

МИКРОСЕРВИСНАЯ АРХИТЕКТУРА ПРИЛОЖЕНИЯ КАК СИСТЕМА, УПРОЩАЮЩАЯ РАЗРАБОТКУ И ПОДДЕРЖКУ КОДА

Петров Д.Г. Email: Petrov1152@scientifictext.ru

*Петров Денис Георгиевич – специалист по защите информации в автоматизированных системах,
кафедра вычислительной техники,
Чувашский государственный университет им. И.Н. Ульянова, г. Чебоксары*

Аннотация: в статье анализируется метод описания объектов данных в строках передаваемых запросов, разбиение возможных запросов на логические единицы и их обработка. Рассмотрен формат передаваемых данных для изменения и добавления новых структур данных. Представлен универсальный протокол передачи данных для сервисов с микросервисной архитектурой, которые не могут быть написаны на разных языках программирования. Приведен пример возможных сущностей системы обработки запросов к базе данных, описанной API REST методологией передачи данных.

Ключевые слова: микросервисная архитектура, распределение данных.

MICROSERVICE APPLICATION ARCHITECTURE AS A SYSTEM THAT SIMPLIFIES CODE DEVELOPMENT AND SUPPORT

Petrov D.G.

*Petrov Denis Georgievich – Specialist in information protection in automated systems,
DEPARTMENT OF COMPUTER ENGINEERING,
CHUVASH STATE UNIVERSITY I.N. ULYANOV, CHEBOKSARY*

Abstract: the article analyzes the method of describing data objects in the rows of transmitted queries, splitting possible queries into logical units and their processing. The format of the transmitted data for changing and adding new data structures is considered. A universal data transfer protocol for services with a multi-service architecture that cannot be written in different programming languages is presented. An example of possible entities of the database query processing system is described, which is described by the REST API data transfer methodology.

Keywords: microservice architecture, data distribution.

УДК: 004.451.26

Метод распределения больших приложений на отдельные модули, взаимодействующие между собой по одному протоколу передачи данных, улучшает скорость разработки и поддержку системы. Разбиение монолитной части программного приложения [1, с. 52] – важная часть разработки больших систем. Для таких систем необходим универсальный метод описания объектов, передаваемых данных. Методологии REST API описывает такой способ передачи данных. Каждый новый запрос открывает соединение и отправляет данные на сервер. Существует несколько методов отправки запроса, используемые в системе:

- GET – получение данных. Ответ – данные от сервера в заданном формате и код возврата;
- POST – обновление данных. В теле запроса содержатся параметры описывающие данные для изменения. Ответ – данные от сервера и код возврата;
- PUT – создание нового объекта данных. В теле запроса содержатся параметры создаваемого объекта. Ответ – код возврата;
- DELETE – удаление существующих данных. В теле запроса могут содержаться параметры удаляемых данных, по которым сервер однозначно определяет, что необходимо удалить. Ответ – код возврата.

Формат передачи и получения данных – JSON, используется для описания объектов данных, имеет свойство вложенности данных (объект в объекте), поддерживает сериализацию данных – перевод текстового представления данных в объект данных, используемый программой.

Параметры запроса формируются из пары – название переменной, символ двоеточия, значение переменной. Например,

```
{  
  Id: int,  
  Name: string,  
  State: bool  
}
```

, где `id` – переменная типа `int` (целочисленное значение), `name` – переменная типа `string` (строковая переменная), `state` – переменная типа `bool` (булева переменная).

Запрос к серверу состоит из:

- Метод запроса;
- Описание системных переменных запроса в `header`;
- URL запроса к конечной точке (сущности) данных, на сервер;

Пример структуры возможных сущностей, описывающих структуру базы данных представлена на рисунке 2.

URL `/api/{version}/pacient/{id_pacient}`, где:

- URL – адрес сервера;
- `api` – вызов внешней функции;
- `{version}` – версия структуры протокола обмена данными;
- `Pacient` – сущность, к которой направлен запрос;
- `{id_pacient}` – `id` пациента, параметр для выбранной сущности.

Например, для запроса данных о пациенте №1057 будет выглядеть следующим образом: `domain.ru/pacient/1057/` Данный запрос будет возвращать информацию о пациенте с уникальным идентификатором, равным 1057. Запрос выполняется с методом `GET`.

Пример отправки запроса на сервер по методологии REST API, по протоколу передачи данных HTTP, представлен на рисунке 1.

