

**1.**

REST (Representational State Transfer) — это архитектурный стиль для создания веб-сервисов, основанный на HTTP.

Доступ к веб-службам RESTful можно получить с помощью HTTP-запросов, таких как GET, POST, PUT и DELETE.

Эти запросы используются для извлечения, создания, обновления и удаления ресурсов соответственно. RESTful API широко используются и популярны для создания веб-приложений и сервисов.

Они гибкие и могут использоваться для взаимодействия с различными системами и платформами. Примеры запросов RESTful API включают получение данных с сервера с помощью запроса GET, создание нового ресурса с помощью запроса POST, обновление существующего ресурса с помощью запроса PUT и удаление ресурса с помощью запроса DELETE.

**2.**

В целом, как запросы HTTP, так и ответы имеют следующую структуру:

1. *Стартовая строка (start line)* — используется для описания версии используемого протокола и другой информации — вроде запрашиваемого ресурса или кода ответа.
2. *HTTP-заголовки (HTTP Headers)* — несколько строчек текста в определенном формате, которые либо уточняют запрос, либо описывают содержимое *тела* сообщения.
3. Пустая строка, которая сообщает, что все метаданные для конкретного запроса или ответа были отправлены.

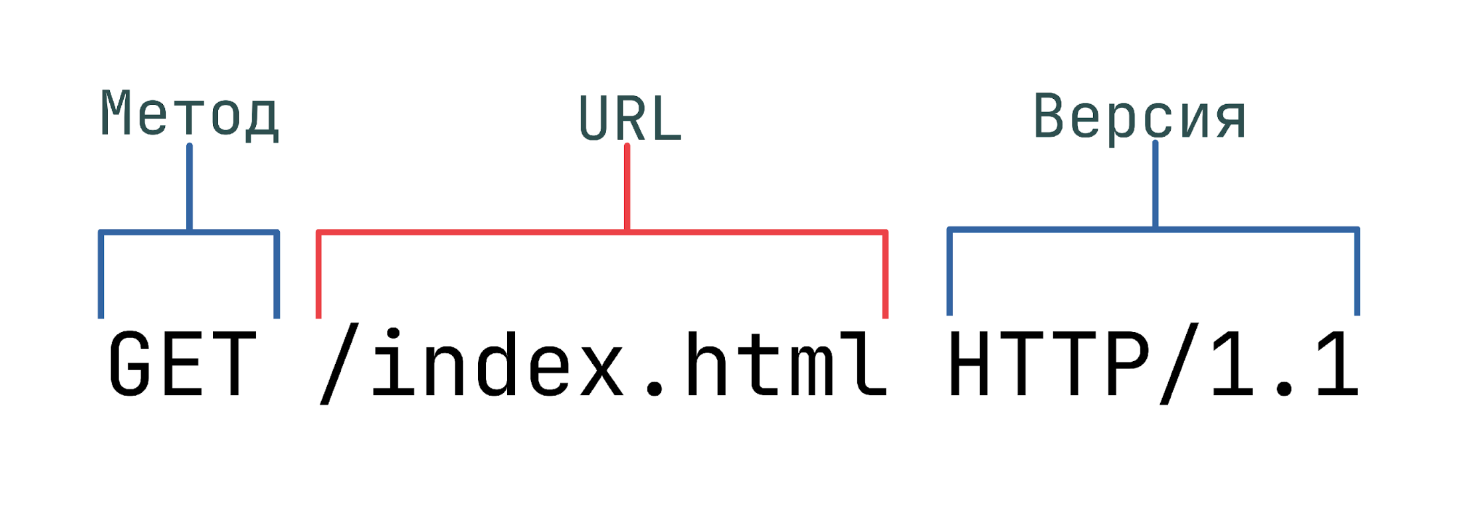
Опциональное *тело сообщения*, которое содержит данные, связанные с запросом, либо документ (например HTML-страницу), передаваемый в ответе.

