

≡

○

└

○

|

+

|

T1

ASYNCAPI ДЛЯ ОПИСАНИЯ АСИНХРОННЫХ ИНТЕГРАЦИЙ

Дмитрий Уланов.
НОТА. Входит в Холдинг T1
Telegram: @dimonier

+ | T1



Дмитрий Уланов

Telegram: @dimonier
<https://github.com/dimonier>

НОТА. Входит в Холдинг Т1
dulanov@nota.tech



В системном анализе 7 лет

Активности: бизнес- и системный анализ,
менторинг, участие в найме

Навыки: модели данных, интеграции, SQL, сети,
PlantUML, Obsidian, Python

Достижения: продукты, спроектированные с
нуля и работающие от 0 до 5+ лет

Интересы: экзомозг, телеграм-боты, управление
собой и другими

ПЛАН ДОКЛАДА

Где используются асинхронные интеграции

01

Что такое AsyncAPI?

02

Ключевые преимущества и особенности

03

Сравнение с другими форматами описания интеграций

04

Примеры использования

05

Личный опыт

06

Инструменты, поддерживающие AsyncAPI

07

Последние изменения

08

План развития

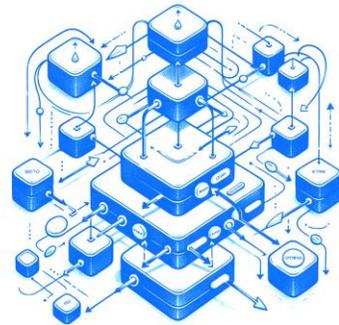
09

Вопросы и ответы

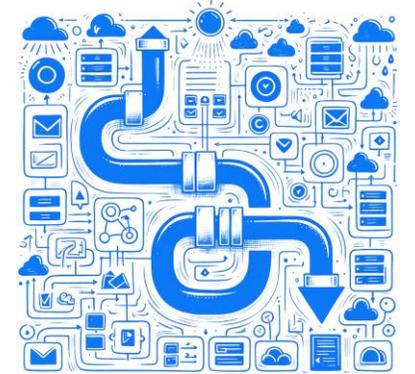
10

ГДЕ ИСПОЛЬЗУЮТСЯ АСИНХРОННЫЕ ИНТЕГРАЦИИ

Микросервисные архитектуры



Событийно-ориентированные системы



Распределенные системы



Интернет вещей (IoT)



ЧТО ТАКОЕ ASYNCAPI?

Спецификация, предназначенная для упрощения

создания

документирования

управления

использования

асинхронных API

Направлена на решение проблем, связанных с асинхронными коммуникациями, аналогично тому, как OpenAPI (ранее известный как Swagger) делает это для синхронных HTTP-взаимодействий



ЗАЧЕМ СПЕЦИФИКАЦИЯ?



Унификация описания контрактов



Поддержка различных протоколов



Интеграция с инструментами разработки



Тестирование и валидация



Генерация кода

КЛЮЧЕВЫЕ ПРЕИМУЩЕСТВА ASYNCAPI



НЕ ПРИВЯЗЫВАЕТСЯ К КОНКРЕТНОМУ ПРОТОКОЛУ

Подходит для:

