

avito.tech

Москва — 2021

РАЗРАБОТКА СЕРВИСОВ

В АВИТО

Павел Агалецкий

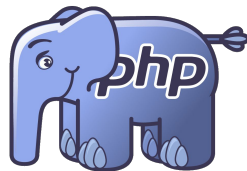
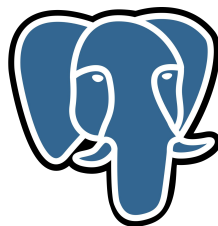
старший разработчик @ PaaS

О ЧЁМ ДОКЛАД

- Локальная разработка
- Создание сервиса
- Запуск сервиса
- Тесты
- Линтеры
- Документирование
- Кодогенерация

О КОМАНДЕ

ПЛАТФОРМА АВИТО



КОМАНДА DEV

- Удобство разработки
- Интерфейсы и инструменты для взаимодействия с платформой
- Базовые общие сервисы
- Библиотеки, бойлерплейты, best-practice

ЛОКАЛЬНАЯ РАЗРАБОТКА

УТИЛИТА AVITO

```
> avito
```

Предоставляет весь необходимый тулинг для работы разработчиков Avito

Usage:

avito [command]

Available Commands:

completion	generate the autocompletion script for the specified shell
config	получить текущий конфиг в виде yaml, toml или json файла
databus	Функционал для работы с DataBus. См. документацию: http://links.k.avito.ru/Eq
delete	Удаляет локальную среду разработки
docs	Работа с документацией. Подробнее https://docs.k.avito.ru/
fmt	Форматирует код или файлы известных форматов
help	Help about any command
hosts	Выводит или сохраняет переданный список хостов в файл hosts
install	установка зависимостей
kubectl	команды упрощающие работу с kubectl
lib	Функционал для работы с библиотеками
lint	Запускает линтеры для проекта
login	Аутентификация в Авито-платформе
queues	Функционал для работы с Queues. См. документацию: http://links.k.avito.ru/Eq
sampler	работа с инструментами сэмплирования баз данных
service	Функционал для работы с сервисами - создание, локальная разработка, деплой и прочее
start	Запускает локальную dev среду на базе миникуба
stop	Останавливает локальную среду разработки

УСТАНОВКА

```
bash -c "$(curl -fsL http://...../install.sh)"
```

AVITO

==> Информация:

Утилита avito предоставляет разработчикам единый консольный интерфейс для работы с платформой Avito.

Документация:

<https://...../paas/start/>

Выполнить установку? [Y/n]:

