Асинхронные коммуникации или зачем чат-боту нужен RESTful интерфейс

RESTful подход подразумевает, что обмен данными между системами происходит в режиме stateless т.е. без сохранения состояния, что означает информация о клиенте не сохраняется между запросами на получение, и каждый запрос является отдельным и не связанным.

Поскольку коммуникация с чат-ботом носит *асинхронный* характер, т. е. время между очередными запросами через чат-бота может быть очень большим и мы не можем заранее знать когда пользователь вернется, чтобы продолжить диалог и не знаем когда он завершит сеанс, значит мы не можем создавать сеанс для каждого пользователя, идентифицироваться во внешней системе *должна* платформа Metabot и обслуживать поток запросов от всех пользователей бота.

На практике вы будете сталкиваться с устаревшими системами и сервисами, которые нельзя назвать RESTful. Вы можете столкнуться с устаревшим подходом к проектированию API взаимодействия когда сессия пользователя истекает через некоторое время и ее нужно обновлять.

Если разработчики системы не могут изменить API и у вас нет возможности сделать доработку, то реализовывайте работу с API в два шага: первым шагом создавайте сессию, которую обновляйте, если она истекла, перед следующем запросом, а вторым шагом выполняйте нужный вам запрос.

Что такое API?

API (Application Programming Interface или интерфейс для прикладного программирования) — это набор определений и протоколов для создания и интеграции прикладного программного обеспечения. Иногда его называют *контрактом* между поставщиком информации и потребителем информации, устанавливающим контент, требуемый от потребителя (вызов), и контент, требуемый поставщиков (ответ). Например, дизайн API для погодной службы может указывать, что потребитель должен предоставить почтовый индекс, а производитель

должен ответить двумя частями: первая — высокая температура, а вторая — низкая.

Другими словами, если вы хотите взаимодействовать с компьютером или системой для получения информации или выполнения какой-либо функции, API поможет вам передать то, что вы хотите, этой системе, чтобы она могла понять и выполнить запрос.

Вы можете думать об API как о *посреднике* между пользователями или клиентами и ресурсами или веб-сервисами, которые они хотят получить. Это также способ для организации обмена ресурсами и информацией, сохраняя при этом безопасность, контроль и аутентификацию, определяя, кто и к чему получает доступ.

Еще одно преимущество API заключается в том, что вам не нужно знать особенности кэширования — как извлекается ваш ресурс или откуда он берется.

Что такое REST?

REST (Representational State Transfer или передача репрезентативного состояния) — это набор архитектурных ограничений, а не протокол или стандарт. Разработчики API могут реализовать REST различными способами. Для веб-служб, построенных с учётом REST (то есть не нарушающих накладываемых им ограничений), применяют термин «RESTful».

REST API (также известный как RESTful API) — это интерфейс прикладного программирования, который соответствует ограничениям архитектурного стиля REST и позволяет взаимодействовать с веб-службами. REST означает передачу репрезентативного состояния (Representational State Transfer) и был создан компьютерным ученым Роем Филдингом.

Когда клиентский запрос выполняется через RESTful API, он передает представление о состоянии ресурса запрашивающей стороне или конечной точке. Эта информация или представление доставляется в одном из нескольких форматов через HTTP: JSON (обозначение объектов Javascript), HTML, XLT, Python, PHP или обычный текст. JSON является наиболее популярным форматом файлов для использования, потому что, несмотря на свое название, он не зависит от языка, а также удобен для чтения как людьми, так и машинами.

Еще кое-что, о чем следует помнить: заголовки и параметры также важны в HTTP-методах HTTP-запроса RESTful API, поскольку они содержат важную информацию об идентификаторе в отношении метаданных запроса, авторизации, универсального идентификатора ресурса (URI), кэширования, файлов cookie и т. д. Существуют заголовки запросов и заголовки ответов, каждый из которых имеет собственную информацию о HTTP-соединении и коды состояния.

Чтобы API считался RESTful, он должен соответствовать следующим критериям:

- Архитектура клиент-сервер, состоящая из клиентов, серверов и ресурсов, с запросами, управляемыми через HTTP;

- Связь клиент-сервер без сохранения состояния (stateless), что означает информация о клиенте не сохраняется между запросами на получение, и каждый запрос является отдельным и не связанным;
- Кэшируемые данные, которые оптимизируют взаимодействие клиент-сервер;
- Единый интерфейс между компонентами, чтобы информация передавалась в стандартном виде. Это требует, чтобы:
  - Запрашиваемые ресурсы были идентифицируемы и отделены от представлений, отправленных клиенту;
  - Ресурсы могли управляться клиентом через представление, которое они получают, потому что представление содержит достаточно информации для этого;
  - Самоописательные сообщения, возвращаемые клиенту, содержали достаточно информации, чтобы описать, как клиент должен их обрабатывать;
  - Гипертекст/гипермедиа доступны, что означает, что после доступа к ресурсу клиент должен иметь возможность использовать гиперссылки, чтобы найти все другие доступные в настоящее время действия, которые он может предпринять;
- Многоуровневая система, организующая каждый тип серверов (отвечающих за безопасность, балансировку нагрузки и т. д.), включающая поиск запрашиваемой информации в виде иерархий, невидимых для клиента;
- Код по запросу (необязательно): возможность отправлять исполняемый код с сервера клиенту по запросу, расширяя функциональные возможности клиента. **Metabot поддерживает эту возможность.**

Хотя REST API должен соответствовать этим критериям, он по-прежнему считается более простым в использовании, чем предписанный протокол, такой как SOAP (Simple Object Access Protocol или простой протокол доступа к объектам), который имеет особые требования, такие как обмен сообщениями XML, а также встроенную безопасность и соответствие транзакциям, которые делают его медленнее и тяжелее.

REST, напротив, представляет собой набор рекомендаций, которые можно реализовать по мере необходимости, делая REST API более быстрыми и легкими, с повышенной масштабируемостью, что идеально подходит для Интернета вещей (IoT) и разработки диалоговых и мобильных приложений.

> В данный момент, платформа Metabot поддерживает только REST интерфейсы с JSON и XML форматами обмена данными. Другие способы интеграции систем, например, SOAP, RPC сейчас не поддерживаются.

Благодарим Redhat за информацию для этой секции.

Версия #3
Юрий Гарашко создал 2023-06-14 13:02:06 UTC
Ирина Петрова обновил 2024-03-20 13:10:35 UTC