

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

RESTful системы: основные принципы и применение

Гаврилюк Владислав Игоревич, студент магистратуры;

Гаврилюк Виктория Игоревна, студент

Белорусский государственный университет информатики и радиоэлектроники (г. Минск)

В этой статье речь пойдет о наиболее часто встречающейся архитектуре REST.

Ключевые слова: контроль информации сервиса, принцип работы, тип архитектуры, html, REST.

Эта технология получила свое распространение в 2000-ом году, после ее подробного описания в докторской диссертации под названием Architectural Styles and the Design of Network-based Software Architectures. Она была презентована Роем Филдингом, который является одним из создателей HyperText Transfer Protocol (далее HTTP).

REST расшифровывается как Representational State Transfer. Это специальный вид архитектуры программного

обеспечения для различных распределенных систем. Наиболее часто такая архитектура применяется в системах для создания веб-служб. Примером такой системы может служить World Wide Web. А все системы, которые используют данный тип архитектуры называются RESTful-системами. Принцип работы такой системы представлен на рис. 1.

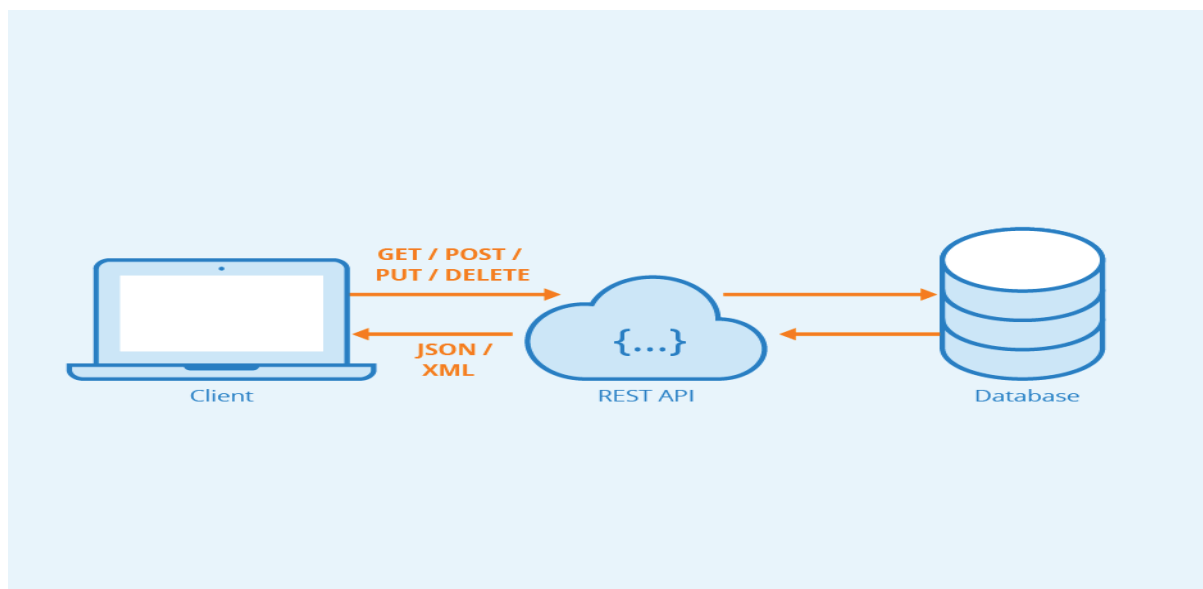


Рис. 1. Принцип работы RESTful системы

Отличительной особенностью всех RESTful-систем является простой интерфейс для контроля информации. Он не подразумевает наличия каких-либо внутренних уровней или подслоев. Каждый блок данных в такой системе имеет свой собственный уникальный спецификатор. Примером такого идентификатора может служить Uniform Resource Locator (далее URL). Каждый

URL-идентификатор имеет строгий формат. Таким образом можно утверждать, что URL — первичный ключ для объекта данных. В это же время не существует каких-либо требований к формату данных по данному ключу. Это может быть, как jpg-файл, так в тоже время это может быть HTML либо docx-документ.

Второй отличительной чертой для данного типа архитектуры является контроль информации сервиса, базирующийся на различных протоколах передачи данных. Самым же распространенным является протокол HTTP.

Любые манипуляции данными в рамках данного протокола осуществляются с использованием следующих методов:

- GET — используется для получения данных
- POST — используется для добавления, изменения и удаления данных
- PUT — используется для добавления и изменения данных
- DELETE — используется для удаления данных
- HEAD — используется для получения заголовка файла либо ресурса

CONNECT — используется для установки связи с сервером на базе какого-либо ресурса

OPTIONS — необходим для описания всех параметров для работы с данными сервера

PATCH — используется для местного изменения ресурсов

TRACE — используется для получения тестовых данных с сервера

Однако следует принимать во внимание, что все CRUD операции могут быть осуществлены как с использованием всех четырех методов, так и с использованием только POST и GET. Дополнительные параметры каждого метода проиллюстрируем на рис. 2.

HTTP Method ↕	Request Has Body ↕	Response Has Body ↕	Safe ↕	Idempotent ↕	Cacheable ↕
GET	Optional	Yes	Yes	Yes	Yes
HEAD	No	No	Yes	Yes	Yes
POST	Yes	Yes	No	No	Yes
PUT	Yes	Yes	No	Yes	No
DELETE	No	Yes	No	Yes	No
CONNECT	Yes	Yes	No	No	No
OPTIONS	Optional	Yes	Yes	Yes	No
TRACE	No	Yes	Yes	Yes	No
PATCH	Yes	Yes	No	No	No

Рис. 2. Параметры HTTP методов

Алгоритм формирования REST запросов

На основании всего выше изложенного мы можем построить общий алгоритм для формирования запроса к серверу с использованием REST архитектуры:

Шаг 1 — необходимо сформировать URL по которому будет производиться запрос.

Шаг 2 — необходимо прописать все заголовки (например, Content-type, Authorization).

Шаг 3 — произвести выбор HTTP-запроса.

Шаг 4 — на данном шаге формируется тело запроса. Тип тела запроса должен быть сформирован в соответствии со значением заголовка Content-type.

Шаг 5 — произвести выполнение запроса.

Шаг 6 — произвести обработку полученных результатов.

Практическое применение REST архитектуры

Главным преимуществом REST сервисов является наличие полной совместимости с любой системой. Это обусловлено тем, что методы парсинга XML, а также выполнение запросов с использованием HTTP протокола используется повсеместно.

Литература:

1. Jon Flanders. RESTful.NET: Build and Consume RESTful Web Services with .NET 3.5. — 1st Edition. — Sebastopol: O’Reilly Media, Inc., 2008. — 310 с.
2. Balaji Varanasi. Spring Rest. — 1st Edition. — New-York: Apress Media LLC, 2015. — 222 с.