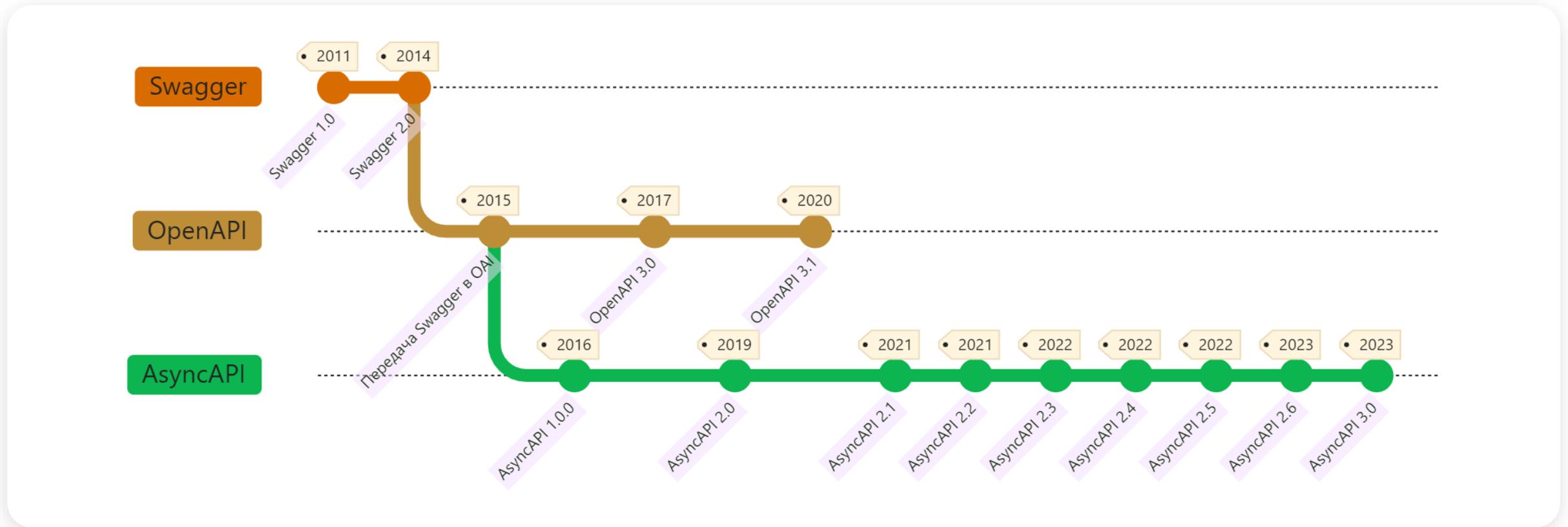


HTTP

и других

АКТИВНО РАЗВИВАЕТСЯ

История развития спецификации AsyncAPI на фоне Swagger/OpenAPI



НЕДОСТАТКИ ASYNCAPI



НЕ ДАЁТ ВОЗМОЖНОСТИ ОТПРАВИТЬ ЗАПРОС (КАК SWAGGER UI)

RECEIVE smartylighting.streetlights.1.0.event.{streetlightId}.lighting.measured

The topic on which measured values may be produced and consumed.
Inform about environmental lighting conditions of a particular streetlight.

Operation ID `receiveLightMeasurement`

Available only on servers:
`scram-connections` `mtls-connections`

Parameters > Expand all **Object**

Operation specific information **KAFKA** > Expand all **Object**

Accepts the following message:

Light measured `lightMeasured`
Inform about environmental lighting conditions of a particular streetlight.

APPLICATION/JSON

Message ID `lightMeasured`

Payload > Expand all **Object**

Headers > Expand all **Object**

Examples

- Payload >
- Headers >

POST /authorize

Handled by our API.

Parameters Cancel

No parameters

Execute **Clear**

Responses

Curl

```
curl -X 'POST' \
  'https://api.clever-cloud.com/v2/authorize' \
  -H 'accept: */*' \
  -d ''
```

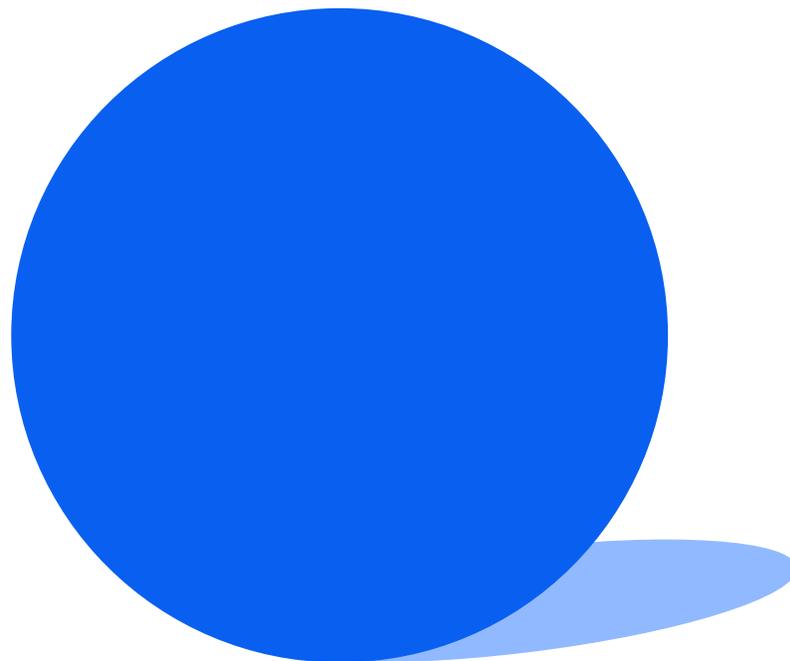
Request URL

```
https://api.clever-cloud.com/v2/authorize
```

Server response

Code **Details**

НЕ ОЧЕНЬ ПОПУЛЯРЕН (ПОКА?)