СОЗДАНИЕ СЕРВИСА

СОЗДАНИЕ СЕРВИСА



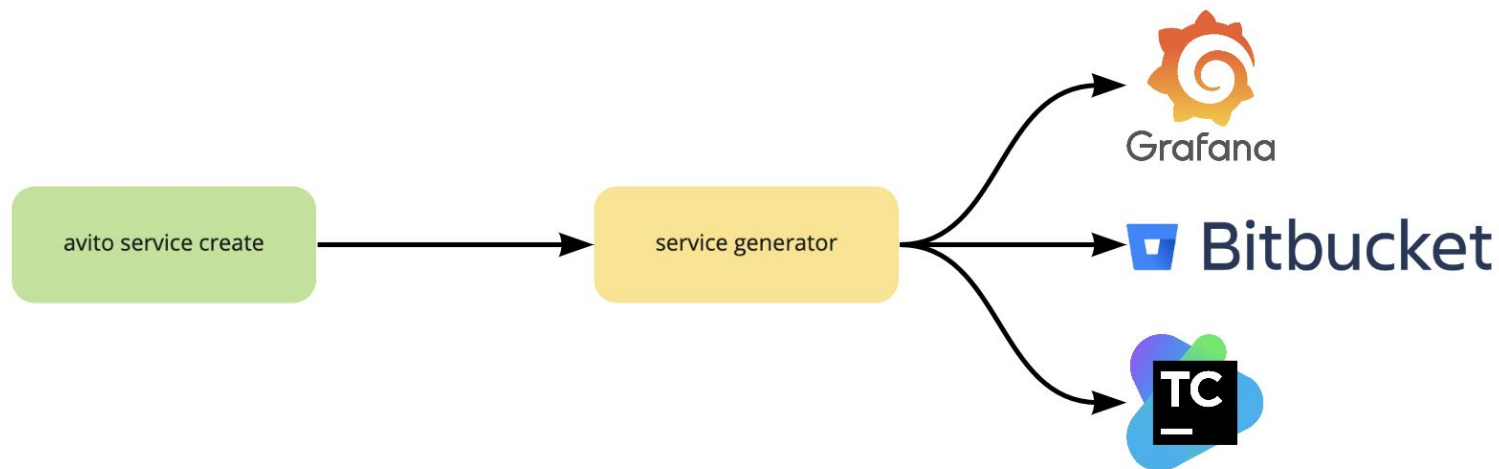
```
> avito service create
```

СОЗДАНИЕ СЕРВИСА

```
→ projects avito service create
👉 Обновление токена доступа пользователя

Выберите шаблон:
1 | Go | Backend |
2 | PHP | Backend |
3 | PHP (MAPI) | Backend |
4 | PHP (Laravel) | Backend |
5 | Python | Backend |
6 | Python (Flask) | Backend | DEPRECATED: используйте Python
7 | Python (Avio) | Backend |
8 | Python (Avio, GPU) | Backend |
9 | JS (Static) | Frontend |
10 | JS (Vue.js) | Frontend |
11 | JS (React.js) | Frontend |
12 | JS (NodeJS, React.js) | Fullstack |
13 | General | 3rd party | Запросите доступ в #paas
Шаблон [1]:
```

СОЗДАНИЕ СЕРВИСА



СОЗДАНИЕ СЕРВИСА

Выберите имя сервиса: `paas-meetup-test-1`

Введите краткое описание сервиса: Сервис для проверки работы

Выберите назначение сервиса:

- | | | |
|---------------------------------|-------------------------|---|
| 1 <code>business</code> | Всё что для прода Avito | Пмиер: <code>user-profile</code> , <code>email-sender</code> , <code>clickstream</code> |
| 2 <code>infrastructure</code> | Внутренние сервисы | Пример: <code>resource-manager</code> , <code>kitchen</code> |
| 3 <code>sandbox</code> | Просто посмотреть ;) | Сервис будет удалён через 30 дней |

Назначение: 3

Выбрано назначение `sandbox`

Создание сервиса `paas-meetup-test-1` из шаблона `Go` с целью `sandbox` будет произведено через 6 сек. (Ctrl+C для отмены)

Создание сервиса `paas-meetup-test-1` из шаблона `Go` с целью `sandbox` будет произведено через 0 сек. (Ctrl+C для отмены)

👉 Идёт создание сервиса...

Перед началом разработки прочитайте `README.md`!

Репозиторий в stash	
Клонировать репозиторий	
Проект в TeamCity	
Проект в Sentry	
Дашборд в Grafana	
Дашборд в PaaS	



СОЗДАНИЕ СЕРВИСА



```
git clone https://...
```

ЗАПУСК СЕРВИСА

ЛОКАЛЬНАЯ СРЕДА



minikube

ЛОКАЛЬНАЯ СРЕДА

- k8s внутри виртуальной машины
- используем
 - a. HyperKit для MacOS
 - b. libvirt для Linux
- похоже на prod

ЛОКАЛЬНАЯ СРЕДА



ЛОКАЛЬНАЯ СРЕДА

```
→ projects avito start
📦 Запускается среда разработки со следующими параметрами:
CPU: 6, Mem: 8192 Mb, Disc 30G
📦 Можно переопределить их через параметры команды
📦 Проверка доступности необходимых зависимостей
📦 Проверка, что локальная среда не запущена
📦 Проверка виртуализации
📦 Проверка ssh
📦 Проверка рантайма
📦 Запуск minikube
📦 Настройка minikube...
📦 Настройка nfs
📦 Введите пароль для изменения файла /etc/exports (sudo)
Password:
📦 Обновление правил nat
No ALTQ support in kernel
ALTQ related functions disabled
nat cleared
pfctl: Use of -f option, could result in flushing of rules
present in the main ruleset added by the system at startup.
See /etc/pf.conf for further details.

No ALTQ support in kernel
ALTQ related functions disabled
nat on utun2 inet proto tcp from 192.168.64.0/24 to any → 10.9.244.43
nat on utun2 inet proto udp from 192.168.64.0/24 to any → 10.9.244.43
nat on utun2 inet proto icmp from 192.168.64.0/24 to any → 10.9.244.43
📦 Проверка доступности vpn
📦 Инициализация сертификатов
📦 Установка секрета с сертификатами для tls
📦 Обновление файла /etc/hosts
📦 Обновления файла hosts выполнено успешно
📦 Среда разработки запущена и готова к работе

Вы можете развернуть свой сервис, выполнив команду в директории проекта
avito service run

Для развертывания монолита выполните
avito provision
→ projects
```