**Строка запроса** – указывает метод передачи, URL-адрес, к которому нужно обратиться и версию протокола HTTP.

Стартовая строка HTTP-запроса состоит из трех элементов:

1. *Метод HTTP-запроса* (method, реже используется термин verb). Обычно это короткое слово на английском, которое указывает, что конкретно нужно сделать с запрашиваемым ресурсом. Например, метод GET сообщает серверу, что пользователь хочет получить некоторые данные, а POST — что некоторые данные должны быть помещены на сервер.
2. Цель запроса. Представлена указателем ресурса URL, который состоит из протокола, доменного имени (или IP-адреса), пути к конкретному ресурсу на сервере. Дополнительно может содержать указание порта, несколько параметров HTTP-запроса и еще ряд опциональных элементов.
3. Версия используемого протокола (либо HTTP/1.1, либо HTTP/2), которая определяет структуру следующих за стартовой строкой данных.

В примере ниже стартовая строка указывает, что в качестве метода используется GET, обращение будет произведено к ресурсу /index.html, по версии протокола HTTP/1.1:



**Заголовки** – описывают тело сообщений, передают различные параметры и др. сведения и информацию.

**HTTP-заголовок** представляет собой строку формата «Имя-Заголовок:Значение», с двоеточием(:) в качестве разделителя. Название заголовка не учитывает регистр, то есть между Host и host, с точки зрения HTTP, нет никакой разницы. Однако в названиях заголовков принято начинать каждое новое слово с заглавной буквы. Структура значения зависит от конкретного заголовка. Несмотря на то, что заголовок вместе со значениями может быть достаточно длинным, занимает он всего одну строчку.

В запросах может передаваться большое число различных заголовков, но все их можно разделить на три категории:

1. О**бщего назначения**, которые применяются ко всему сообщению целиком.
2. **Заголовки запроса** уточняют некоторую информацию о запросе, сообщая дополнительный контекст или ограничивая его некоторыми логическими условиями.
3. **Заголовки представления**, которые описывают формат данных сообщения и используемую кодировку. Добавляются к запросу только в тех случаях, когда с ним передается некоторое тело.

**Тело сообщения** - это сами данные, которые передаются в запросе. Тело сообщения – это необязательный параметр и может отсутствовать.

Завершающая часть HTTP-запроса — это его тело. Не у каждого HTTP-метода предполагается наличие тела. Так, например, методам вроде GET, HEAD, DELETE, OPTIONS обычно не требуется тело. Некоторые виды запросов могут отправлять данные на сервер в теле запроса: самый распространенный из таких методов — POST.

**3.**

Maven — это инструмент автоматизации сборки, который используется для управления зависимостями, сборки и упаковки проектов Java, а также создания документации.

Он использует файл объектной модели проекта (POM) для определения структуры проекта, зависимостей и процесса сборки.

При создании проекта с помощью Maven следует стандартный жизненный цикл сборки, включающий такие этапы, как компиляция, тестирование, упаковка и установка.

В процессе сборки Maven загружает необходимые зависимости из удаленных репозиториев и сохраняет их в локальном репозитории.

Это позволяет эффективно управлять зависимостями и гарантирует, что проект построен с правильными версиями необходимых библиотек.

Maven также поддерживает подключаемые модули, которые можно использовать для расширения функциональности и автоматизации таких задач, как анализ кода, тестирование и развертывание.

В целом, Maven упрощает процесс сборки и помогает обеспечить согласованность и надежность сборки Java-проектов.

**4.**

Quarkus — это платформа Java, разработанная RedHat и предназначенная для создания легких и быстрых приложений, которые могут работать в контейнерах.

Он ориентирован на использование установленных библиотек и стандартов Java и поддерживает несколько механизмов аутентификации через свою структуру безопасности.