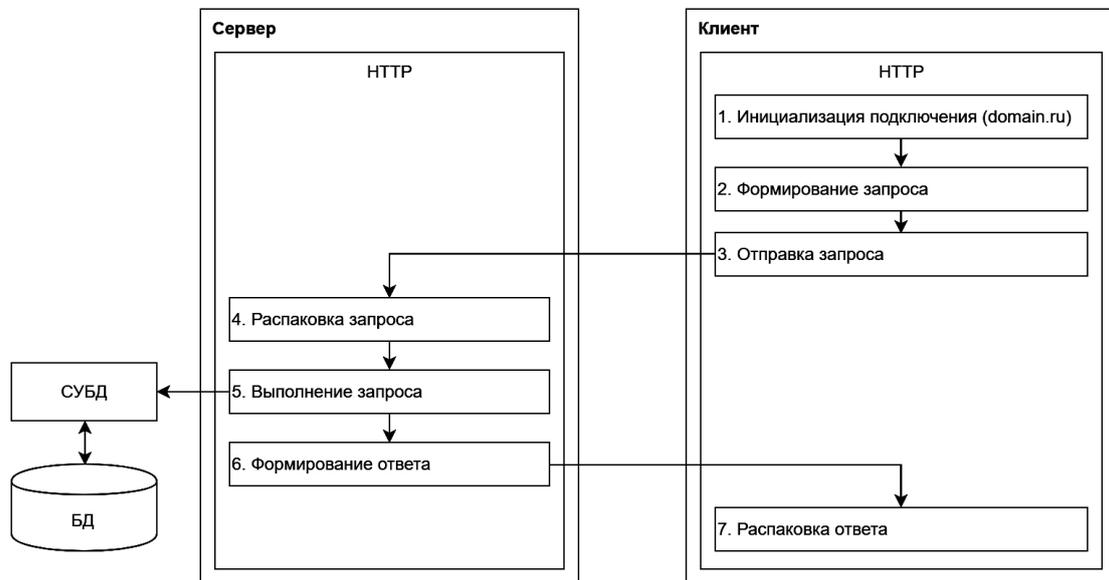


Рис. 1. Схема действий при отправке запроса на сервер

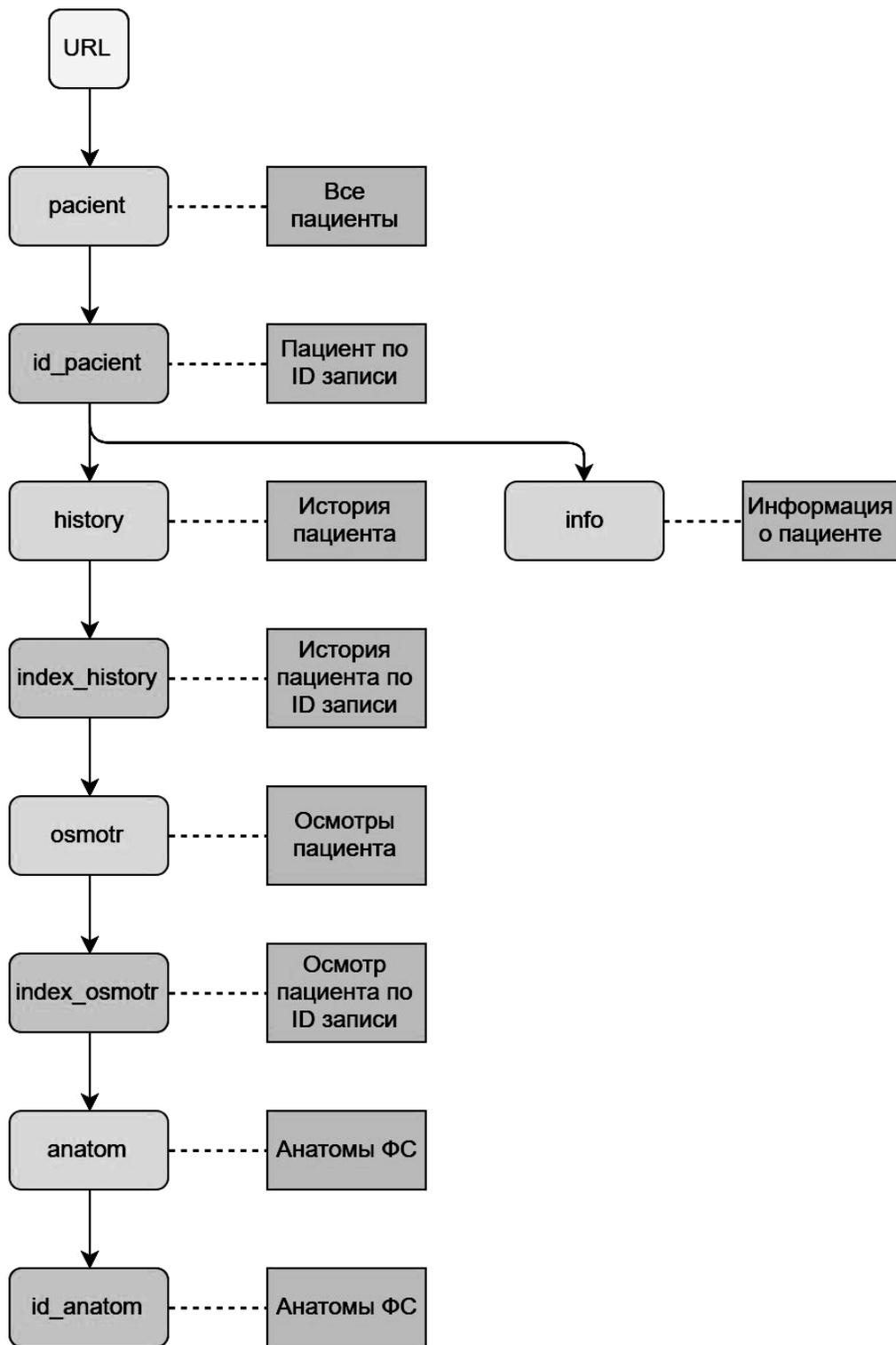


Рис. 2. Пример структуры API-методов

Список литературы / References

1. Newman Sam, Building Microservices, 2016.