ЗАПУСК СЕРВИСА



```
avito service run
```

```
avito service debug
```

AVITO SERVICE RUN

- запускает сам сервис и его зависимости
 - a. базы данных
 - b. очереди
 - c. ...
- не требует локального инструментария
- менее удобно выполнять отладку

AVITO SERVICE DEBUG

- запускает только зависимости
 - а. базы данных
 - б. очереди
 - с. ...
- нужен локальный инструментарий
- можно запускать worker, cron, ...

AVITO SERVICE DEBUG

```
+ service-paas-meetup-test-1 git:(master) avito service debug
👉 Валидация файлов и аргументов
👉 Добавление игнорируемых файлов в .gitignore
👉 Генерация манифестов деплоя
👉 Сборка контейнера миграций
👉 Деплой зависимостей сервиса
🚀 Сервис готов к запуску 🚀

Настройка:
- Сервис:
  - Если вы используете "gl/app-boilerplate" обновите его до 1.1.0
- Кроны/воркеры/интеграционные тесты (все что не на "app-boilerplate"):
  - Должны вызвать метод "dotenv.Load()" из пакета "gl/dotenv" на самой первой строке точки входа, е.г.:
    - функция "main()"
    - функция "init()"
    - функция "Test_Anything" (в файле с тегом // +build integration)

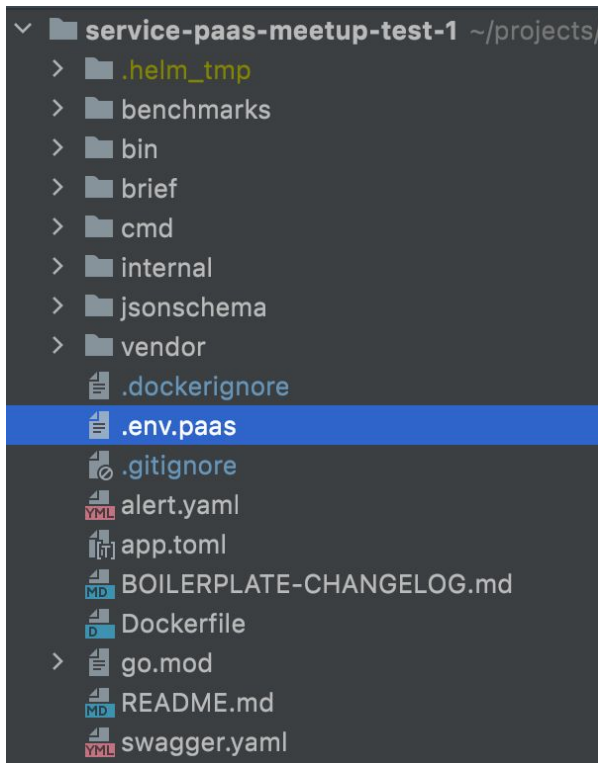
Запускать сервис/воркеры/скрипты можно любым удобным способом: из консоли или в IDE.
Примеры запуска:
- Любое приложение: `go run cmd/service/main.go`
- Любые тесты: `go test -v -tags integration ./...`
- Кнопка `RunStart/Debug` в любой IDE, запустит любую исполняемую функцию, не только main,
  а даже 1 тест, который ходит в базу.
- dlv и прочие раннеры
- Expert-mode для сложных кейсов: docker run --some_args $(docker build -q .) your_cmd (базы будут доступны,
  некоторые переменные придется переопределить)

Notice: Для запуска e2e тестов, нужно поднять сервис в одном терминале, а в другом запускать.
Адрес сервиса в переменной "SERVICE_URL".

Базы, моки и т.д. задеплоены, переменные окружения будут доступны внутри приложения.
Посмотреть конекты к базам: `avito service connect`
Посмотреть переменные окружения: `cat .env.paas`
Переменные окружения можно переопределять любыми способами. е.г "MY_VAR=111 my_app"

При изменении app.toml или миграций нужно нажать Ctrl+C, затем запустить эту команду заново.
```

