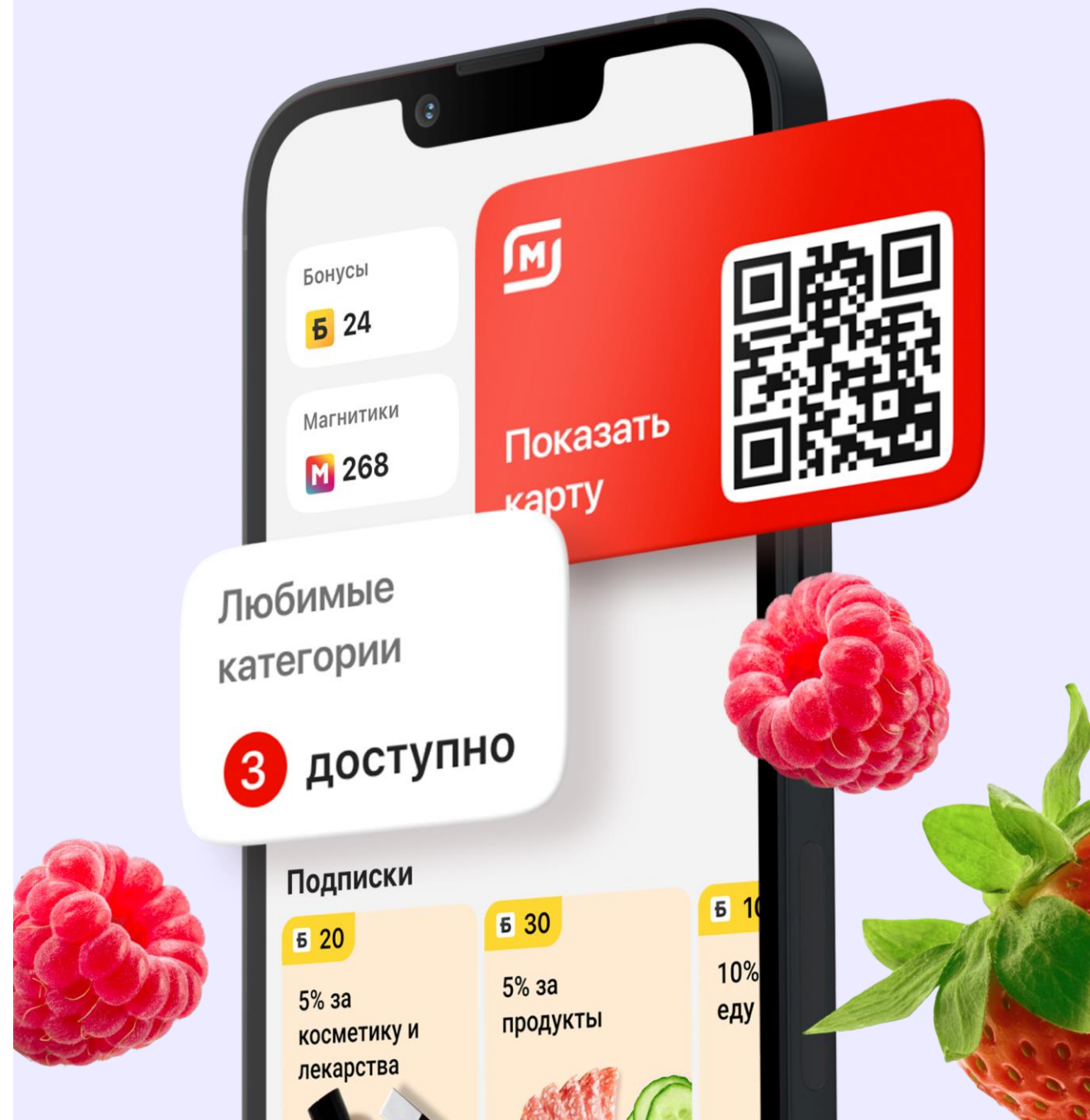
Приложения

- Магнит: акции и доставка
- Магнит: доставка продуктов

Сайты

- apteka.magnit.ru
- dostavka.magnit.ru
- magnit.ru
- magnitcosmetic.ru
- my.magnit.ru