OpenAPI / Swagger



AsyncAPI

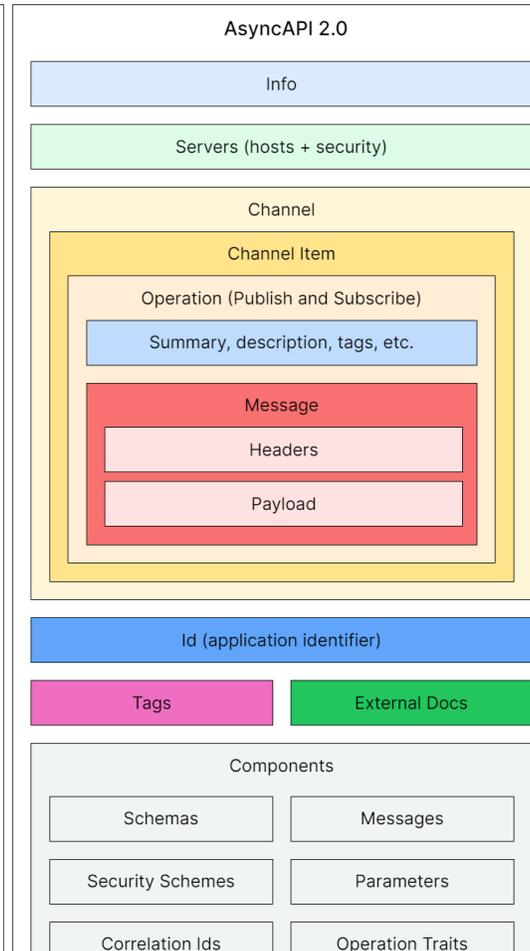
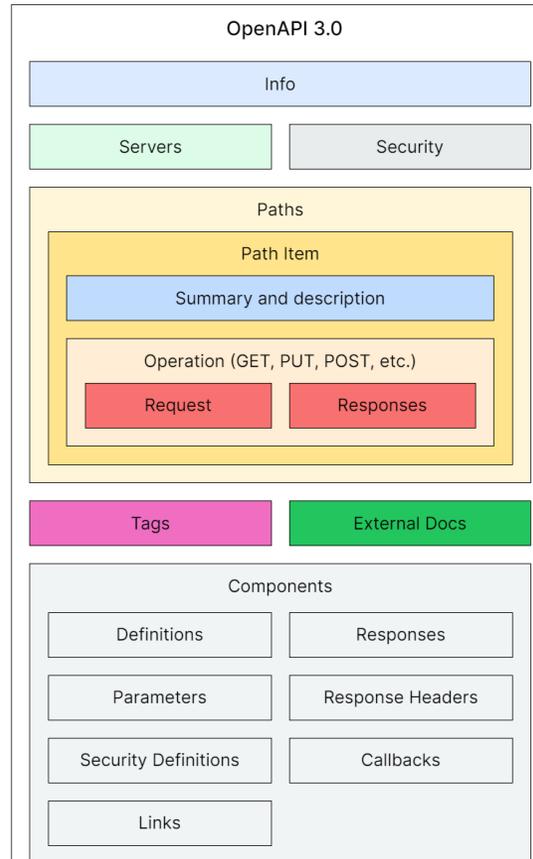
ОБЛАСТИ ПРИМЕНИМОСТИ ASYNC API

Характеристика	AsyncAPI	OpenAPI
Основное назначение	Описание асинхронных API, таких как IoT, микросервисы, событийно-ориентированные системы	Описание RESTful API, включая эндпоинты, операции, параметры запросов и тела ответов
Тип взаимодействия	Асинхронное	Синхронное
Поддерживаемые протоколы	MQTT, WebSocket, AMQP, Kafka, HTTP и др.	HTTP, HTTPS
Фокус	На обмене сообщениями и событийной модели	На операциях и ресурсах
Автоматическая генерация кода и прочего из спецификации	Да, поддерживает генерацию кода, документации и тестов для асинхронных API	Да, широкие возможности для генерации серверного и клиентского кода, документации и тестов

СТРУКТУРЫ СПЕЦИФИКАЦИЙ OPENAPI И ASYNCAPI: ОБЗОР

OpenAPI

используется для описания клиент-серверных API (запрос и ответ)



AsyncAPI

используется преимущественно для описания событийно-ориентированных API с использованием брокеров (за исключением Websocket)



СХОДСТВА OPENAPI И ASYNCAPI

Разделы OpenAPI, которые можно переиспользовать в AsyncAPI*	Назначение раздела
info	Описание API, включая версию, название, лицензию и контактную информацию
servers	Описание серверов, с которыми API может взаимодействовать
components/schemas	Определения схем данных, которые могут быть использованы в сообщениях
components/securitySchemes	Определения схем безопасности, применимые для аутентификации и авторизации
tags	Группировка операций для лучшей навигации и организации
externalDocs	Ссылки на внешнюю документацию