AVITO SERVICE DEBUG

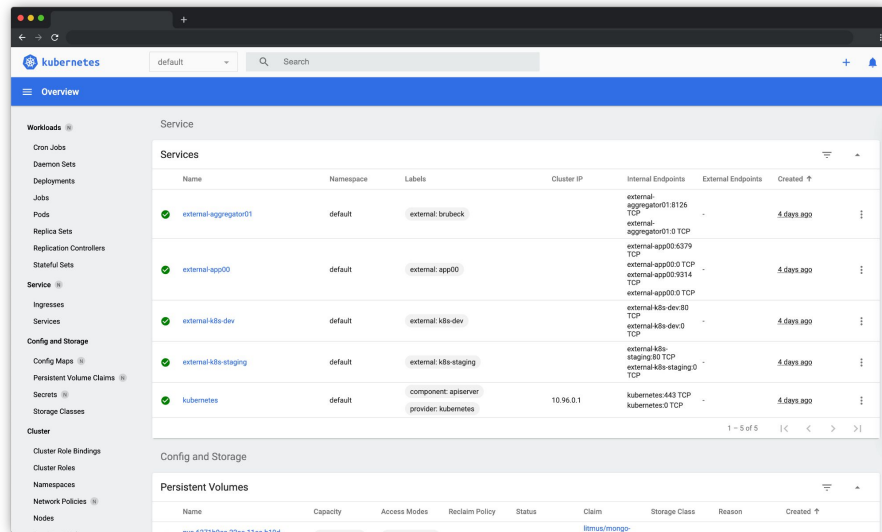


AVITO SERVICE DEBUG



MINIKUBE DASHBOARD

```
> minikube dashboard
```



The screenshot shows the Minikube Dashboard interface. The left sidebar contains a navigation menu with categories like Workloads, Service, Config and Storage, and Cluster. The main content area is titled 'Service' and displays a table of Services. The table has columns for Name, Namespace, Labels, Cluster IP, Internal Endpoints, External Endpoints, and Created. Below the Services table, there is a section for 'Config and Storage' which includes a 'Persistent Volumes' table. The Services table lists several services, including 'external-aggregator01', 'external-app00', 'external-k8s-dev', 'external-k8s-staging', and 'kubernetes'. The Persistent Volumes table is partially visible at the bottom.

Name	Namespace	Labels	Cluster IP	Internal Endpoints	External Endpoints	Created
external-aggregator01	default	external: brubeck		external-aggregator01:8126 TCP	-	4 days ago
external-app00	default	external: app00		external-app00:6379 TCP external-app00:0 TCP external-app00:9314 TCP	-	4 days ago
external-k8s-dev	default	external: k8s-dev		external-k8s-dev:80 TCP external-k8s-dev:0 TCP	-	4 days ago
external-k8s-staging	default	external: k8s-staging		external-k8s-staging:80 TCP external-k8s-staging:0 TCP	-	4 days ago
kubernetes	default	component: apiserver provider: kubernetes	10.96.0.1	kubernetes:443 TCP kubernetes:0 TCP	-	4 days ago

Name	Capacity	Access Modes	Reclaim Policy	Status	Claim	Storage Class	Reason	Created
aws-efs2018m-20m-11m-8104					bitnami/mongo-			

КАК ЭТО РАБОТАЕТ: APP.TOML



```
tree -L 1
.
├── Dockerfile
├── ...
├── app.toml
├── ...
```

КАК ЭТО РАБОТАЕТ: APP.TOML

```
name = "paas-meetup-test-1"
description = "Сервис для проверки работы"
kind = "sandbox"
replicas = 1

[engine]
name = "golang"
version = "1.16"
size = "small"

[envs.prod]
replicas = 1

[[dependencies]]
name = "wordcounter"
```