GitHub





О нас

33

Команды

>150

Сервисов

>30

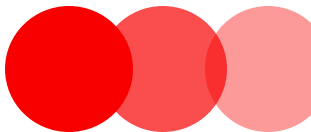
Сервисов
на Python

>10

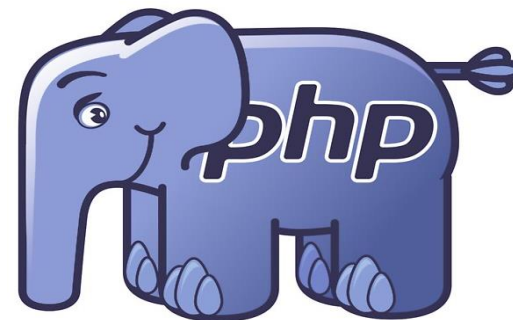
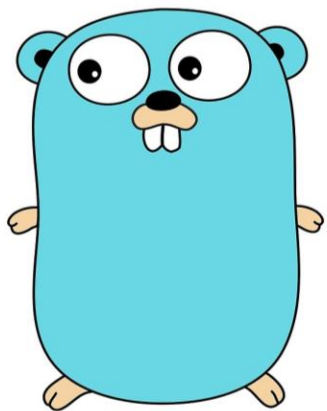
Внешних
сервисов

Agenda

1. Спецификации сервисов
2. Кодогенерация
3. Динамические gitlab пайплайны



Спецификации



Спецификации

A person wearing a VR headset and holding a blue rectangular object in a laboratory setting. The person is wearing a purple hoodie and a VR headset. They are holding a blue rectangular object in their right hand. The background shows a laboratory environment with a microscope and other equipment.

Когда разбираешься в API незнакомого сервиса без спецификации

Спецификации

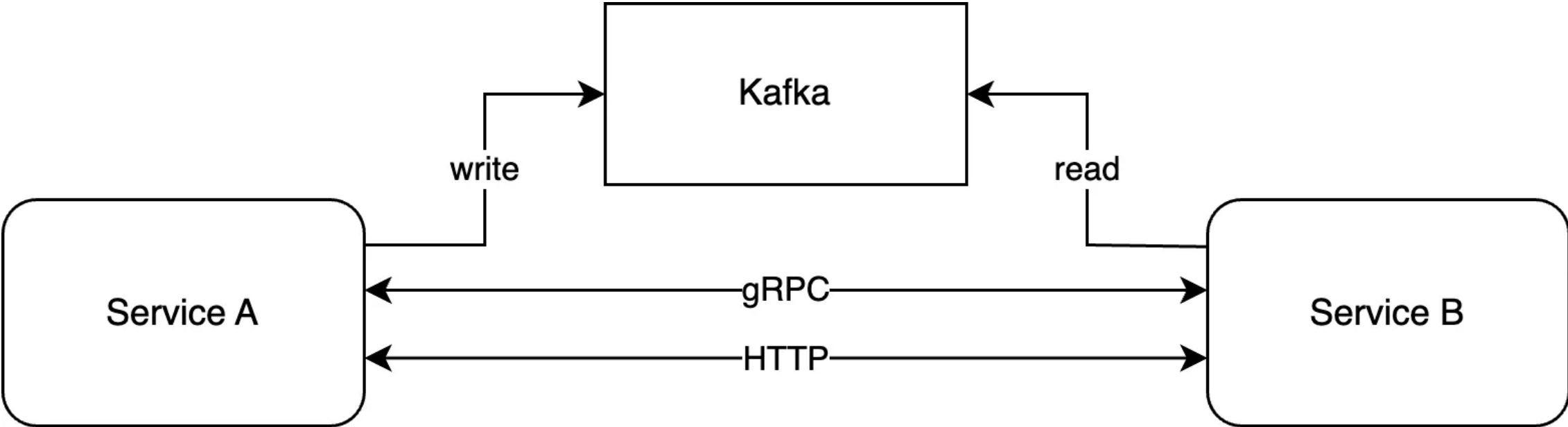
Описание сервиса на независимом языке

Главная задача

Позволить клиентам и документации синхронизировать свои обновления с изменениями на сервере

```
1  openapi: "3.0.0"
2  info:
3    version: 1.0.0
4    title: Swagger Petstore
5    license:
6      name: MIT
7  servers:
8    - url: http://petstore.swagger.io/v1
9  paths:
10 /pets:
11   get:
12     summary: List all pets
13     operationId: listPets
14     tags:
15       - pets
16     parameters:
17       - name: limit
18         in: query
19         description: How many items to return at one
20           time (max 100)
21         required: false
22         schema:
23           type: integer
24           format: int32
25     responses:
26       '200':
27         description: A paged array of pets
28         headers:
29           x-next:
30             description: A link to the next page of
31               responses
32             schema:
33               type: string
34             content:
```