Quarkus использует архитектуру на основе расширений, которая позволяет разработчикам добавлять в свои проекты только необходимые зависимости, уменьшая размер и время запуска приложения.

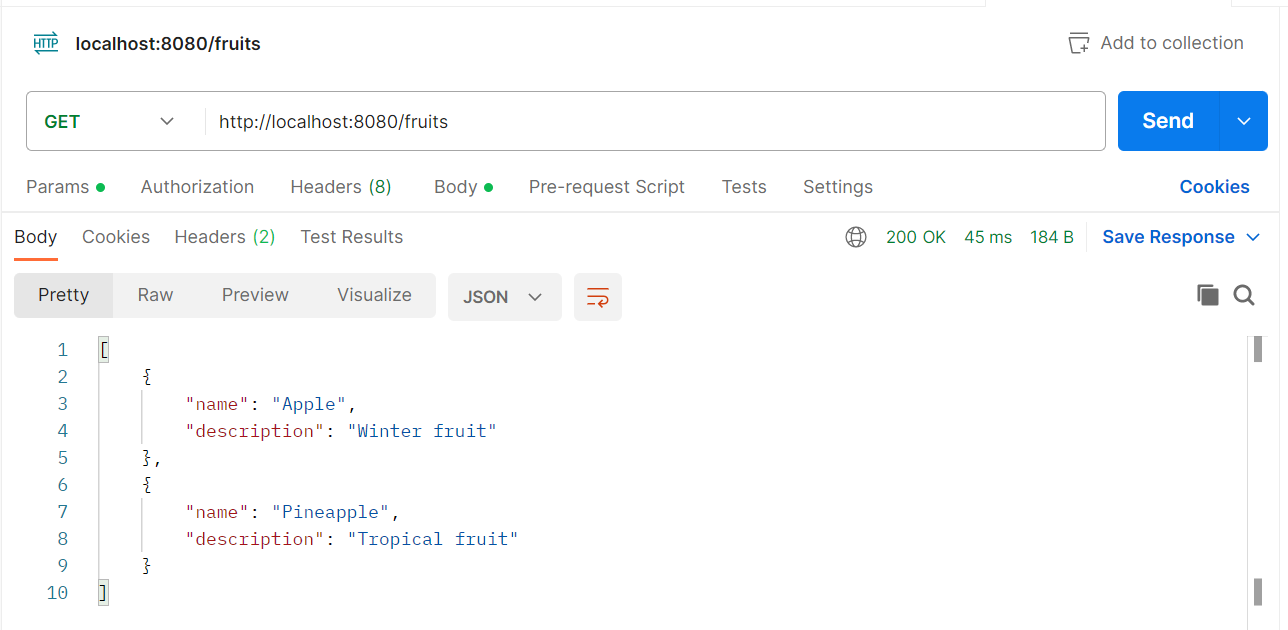
Он также поддерживает горячую перезагрузку, что позволяет разработчикам вносить изменения в код и сразу же видеть результаты, не перезапуская приложение.

Quarkus предоставляет инструменты для тестирования приложений, включая управление зависимостями, внедрение, имитацию и профили конфигурации.