КАК ЭТО РАБОТАЕТ: DOCKERFILE

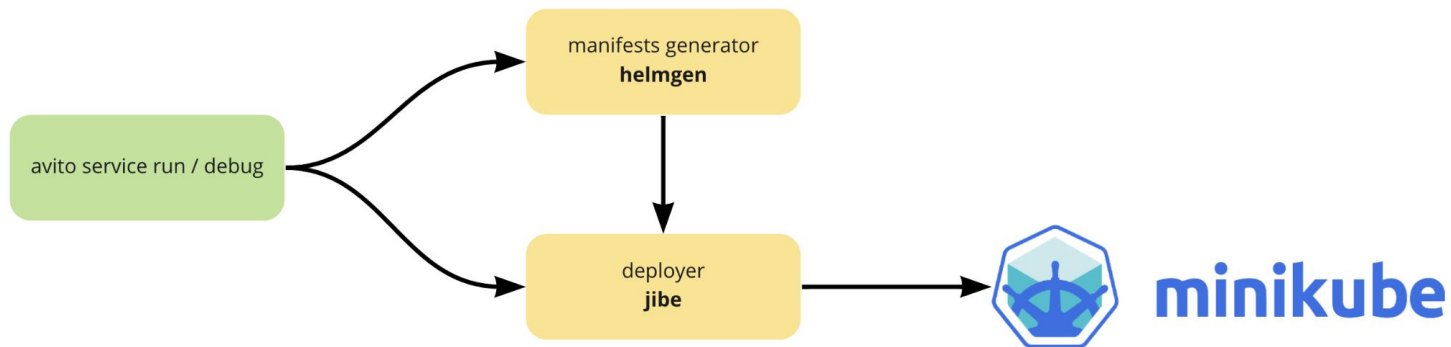


```
FROM registry.name/service-golang:1.16

WORKDIR ${GOPATH}/src/go.avito.ru/av/service-paas-meetup-test-1/
COPY . ${GOPATH}/src/go.avito.ru/av/service-paas-meetup-test-1/


RUN go build -o ${GOPATH}/bin/service-entrpoint ./cmd/service
```


КАК ЭТО РАБОТАЕТ




ТЯЖЕЛОВАТО...

```
→ projects avito start
```

 Запускается среда разработки со следующими параметрами:

 CPU: 6, Mem: 8192 Mb, Disc 30G

 Можно переопределить их через параметры команды

ЗАВИСИМОСТИ В ОБЛАКЕ



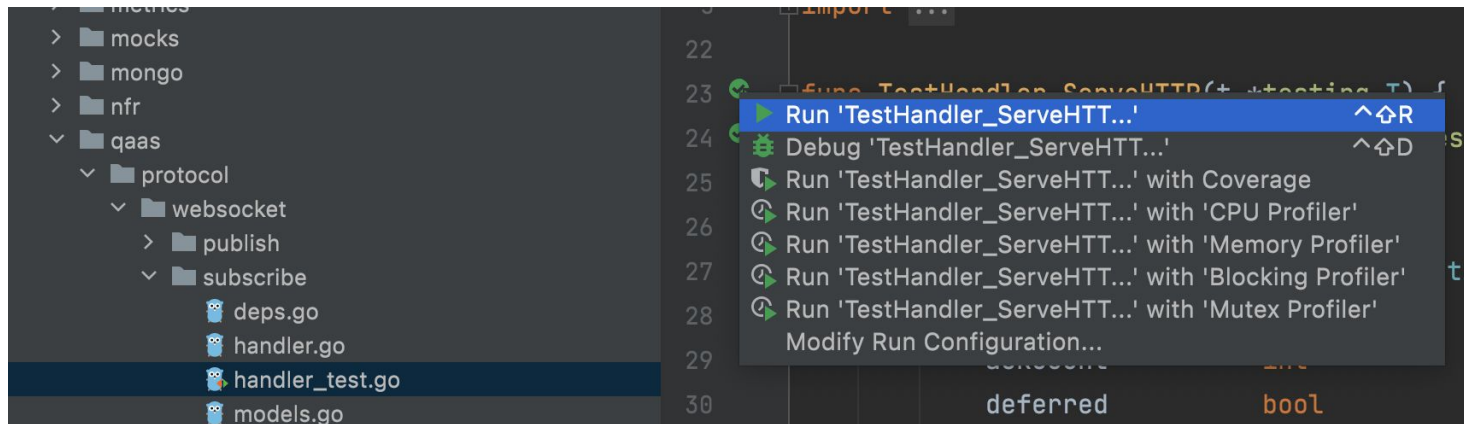
```
avito service debug --cloud
```


ЗАВИСИМОСТИ В ОБЛАКЕ

- не нужен minikube
 - а. можно не выполнять avito start
- все зависимости поднимаются во внешнем кластере
- нужен только локальный Docker
 - а. Docker for MacOS
 - б. native Linux
- **нам проще помочь :)**

