

Основы API

1. Что такое API и для чего он используется?

API – это механизмы, которые позволяют двум программным компонентам взаимодействовать друг с другом, используя набор определений и протоколов.

2. В чем различие между SOAP и REST API?

SOAP – это протокол, а REST – архитектурный стиль.

3. Какие основные преимущества использования API для бизнеса?

Одно из главных преимуществ использования технологии API — это упрощение и автоматизация бизнес-процессов. Зачастую компании используют уже готовые решения и платформы для получения, анализа и хранения информации.

Методы HTTP

1. Какие основные методы HTTP есть и для чего каждый из них используется?

GET – Метод запрашивает представление ресурса. Запросы с использованием этого метода могут только извлекать данные.

HEAD - запрашивает ресурс так же, как и метод GET, но без тела ответа.

POST - используется для отправки сущностей к определённому ресурсу. Часто вызывает изменение состояния или какие-то побочные эффекты на сервере.

PUT - заменяет все текущие представления ресурса данными запроса.

DELETE - удаляет указанный ресурс.

CONNECT - устанавливает "туннель" к серверу, определённому по ресурсу.

OPTIONS - используется для описания параметров соединения с ресурсом.

TRACE - выполняет вызов возвращаемого тестового сообщения с ресурса.

PATCH - используется для частичного изменения ресурса.

2. Что такое статусные коды HTTP и как они классифицируются?

HTTP (status code) — это трёхзначное число, с которого начинается любой ответ сервера на запрос по протоколу HTTP. Код кратко сообщает суть ответа — был ли выполнен запрос или возникла ошибка. В зависимости от кода ответа посетитель либо будет видеть содержимое страницы или результаты какого-то действия, либо не будет.

3. Какие существуют различия между методами POST и PUT?

Разница между PUT и POST в том, что PUT является идемпотентным, т. е. единичный и множественные вызовы этого метода, с идентичным набором данных, будут иметь тот же результат выполнения (без побочных эффектов), в случае с POST, множественный вызов с идентичным набором данных может повлечь за собой сторонние эффекты.

REST API

1. Что такое REST и каковы его основные принципы?

REST API представляет собой архитектурный стиль API и используется для регламентирования взаимодействия между клиентской и серверной частью ПО. Он основан на принципах REST архитектуры (Representational State Transfer, или передача состояния представления), которые обеспечивают эффективное взаимодействие между клиентами и серверами в распределенной сетевой среде.

2. Как в REST API используются стандарты URI (Uniform Resource Identifier)?

Ресурсы API REST разрешаются при помощи идентификаторов ресурсов, входящих в состав унифицированного идентификатора ресурса (Uniform Resource Identifier, URI). Эти URI используются для разрешения объектов, представляемых как ресурсы REST. Например, следующий URI обеспечивает доступ к собранию бизнес-объектов PERSON.

3. Как REST API обеспечивает безопасность и аутентификацию пользователей?

Безопасность: API-интерфейсы RESTful можно защитить с помощью различных методов, таких как шифрование SSL/TLS и аутентификация OAuth. Это помогает обеспечить защиту API от несанкционированного доступа и безопасность клиентских данных.

Расширенные вопросы

1. Что такое OAuth и как оно используется в контексте REST API?

OAuth - открытый протокол авторизации, который позволяет предоставить третьей стороне ограниченный доступ к защищенным ресурсам пользователя без необходимости передавать ей (третьей стороне) логин и пароль. Примечание: Протокол OAuth авторизация используется только в контексте портала.

2. Каковы основные различия между REST и GraphQL API?

В REST вызываемая вами конечная точка (endpoint) — это и есть сущность объекта. В GraphQL сущность объекта отделена от того, как именно вы его получаете.

В REST структура и объем ресурса определяются сервером. В GraphQL сервер определяет набор доступных ресурсов, а клиент указывает необходимые ему данные прямо в запросе.

3. Какие лучшие практики следует применять при проектировании REST API?

- Используйте kebab-case для URL

Google рекомендует использовать kebab-case.

- Используйте camelCase для параметров

Вот пример получения списка продуктов в магазине.

- Используйте множественное число для коллекций

Если вы хотите получить всех пользователей.

- URL должен начинаться с коллекции и заканчиваться идентификатором

Это позволяет обеспечить единство и непротиворечивость дизайна.

- Не используйте глаголы в URL ресурсов

Вместо этого пользуйтесь HTTP методами для описания операций.

- Пользуйтесь глаголами в URL операций

Например, если вы хотите переслать уведомление пользователю.

- Используйте camelCase для JSON свойств

- Состояние

RESTful HTTP сервисы обязаны реализовывать методы /health, /version и /metrics.

- Используйте инструменты для разработки REST API

Для этих целей достаточно хороших решений, например API Blueprint и Swagger.

- Используйте простой порядковый номер для версий

И всегда указывайте его на самом верхнем уровне.

- Указывайте количество ресурсов в ответе на запрос

4. Каким образом можно тестировать API? Назовите инструменты и подходы.

- Тестирование методов

Каждый метод API проверяют по отдельности — чтобы убедиться, что они работают правильно и возвращают ожидаемые результаты. Тестирование включает проверку входных данных, выполнение операций и проверку вывода.

- Тестирование взаимодействий

Проверка взаимодействия API с другими API, компонентами программы и сервисами. Показывает, что он успешно отправляет и получает данные, а также обрабатывает различные сценарии использования.

- Тестирование авторизации и аутентификации

Проверка доступа к API: как работают механизмы авторизации, кто и к каким функциям и данным имеет доступ.

- Тестирование обработки ошибок

Проверка поведения API в случае непредвиденных ситуаций и ошибок — например, передачи некорректных данных. Позволяет убедиться, что API правильно обрабатывает исключения и передаёт в программу верные коды ошибок.

- Тестирование производительности

Проверка работы API при повышенных нагрузках — его пропускной способности и производительности. Позволяет убедиться, что API не «отвалится» в случае повышенного спроса на программу.

- Тестирование безопасности

Проверка уязвимостей безопасности API, которая помогает предотвратить утечку данных или несанкционированный доступ. Проверяются меры безопасности API и проводятся тесты на проникновение для установления возможных уязвимостей.

5. Какие существуют типы авторизации в API и как они реализуются на практике?

API ключ - API представляет собой длинную строку, которую обычно включают либо в URL запроса, либо в заголовок запроса.

Basic Auth - С помощью этого метода отправитель помещает пару имя пользователя: пароль в заголовок запроса. Имя пользователя и пароль кодируются с помощью Base64, который представляет собой метод кодирования, который преобразует имя пользователя и пароль в набор из 64 символов для обеспечения безопасной передачи.

НМАС (код авторизации сообщений на основе хэша) - НМАС только отправитель, и получатель знают секретный ключ, который больше неизвестен никому. Отправитель создает сообщение на основе некоторых системных свойств (например, отметка времени запроса плюс идентификатор учетной записи).

OAuth 2.0 - Такой подход опирается на сервер аутентификации для связи с сервером API для предоставления доступа.