Спецификации



Спецификации



Проблемы



Отрицание



Гнев



Торг



Депрессия



Принятие



Поддержка нового обязательного заголовка во всех интеграциях

Автосгенерированные клиенты (или SDK)

Требования к клиентам:

- Метрики
- Логгирование
- Трейсы
- Обязательные заголовки
- Упакован в Python-пакет
- ...

Кодогенерация

Pythogen – инструмент для генерации python HTTP-клиентов из OpenApi спецификаций, основанный на httpx и pydantic

Contributors 6



[https://github.com/artsmolin/
pythogen](https://github.com/artsmolin/pythogen)

Кодогенерация



```
openapi: 3.0.2
info: {title: Foo, version: 0.1.0}
paths:
  /user:
    post:
      tags:
        - user
      summary: Create user
      operationId: createUser
      responses:
        default:
          description: successful operation
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/User'
      requestBody:
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/User'
components:
  schemas:
    User:
      properties:
        id:
          type: integer
        username:
          type: string
      type: object
```

Кодогенерация



```
class Client:
```

```
    def __init__(
        self,
        base_url: str,
        timeout: int = 5,
        client_name: str = "",
        client: httpx.AsyncClient | None = None,
        headers: dict[str, str] | None = None,
        metrics_integration: MetricsIntegration | None = None,
        logs_integration: LogsIntegration | None = DefaultLogsIntegration(),
    ):
        """ ... """
```

```
        """ ... """
```

```
        self.client = client or httpx.AsyncClient(timeout=Timeout(timeout))
        self.base_url = base_url
        self.headers = headers or {}
        self.metrics_integration = metrics_integration
        self.logs_integration = logs_integration
        self.client_name = client_name
```

```
    async def createUser(
        self,
        *,
        auth: BasicAuth | None = None,
        content: str | bytes | None = None,
        body: User | dict[str, Any] | None = None,
        meta: PythogenMetaBox | None = None,
    ) -> None:
```

```
        """
```

```
        ~~~~~
        POST /user
```

```
        Operation ID: createUser
```

```
        Summary:      Create user
```

Кодогенерация

Pyprotogen – обёртка над protoc с дополнительными интерцепторами, которые покрывают наши требования



<https://github.com/sollof/pyprotogen>

```
1 # Generated by the gRPC Python protocol compiler plugin. DO NOT EDIT!
2 """Client and server classes corresponding to protobuf-defined services."""
3 import grpc
4
5 from . import cards_pb2 as core_dot_cards__v2_dot_cards__pb2
6
7
8 class CardsServiceStub(object):
9     """CardsService сервис для работы с картами пользователя
10     """
11
12     def __init__(self, channel):
13         """Constructor.
14
15         Args:
16             channel: A grpc.Channel.
17         """
18         self.V3GetCardList = channel.unary_unary(
19             '/cards.CardsService/V3GetCardList',
20             request_serializer=core_dot_cards__v2_dot_cards__pb2.V3CardListRequest.SerializeToString,
21             response_deserializer=core_dot_cards__v2_dot_cards__pb2.V3CardListReply.FromString,
22             )
```

Кодогенерация

A man in a red shirt and headset is looking intently at a screen. In the background, other people are working in what appears to be a call center or office environment.