* Возможно переиспользование «как есть» либо с минимальными изменениями

РАЗЛИЧИЯ OPENAPI И ASYNCAPI

Аспект	OpenAPI	AsyncAPI	Комментарий
Описание nullable	nullable: true	type: [type, null]	В OpenAPI используется nullable, в AsyncAPI — объединение типов
Множественные типы	Не поддерживаются напрямую	Поддерживаются, например, type: [string, number]	AsyncAPI позволяет указывать множественные типы данных для одного поля
Потоки сообщений	Не применимо	channels для описания потоков сообщений	AsyncAPI использует channels для описания коммуникационных каналов
Операции	paths с поддержкой операций get , post и т.д.	Операции send и receive (pub/sub)	OpenAPI описывает конкретные HTTP операции для взаимодействия
Доступность документации и ресурсов	https://swagger.io/ недоступен из РФ	https://www.asyncapi.com/ доступен из РФ	Ситуация может меняться во времени

ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ ASYNCAPI

StreetLights	https://studio.asyncapi.com/
ZBOS by Zora Robotics	https://apis.guru/asyncapi-directory/html/zora+ZBOS@v2.4.2.html
OpenWeatherMap	https://apis.guru/asyncapi-directory/html/ably-openweathermap+weather@v1.0.0.html

Больше в apitracker.io и [APIs.guru](https://apis.guru)

ЛИЧНЫЙ ОПЫТ

Задача

Описать контракт интеграции продуктов платформы Сфера с общеплатформенным сервисом уведомлений через Kafka



Результаты

Все подробности интеграции в одной спецификации на ~600 строк
Изучил новый инструмент, получил удовольствие 😊



Выводы

Годный инструмент
Нет интерактива, как в Swagger UI
При обновлении спецификации генерация
человекопонятной документации –
дело нескольких минут



ИНСТРУМЕНТЫ, ПОДДЕРЖИВАЮЩИЕ РАБОТУ С ASYNCAPI

Создание, редактирование, валидация спецификации

- <https://studio.asyncapi.com/>
- VS Code + расширение asyncapi-preview
- IDEA + плагин AsyncAPI

Генерация документации (html, markdown, asciidoc)

- AsyncAPI Generator
- AsyncAPI CLI
- Bump.sh
- Springwolf

Генерация кода (typescript, c#, golang, java, javascript, dart, python, rust, kotlin, php, c++)

- AsyncAPI Generator
- AsyncAPI CLI

Генерация моков и тестирование

- Microcks
- Spescmatic
- Virtualan

CI/CD

AsyncAPI GitHub Action

Указаны не все инструменты.

Подробнее на

<https://www.asyncapi.com/tools>

ПОСЛЕДНИЕ ИЗМЕНЕНИЯ ASYNCAPI

В конце 2023 вышла версия 3.0.0 спецификации AsyncAPI:

Изменена структура объектов `channels`, `messages`, `operations`, `payload`, `schema` для большей гибкости и наследования свойств



Свойство `url` разделено на `host` и `pathname`



Понятия `PUB/SUB` заменены более универсальными `SEND/RECEIVE`, внесены другие подобные изменения, чтобы сделать спецификацию интуитивно понятной



Совместимость с `OpenAPI` больше не является приоритетом, т.к. эти стандарты уже довольно сильно разошлись



Текущие планы:

- Обновление инструментария под будущие фичи
- Снижение порога входа, чтобы новичок мог начать описывать API через 20 минут после начала знакомства с AsyncAPI
- Устранение технических препятствий
- Привлечение новых участников в сообщество AsyncAPI
- Возможность встраивания интерактивной документации AsyncAPI в существующую документацию

Сообщество AsyncAPI принимает предложения по развитию спецификации, в том числе документации к ней.

Подведём итоги



КЛЮЧЕВЫЕ АСПЕКТЫ ЗНАЧИМОСТИ ASYNCAPI

Согласованность всех интеграций продукта: всё описано в одной спецификации с использованием одних и тех же схем



Поддержка разных протоколов в одной спецификации



Сходство формата спецификации с OpenAPI / Swagger: сниженный порог входа



Сокращение затрат на разработку, документирование и тестирование (генерация кода и т.п. из спецификации)



Удобство разработки в IDE, хранения и версионирования в git: подход docs-as-code



УРОВНИ ДОКУМЕНТИРОВАНИЯ API

Не документирован,
«по коду всё понятно»



Есть описание
в Confluence



Есть Swagger,
автоматически
сгенерированный
из кода



API first:
спецификация,
по которой
генерируется код
и всё остальное



Н

С

Т

С

|

+

|

Т

СПАСИБО
ЗА ВНИМАНИЕ

+

|

Т