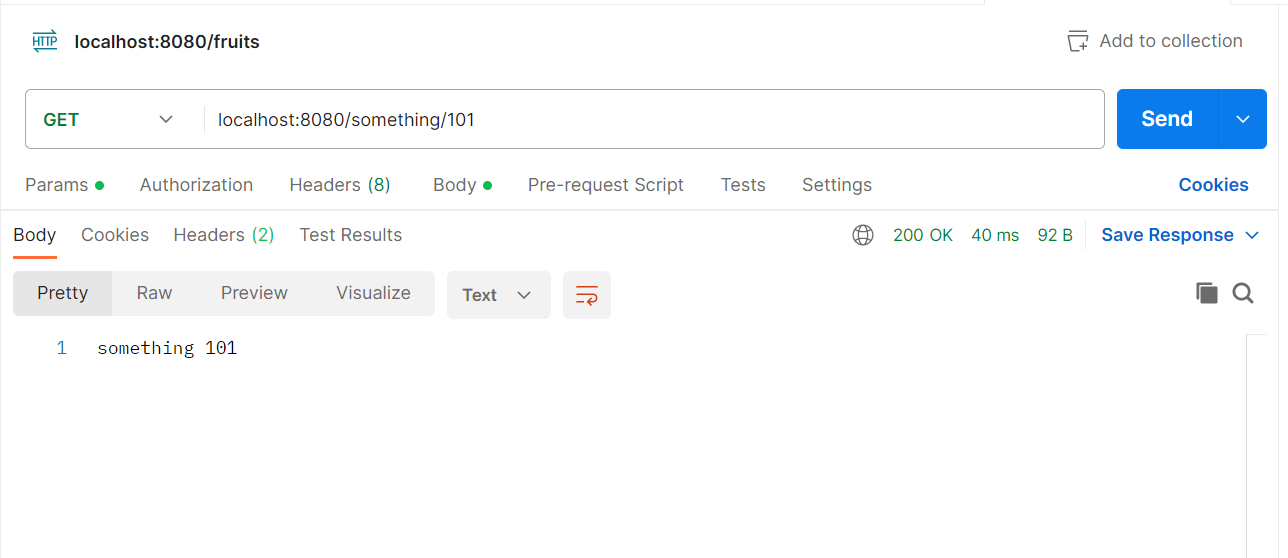
Кроме того, Quarkus поддерживает создание собственных образов, что позволяет дополнительно уменьшить размер и время запуска приложения.

В целом Quarkus предоставляет современный и эффективный способ разработки приложений Java, которые могут работать в контейнерах и использовать преимущества новейших технологий и стандартов.

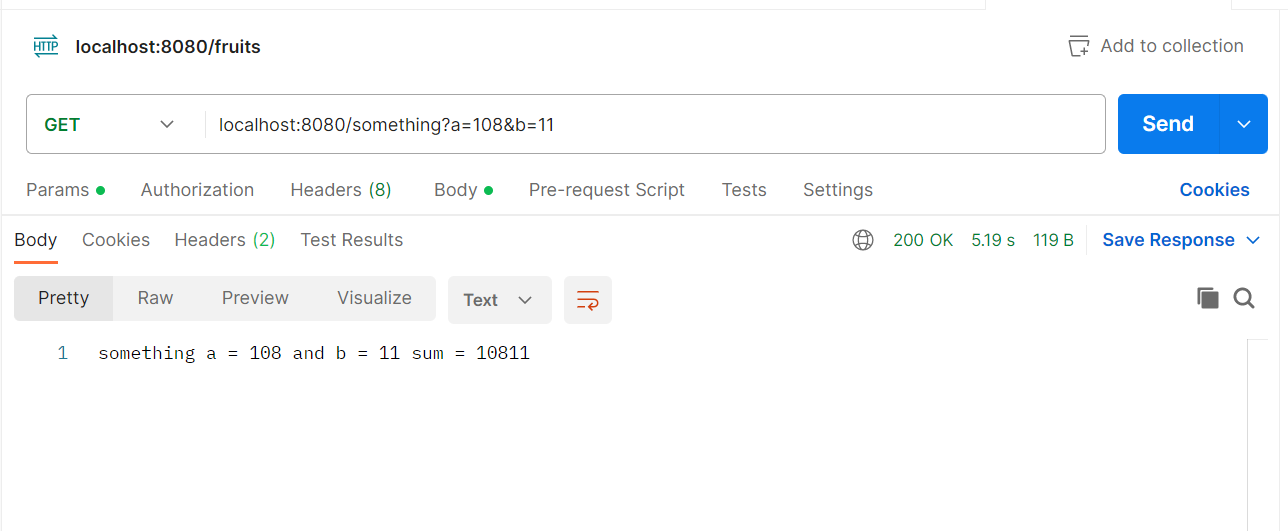
Тестирование написанной программы с помощью Postman