Когда сгенерировал клиенты и уже пишешь бизнес логику

Кодогенерация

01

Клиент генерируется
отдельной джобой
в каждом сервисе

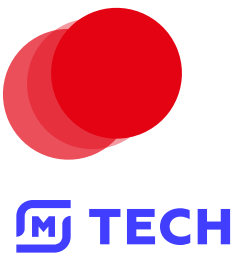
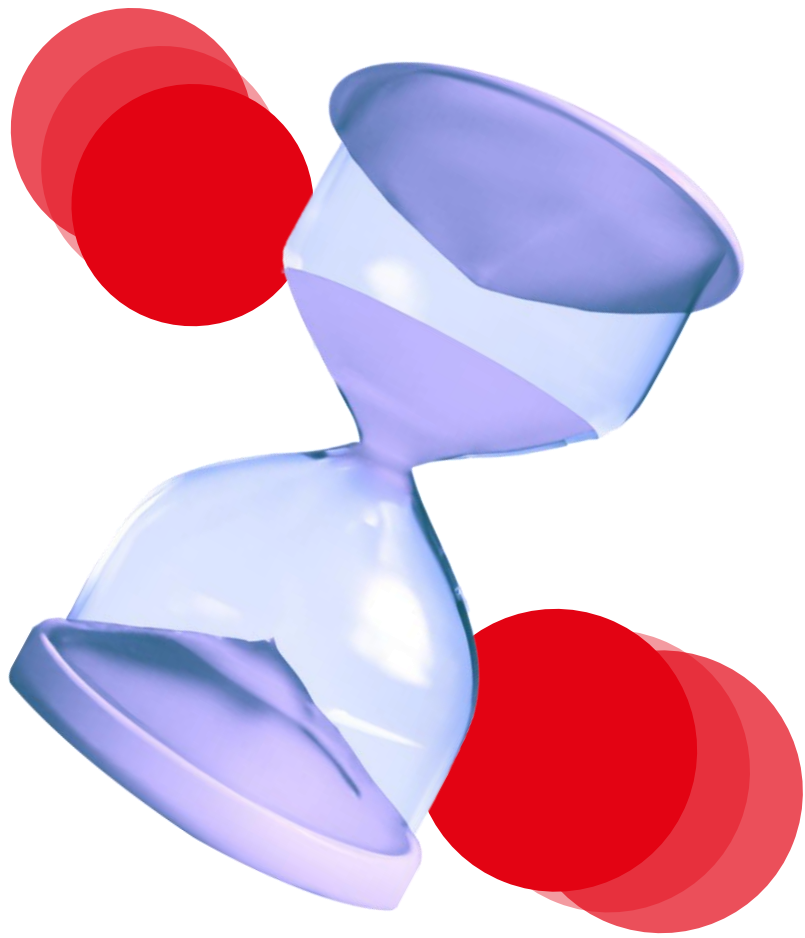
02