ТЕСТЫ

UNIT TESTS



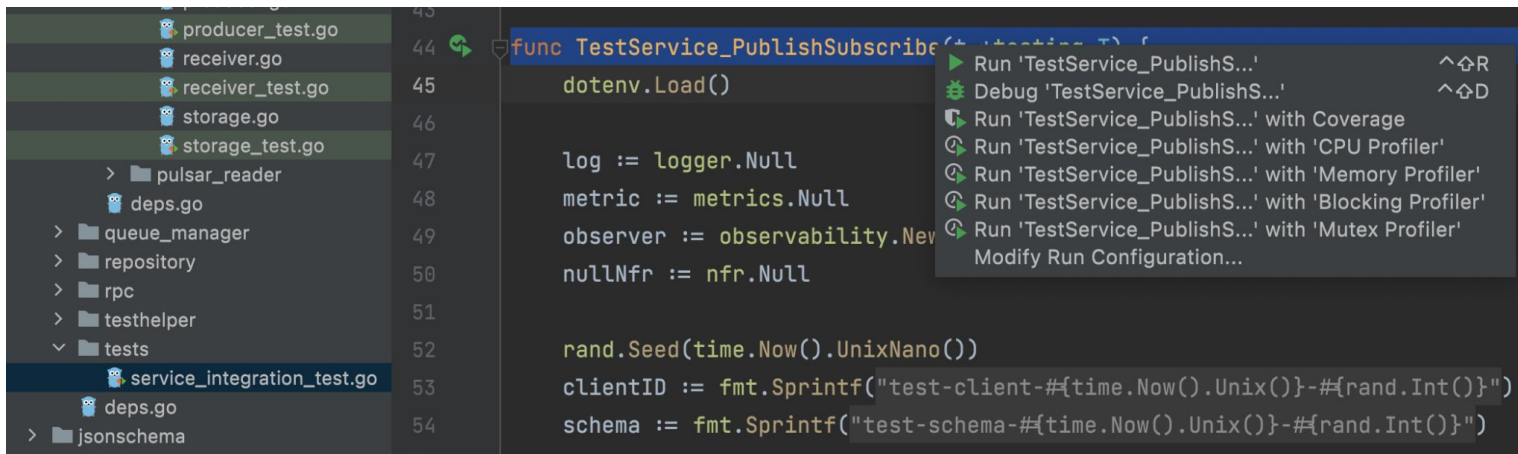
INTEGRATION TESTS



```
avito service debug --test
```

```
avito service debug --cloud --test
```

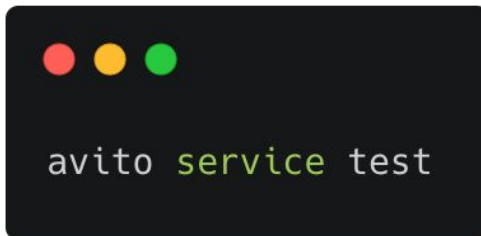
INTEGRATION TESTS



```
43  
44 func TestService_PublishSubscribe(t *testing.T) {  
45     dotenv.Load()  
46  
47     log := logger.Null  
48     metric := metrics.Null  
49     observer := observability.NewNullObserver()  
50     nullNfr := nfr.Null  
51  
52     rand.Seed(time.Now().UnixNano())  
53     clientID := fmt.Sprintf("test-client-#{time.Now().Unix()}-#{rand.Int()}")  
54     schema := fmt.Sprintf("test-schema-#{time.Now().Unix()}-#{rand.Int()}")  
55 }
```

- Run 'TestService_PublishS...'
- Debug 'TestService_PublishS...'
- Run 'TestService_PublishS...' with Coverage
- Run 'TestService_PublishS...' with 'CPU Profiler'
- Run 'TestService_PublishS...' with 'Memory Profiler'
- Run 'TestService_PublishS...' with 'Blocking Profiler'
- Run 'TestService_PublishS...' with 'Mutex Profiler'
- Modify Run Configuration...