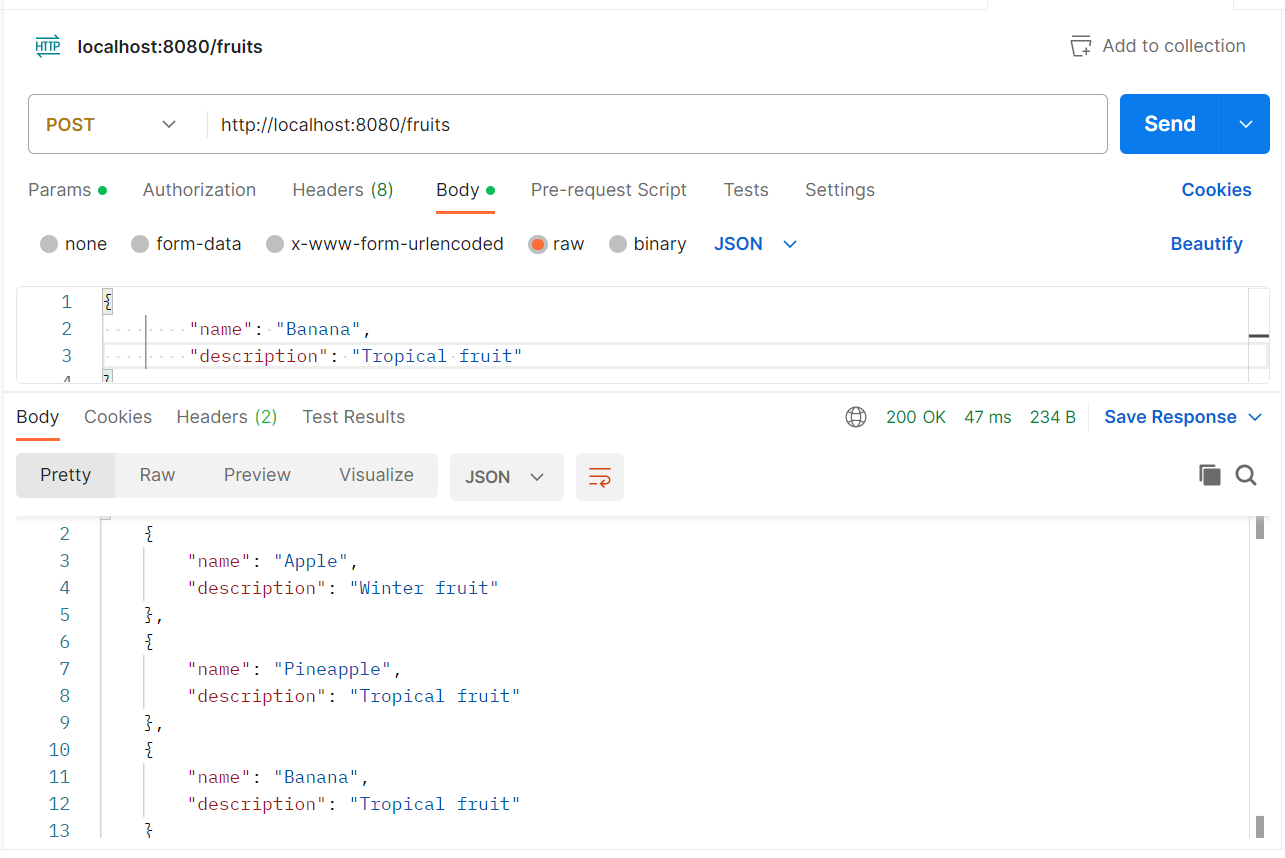
Простой GET запрос



GET запрос с PathParam



GET запрос с QueryParam



POST запрос на добавление фрукта “Banana”