Питонисты должны
актуализировать версии
клиента самостоятельно

03




Питонисты должны
актуализировать версии
генератора
самостоятельно

Кодогенерация



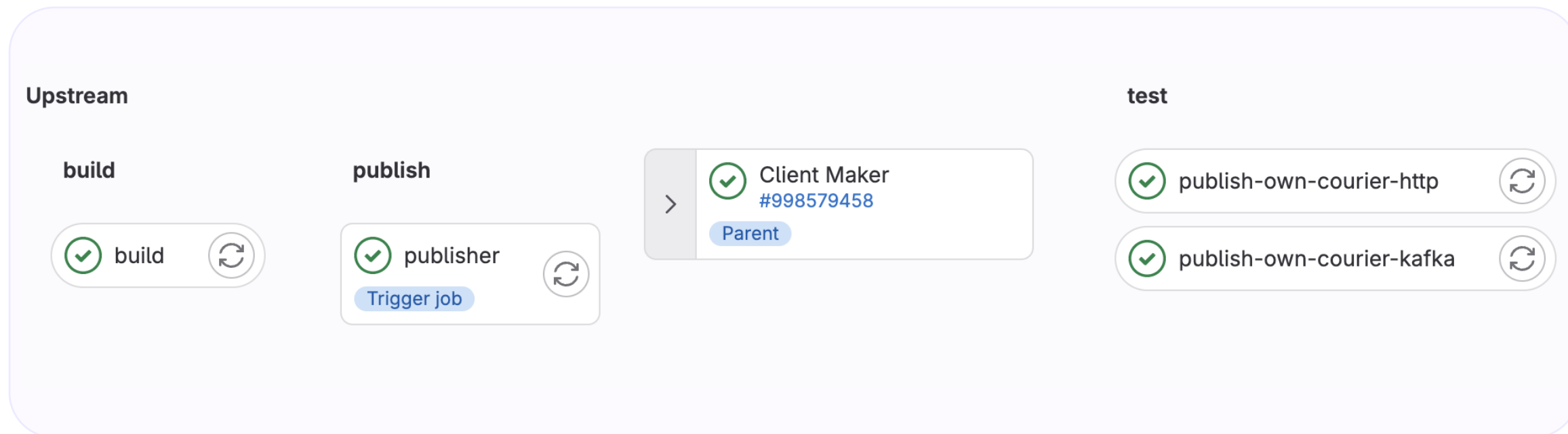
Spec Storage

Хранилище спецификаций всех сервисов в s3

<input type="checkbox"/>	▲ Name	Last Modified
<input type="checkbox"/>	 grpc.zip	Thu, Sep 14 2023 13:09 (GMT+3)
<input type="checkbox"/>	 kafka.zip	Thu, Sep 14 2023 13:09 (GMT+3)
<input type="checkbox"/>	 swagger.yaml	Thu, Sep 14 2023 13:09 (GMT+3)

Динамические gitlab пайплайны

Подвид parent-child пайплайна, когда parent джоба генерирует конфигурацию для дочерних стадий



Динамические gitlab пайплайны












```
generate-config:  
  stage: build  
  script: generate-ci-config > generated-config.yml  
  artifacts:  
    paths:  
      - generated-config.yml
```

```
child-pipeline:  
  stage: test  
  trigger:  
    include:  
      - artifact: generated-config.yml  
        job: generate-config
```

Преимущества

- 01 Централизованность
- 02 Легко расширять/поддерживать
- 03 Публикация по расписанию/
ручная публикация

test

-  publish-ab-grpc
-  publish-ab-http
-  publish-arm-http
-  publish-assembly-routes-http
-  publish-audit-grpc
-  publish-audit-kafka
-  publish-auth-v2-grpc
-  publish-auth-v2-http
-  publish-auth-v2-kafka

Нечитаемые версии



```
service-a-client = "3.4031961938"  
service-b-client = "3.4031961945"  
service-c-client = "3.4031961953"  
service-d-client = "3.4988184347"  
grpc-service-a-client = "2.4411691886"  
kafka-service-a = "2.4650508290"
```