INTEGRATION TESTS



INTEGRATION TESTS

```
go.avito.ru/av/service-queues/internal/rpc/create_queues/create_queues_mock.go:38:      Handle      100.0%
go.avito.ru/av/service-queues/internal/rpc/create_queues/create_queues_mock.go:46:      Handle      100.0%
go.avito.ru/av/service-queues/internal/rpc/create_queues/handler.go:17:              New          100.0%
go.avito.ru/av/service-queues/internal/rpc/create_queues/handler.go:26:              Handle      85.7%
go.avito.ru/av/service-queues/internal/testhelper/db.go:16:                       NewDB        100.0%
total:                                       (statements) 84.7%
Tests passed
```

AVITO SERVICE TEST

- Запускает всё в minikube
- Не требует локального инструментария
- Сложнее отлаживать тесты

ЛИНТЕРЫ

ЛИНТИНГ И ФОРМАТИРОВАНИЕ



```
avito fmt
```

```
avito lint
```

ЛИНТИНГ И ФОРМАТИРОВАНИЕ

```
→ service-paas-meetup-test-1 git:(master) x avito lint
🐼 golangci-lint run
👉 Получение новой версии quality gates

INFO [config_reader] Config search paths: [./ /go/src/go.avito.ru/av/service-paas-meetup-test-1 /go/src/go.avito.ru/av /go/src/go.avito.ru /go/src /go / /root]
INFO [lintersdb] Active 10 linters: [deadcode errcheck gosimple govet ineffassign staticcheck structcheck typecheck unused varcheck]
INFO [loader] Go packages loading at mode 575 (compiled_files|exports_file|files|name|deps|imports|types_sizes) took 28.062005127s
INFO [runner/filename_unadjuster] Pre-built 0 adjustments in 2.508203ms
INFO [linters context/goanalysis] analyzers took 10.765075314s with top 10 stages: builddir: 7.270837931s, inspect: 608.740865ms, fact_deprecated: 402.84074ms, fact_purity: 298.472549ms, printf: 279.709127ms, nilness: 271.391347ms, ctrlflow: 230.82246ms, SA5012: 183.427254ms, typedness: 179.269098ms, S1010: 24.869748ms
INFO [linters context/goanalysis] analyzers took 110.453552ms with top 10 stages: builddir: 79.312458ms, directives: 14.650886ms, isgenerated: 13.495074ms, unused: 2.995134ms
INFO [runner] processing took 2.899µs with stages: max_same_issues: 759ns, nolint: 279ns, skip_dirs: 267ns, max_from_linter: 197ns, filename_unadjuster: 182ns, cgo: 136ns, identifier_marker: 118ns, autogenerated_exclude: 117ns, uniq_by_line: 117ns, skip_files: 116ns, diff: 115ns, path_prettifier: 112ns, path_shortener: 52ns, exclude: 50ns, source_code: 49ns, sort_results: 49ns, max_per_file_from_linter: 48ns, severity_rules: 47ns, path_prefixer: 45ns, exclude_rules: 44ns
INFO [runner] linters took 4.006506328s with stages: goanalysis_metalinter: 3.815998909s, unused: 190.332182ms
INFO File cache stats: 0 entries of total size 0B
INFO Memory: 323 samples, avg is 96.7MB, max is 342.1MB
INFO Execution took 32.119077909s
🐼 security check
👉 Если будут найдены проблемы безопасности, отчёт о них придёт в Slack в течение пары минут
→ service-paas-meetup-test-1 git:(master) x
```

ЛИНТИНГ И ФОРМАТИРОВАНИЕ

- настраивается через стандартные для языков конфиги
 - a. `.golangci.yml`
 - b. `.php-cs-fixer.dist.php`
 - c. ...

ДОКУМЕНТИРОВАНИЕ

ДОКУМЕНТИРОВАНИЕ

avito docs add



Создание конфигурационных файлов



Успешно создан лэйаут для работы с DaaS на основе MkDocs. Подробнее см.
<https://www.mkdocs.org/#getting-started>

Руководство по написанию документации - <https://paas/daas/>

ДОКУМЕНТИРОВАНИЕ

- создаётся конфиг на основе MkDocs
- документация в Markdown
- лежит в репозитории сервиса

