

API КАК ПРОДУКТ В РАЗРАБОТКЕ КОМПЛЕКСНЫХ ИТ-РЕШЕНИЙ НА ПРИМЕРЕ АВТОМАТИЗАЦИИ ЗАКУПОК И УПРАВЛЕНИЯ КРОСС-ПРОДАЖАМИ

Тиунчик Д.Р.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Научный руководитель: Шелест А.В. – магистр техники и технологии, ассистент кафедры ПИКС

Аннотация. В статье рассматривается модель «API как продукт» что позволяет централизовать документацию для всех микросервисов и превратить API в самостоятельное направление развития и источника дохода в B2B-сегменте.

Ключевые слова: API как продукт, OpenAPI, AsyncAPI, микросервисная архитектура.

Введение. Современные компании, ориентированные на цифровую трансформацию, уже не рассматривают *Application Programming Interface (API)* просто как внутренний способ интеграции сервисов. Всё чаще API позиционируют как продукт, создавая новые каналы монетизации и расширяя экосистему партнёров и клиентов. В сфере автоматизации закупок и управления кросс-продажами – продажи сопутствующих основной покупке товаров и услуг – это особенно актуально, так как возникает необходимость непрерывного обмена данными между различными сервисами (например, сборщик тендеров, аналитическая панель, модули кросс-продаж, системы оплаты и учёта). Потенциал такого подхода обсуждался в разных докладах, в том числе в выступлении А. Митина и Д. Смирнова на *T-Meetup: Java*, где подчёркивалась важность грамотного документирования API для повышения эффективности взаимодействия [1].

Основная часть. Основная проблема при проектировании комплексных систем типа API как продукт в том, что контракты¹ между сервисами (описанные через *OpenAPI*, *AsyncAPI* или другие спецификации) разбросаны и часто не синхронизированы. Разработчики и аналитики нередко тратят время на поиск актуальных версий, проверку изменений и согласование форматов запросов. Это приводит к ошибкам при интеграции, росту временных затрат и увеличению технического долга. В ходе реализации проекта по автоматизации закупок и управления кросс-продажами была выявлена проблема: необходимо было иметь все контракты в одном месте и в удобном виде, позволяющем быстро понимать, какие операции поддерживает тот или иной сервис.

Чтобы решить описанную задачу, был внедрен механизм централизованного документирования и генерации кода на основе *OpenAPI* и *AsyncAPI*. В частности, автоматически собираются спецификации от разных микросервисов, проверяется их валидность и актуальность, а затем генерируется код для *Java (backend)* и *React (frontend)*. С помощью этого подхода система получает единый источник, где каждая сущность или метод неизменно отражены в соответствующей спецификации. При любых изменениях в контракте разработчик обновляет саму спецификацию, а затем повторно генерирует код, исключая расхождения между сервисами. При этом было принято решение не передавать исходники в *Swagger* напрямую, а обрабатывать их собственным инструментом, который выдаёт только действительно нужные для клиентов части контракта. Это избавляет

¹ **Контракт в микросервисной архитектуре** – это соглашение, определяющее, как два или более микросервиса взаимодействуют друг с другом. Он описывает формат и структуру данных, которые передаются между сервисами, а также протоколы и методы для обмена информацией. Контракт позволяет гарантировать, что сервисы будут правильно интерпретировать данные и обеспечивать совместимость между различными компонентами системы [2].

конечных пользователей от потока лишних деталей и технических полей, делая интеграцию проще и понятнее.

Создание *API* как продукта предполагает не только техническую реализацию интерфейсов, но и разработку модели монетизации. В предложенном решении пользователи платят за ежемесячную подписку, что даёт им доступ к сервисам на период действия выбранного тарифа, независимо от общего числа запросов. При этом действует лимит на количество запросов в час – это позволяет распределять нагрузку на систему и даёт определённую гарантию качества обслуживания, при этом не отпугивая клиентов жёсткими ограничениями по общему количеству обращений. Данный подход облегчает прогнозирование доходов от монетизации и одновременно сохраняет удобство для сторонних интеграторов. Они могут стабильно обращаться к *API*, придерживаясь лимитов, но не переживая о глобальных квотах.

Один из важных эффектов работы в парадигме «*API* как продукт» – масштабирование рынка: разработчики сторонних решений могут создавать надстройки и плагины, которые автоматически интегрируются в закупочные и кросс-продажные процессы. По мере расширения числа подписчиков улучшается лояльность к продукту и растёт сетевой эффект. Такой подход позволяет компании-разработчику конкурировать не просто на уровне предлагаемой функциональности (например, автоматического сбора данных о тендерах), но и на уровне экосистемы, к которой подключается всё больше участников. Для контроля этого процесса крайне важно обеспечить хорошо структурированную документацию, генерация которой автоматизирована с использованием *OpenAPI/AsyncAPI*, а также поддерживать стабильность версий, чтобы не нарушать взаимодействие с уже существующими интеграциями.

Заключение. В результате представленного подхода преимущества получают все стороны. Разработчики экономят время за счёт автоматизации документации и выстраивания стройной архитектуры взаимодействия сервисов. Клиенты (компании-покупатели) получают гарантированный доступ к тендерным данным в удобном формате и могут быстро встраивать новую функциональность в свои *ERP/CRM*-системы. Партнёры, интегрируясь с платформой, расширяют свой бизнес-профиль, а разработчик платформы укрепляет свой бренд, формируя новое направление дохода на рынке *B2B*-решений.

Список литературы

1. Митин, А., Смирнов, Д. Документирование *API* / Доклад на T-Meetup: Java, 20 ноября 2024.
2. Тестирование контрактов микросервиса [Электронный ресурс]. Режим доступа: <https://yapro.ru/article/17798>. Дата доступа: 18.03.2025.

UDC 004.01,004.42,004.75+339.1

API AS A PRODUCT IN THE DEVELOPMENT OF COMPLEX IT SOLUTIONS: A CASE STUDY OF PROCUREMENT AUTOMATION AND CROSS-SELLING MANAGEMENT

Tsuinchyk D.R.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Shelest H.V. – Master of Engineering and Technology, assistant of the department of ICSD

Annotation. The article examines the 'API as a product' model, which enables the centralization of documentation across all microservices and transforms the API into a distinct development pathway and a revenue source within the *B2B* sector.

Keywords: API as a product, *OpenAPI*, *AsyncAPI*, microservices architecture.