Ограничения гитлаба

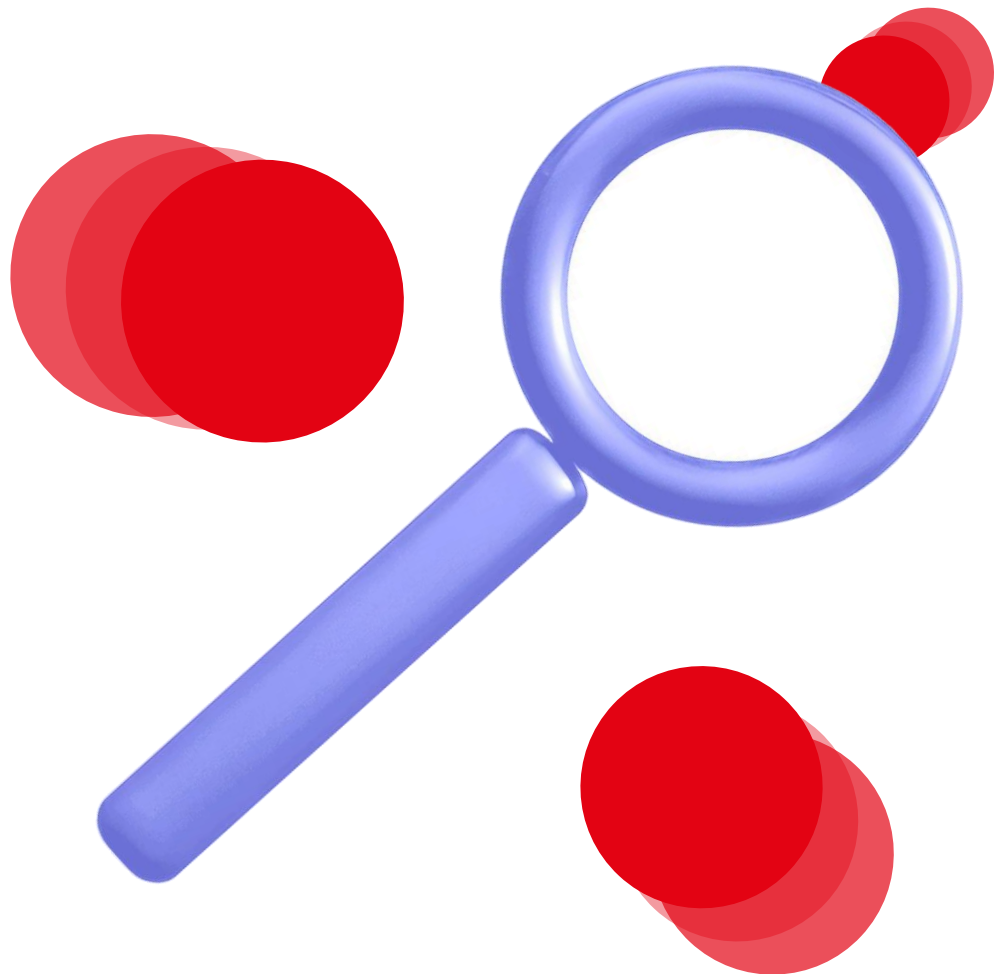


Unable to create pipeline

- ``publish-client-job.yml``: The parsed YAML is too big

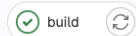
[Go to the pipeline editor](#)

Сложно найти сервис

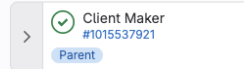
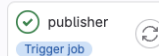


Upstream

























































build



publish



test

-  publish-ab-grpc 
-  publish-ab-http 
-  publish-arm-http 
-  publish-assembly-routes-http 
-  publish-audit-grpc 
-  publish-audit-kafka 
-  publish-auth-v2-grpc 
-  publish-auth-v2-http 
-  publish-auth-v2-kafka 
-  publish-auto-assign-service-http 
-  publish-balance-http 
-  publish-cards-v2-grpc 
-  publish-cards-v2-http 
-  publish-carts-grpc 
-  publish-carts-http 
-  publish-catalog-bff-grpc 
-  publish-catalog-bff-http 
-  publish-checkout-http 
-  publish-ci-admin-http 
-  publish-cli-executor-grpc 
-  publish-clubs-http 
-  publish-clubs-kafka 
-  publish-cms-http 
-  publish-communication-service-grpc 
-  publish-communication-service-http 
-  publish-configurator-api-http 
-  publish-counters-http 
-  publish-coupons-api-http 

Что можно улучшить

01

Версии

02

Поиском необходимой джобы для публикации/ последней завершенной джобы

03

Не публиковать дубликаты клиентов

Альтернативы

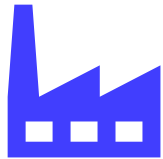
01

Утилита для генерации только необходимых методов в каждом отдельном сервиса

02

Публикация клиентов на отдельной стадии в ci-template

Выводы



Спецификации должны
быть в каждом сервисе,
если проект сложнее
«Hello world»



Кодогенерация
убирает лишнюю
рутину

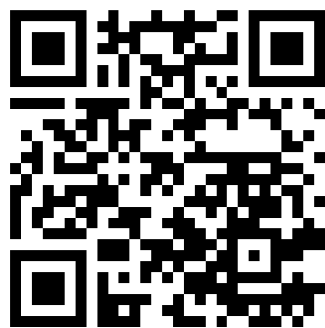


Как её удобно
завернуть – решение
каждой команды

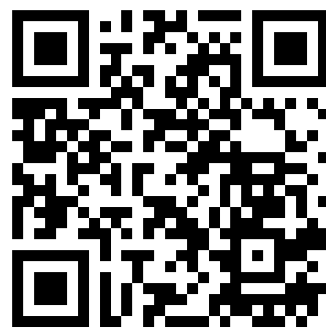
ССЫЛКИ



[Magnit Online](#)



[Pythogen](#)



[Pyprotogen](#)



[Подкаст](#) - как мы работаем
с автосгенеренным кодом



Спасибо за внимание!

Сергей Солдатов
soldatov_sr@magnit.ru