МТУСИ

МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ СВЯЗИ И ИНФОРМАТИКИ

Сравнение подходов к разработке API: GraphQL vs REST API

Цыганков Р.О.

ВВЕДЕНИЕ В API И ИХ РОЛЬ

.

01

API обеспечивают интеграцию систем и приложений.

Интерфейсы API позволяют различным системам работать совместно, обмениваться данными и функциями для повышения эффективности.

02

API упрощают обмен данными между приложениями.

С помощью API приложения могут обмениваться данными в реальном времени, улучшая взаимодействие и функциональность.

03

REST и GraphQL — популярные архитектуры API.

Эти архитектуры предоставляют разные подходы к разработке API и удовлетворению потребностей разработчиков.

ОСНОВЫ REST API

Понимание архитектурного стиля REST

Архитектурный стиль REST

REST (Representational State Transfer) — это архитектурный стиль, который использует стандартные **HTTP**-методы для взаимодействия с ресурсами.

Стандартные методы HTTP

Основные методы включают **GET**, **POST**, **PUT** и **DELETE**, которые определяют действия над ресурсами.

Уникальные URL для ресурсов

Каждый ресурс в **REST API** имеет уникальный **URI**, что позволяет к нему обращаться напрямую.

Форматы ответов

Ответы от **REST API** обычно формируются в формате **JSON** или **XML**, что упрощает их обработку.

ЧТО ТАКОЕ GRAPHQL?

Язык запросов для гибкости данных

ЕДИНЫЙ КОНЕЧНЫЙ ПУНКТ ЗАПРОСОВ

GraphQL позволяет использовать **один** конечный пункт для всех запросов, что упрощает взаимодействие с API.

ГИБКОСТЬ ВЫБОРА ПОЛЕЙ

Клиенты могут запрашивать только те **поля** данных, которые им действительно нужны, что снижает объем передаваемых данных.

АВТОМАТИЧЕСКАЯ ДОКУМЕНТАЦИЯ

Документация генерируется **автоматически** на основе схемы, что облегчает понимание и использование API.

КЛЮЧЕВЫЕ ПРЕИМУЩЕСТВА REST API

поддержка и производительность REST API

ПРОСТОТА И ЛЕГКОСТЬ

REST API предлагают интуитивно понятный и легкий в использовании интерфейс, что упрощает разработку и интеграцию.

ШИРОКАЯ ПОДДЕРЖКА

REST API широко используются и поддерживаются во многих языках программирования и фреймворках, что способствует их популярности.

КЭШИРОВАНИЕ ДАННЫХ

REST API поддерживают кэширование, что позволяет значительно увеличить производительность приложений за счет снижения нагрузки на сервер.

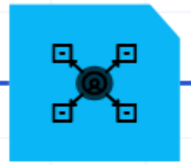
СТАНДАРТИЗАЦИЯ

REST API следуют стандартным протоколам HTTP, что обеспечивает совместимость и упрощает интеграцию с другими системами.

НЕДОСТАТКИ REST API

ЧРЕЗМЕРНАЯ ВЫБОРКА ДАННЫХ

REST APIs могут возвращать больше данных, чем необходимо, что приводит к неэффективной передаче информации.



МНОГОЧИСЛЕННЫЕ ЗАПРОСЫ

Для получения связанных данных часто требуется выполнять несколько запросов, что увеличивает нагрузку на сервер.

УПРАВЛЕНИЕ ВЕРСИЯМИ API

Сложности с управлением версиями API могут привести к проблемам с совместимостью и поддержкой.

ПРЕИМУЩЕСТВА ИСПОЛЬЗОВАНИЯ GRAPHQL

ключевые преимущества GraphQL по сравнению с REST API, включая гибкость запросов и проверку типов.

ЗАПРОС ТОЛЬКО НЕОБХОДИМЫХ ДАННЫХ

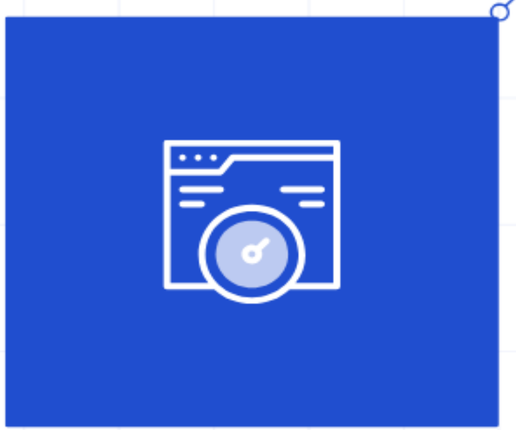
С помощью **GraphQL** вы можете запрашивать только те данные, которые вам действительно нужны, избегая избыточности и сокращая объем передаваемой информации.

ГИБКОСТЬ В ФОРМИРОВАНИИ ЗАПРОСОВ

GraphQL позволяет разработчикам формировать запросы по своему усмотрению, что упрощает процесс получения данных и улучшает пользовательский опыт.

АВТОМАТИЧЕСКАЯ ПРОВЕРКА ТИПОВ ДАННЫХ

Система **GraphQL** автоматически проверяет типы данных, что снижает вероятность ошибок и улучшает надежность запросов.



УЛУЧШЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ ПРИЛОЖЕНИЙ

Используя **GraphQL**, приложения могут работать более эффективно благодаря возможности запрашивать только нужные данные, что улучшает общую производительность.

НЕДОСТАТКИ GRAPHQL

проблемы и сложности, связанных с использованием GraphQL

СЛОЖНОСТЬ РЕАЛИЗАЦИИ

GraphQL требует значительных усилий для настройки и реализации, что может быть сложно для разработчиков.

НЕОБХОДИМОСТЬ ИНСТРУМЕНТОВ

Для эффективной работы с **GraphQL** необходимо внедрять дополнительные инструменты для кэширования и мониторинга.

ОБРАБОТКА ОШИБОК

GraphQL обеспечивает более сложный механизм обработки ошибок, что может усложнить разработку и отладку приложений.

КОГДА ИСПОЛЬЗОВАТЬ REST ИЛИ GRAPHQL?

Сравнение применения REST и GraphQL в API

Выбор между REST и GraphQL зависит от специфики вашего проекта. Оба подхода имеют свои сильные и слабые стороны, и понимание их различий поможет вам принять обоснованное решение.

REST ПОДХОДИТ ДЛЯ ПРОСТЫХ API

REST лучше всего работает с простыми и хорошо структурированными API, обеспечивая легкость в использовании и понимании.

GRAPHQL ДЛЯ СЛОЖНЫХ СИСТЕМ

GraphQL рекомендуется для сложных систем, где много связанных данных, позволяя запрашивать только необходимые данные.

УЧИТЫВАЙТЕ ТРЕБОВАНИЯ ПРОЕКТА

Важно учитывать требования проекта при выборе между **REST** и **GraphQL**, так как это влияет на архитектуру и производительность.

ПРЕДПОЧТЕНИЯ КОМАНДЫ

Предпочтения команды также играют важную роль в выборе между **REST** и **GraphQL**, так как разные команды имеют разные уровни опыта.

СПАСИБО
ЗА ВНИМАНИЕ!

Контактная информация

111024, г. Москва,
улица Авиамоторная, 8а
АДРЕС ОРГАНИЗАЦИИ

+7 (495) 957-79-17
ТЕЛЕФОН

mtuci@mtuci.ru
E-MAIL