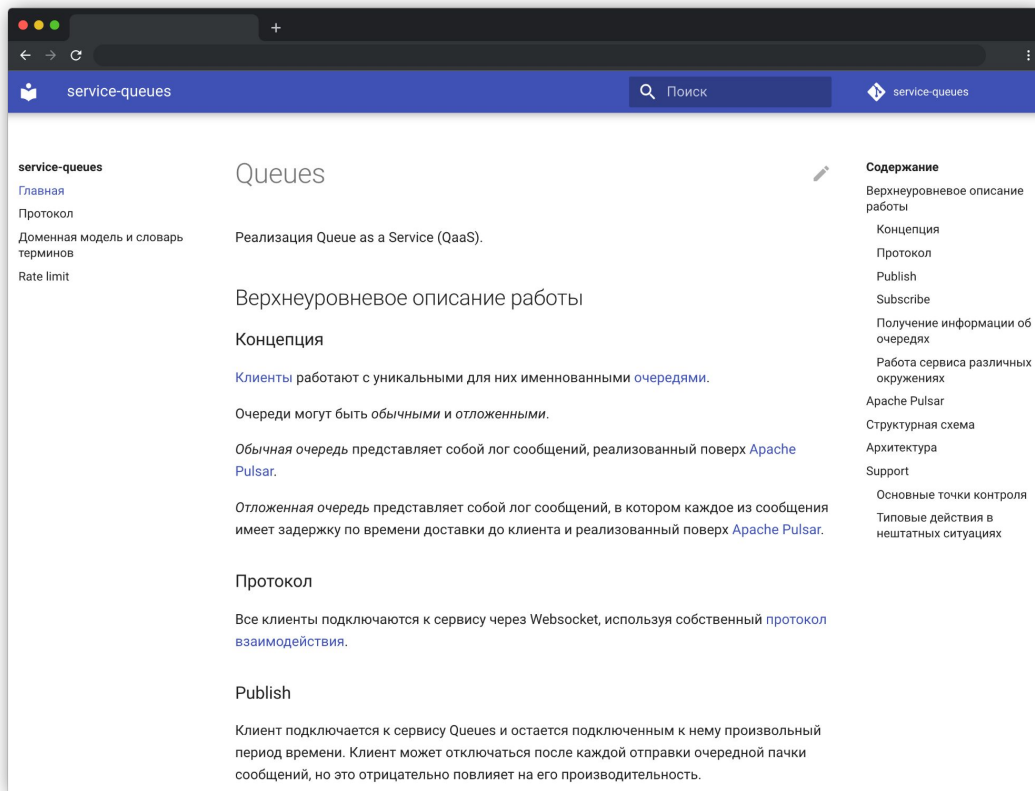
ДОКУМЕНТИРОВАНИЕ



avito docs dev

- 👉 Получение названия последнего образа DaaS
- 👉 Скачивание DaaS
- 👉 Удаление старых образов DaaS
- 👉 Выполнение проверки документации
- 🤖 Документация доступна по адресу <http://127.0.0.1:8080>

ДОКУМЕНТИРОВАНИЕ



КОДОГЕНЕРАЦИЯ

КОДОГЕНЕРАЦИЯ



```
avito service codegen
```

```
Генерирую код...
```

```
Валидация схем успешно пройдена. Ошибок не обнаружено.
```

```
Сгенерирован вспомогательный код для сервиса
```

```
Сгенерирован swagger.yaml для сервиса
```

```
Добавлены и сгенерированы зависимости:
```

```
- wordcounter
```

КОДОГЕНЕРАЦИЯ

```
service "paas-meetup-test-1"

idempotent rpc sum (SumIn) SumOut `A sum method`
rpc countWords (CountWordsIn) CountWordsOut `Counts words in text`

message SumIn {
    a    int    `A first number`
    b    int    `A second number`
}

message SumOut {
    sum    int    `A sum of the numbers`
}

message CountWordsIn {
    text string
}

message CountWordsOut {
    words int
}
```

КОДОГЕНЕРАЦИЯ

```
// Code generated - DO NOT EDIT.
package wordcounter

import (
    "context"

    rpcprotocol "go.avito.ru/gl/rpc-protocol"
)

const Name = "wordcounter"

type Client interface {
    Count(context.Context, *CountIn) (*CountOut, error)
}

type client struct {
    count func(ctx context.Context, in interface{}) (out interface{}, err error)
}

func New(c rpcprotocol.Client) Client {
    return &client{
        count: c.Method("count", false, (*CountOut)(nil), nil, nil),
    }
}

func (c *client) Count(ctx context.Context, in *CountIn) (*CountOut, error) {
    o, err := c.count(ctx, in)
    out, _ := o.(*CountOut)
    return out, err
}
```

ОСТАЛЬНЫЕ ВОЗМОЖНОСТИ

AVITO CLI ...

- Сэмплирование баз данных
 - а. `avito sample ...`
- Управление зависимостями
 - а. `avito dependency ...`
- Развёртывание монолита
 - а. `avito provision`
- Работа с k8s
 - а. `avito kubectl ...`

avito.tech

Москва — 2021

Павел Агалецкий

Старший инженер



psagaletskiy@avito.ru

Спасибо!