

REST

Предпосылки

REST (Representational state transfer) – это стиль архитектуры программного обеспечения для распределенных систем, таких как World Wide Web, который.

Термин REST был введен в 2000 году Роем Филдингом, одним из авторов HTTP-протокола. Системы, поддерживающие REST, называются RESTful-системами.

Структура

1. Маршрут отправки
2. Тип метода
3. Заголовки
4. Тело (или данные)

Маршрут (URL)

Маршрут (URL) – это адрес, по которому отправляется ваш запрос. Его структура примерно следующая:

`/api/1.0/...`

Остальные пути в маршрутах определяются в документации

Примеры маршрутов

- `/api/1.0/papers`
- `api/1.0/papers/1`
- `/api/1.0/balance-validation?versionId=2&_=1538599269230`

JSON

JSON – JavaScript Object Notation – общий формат для отправки и приема данных посредством REST API.

```
{
  "data": [
    {
      "id": 1,
      "title": "string",
      "status": "ATTENTION",
      "createDate": 1538599148462,
      "updateDate": 1538599148462,
      "comment": "string",
      "locked": true,
      "tmpFileName": null,
      "authors": [],
      "fileId": null,
      "fileName": null,
      "fileCreateDate": null
    }
  ],
  "error": null
}
```

Тип метода

Метод обозначает тип производимого запроса. Всего существует пять типов запросов:

- GET
- POST
- PUT
- PATCH
- DELETE

Типы методов

GET – используется для получения со стороны сервера определенного ресурса. Если вы производите этот запрос, сервер ищет информацию и отправляет ее вам назад.

POST – нужен для создания определенного ресурса на сервере. Сервер создает в базе данных новую сущность и оповещает вас, был ли процесс создания успешным.

PUT и PATCH – используются для обновления определенной информации на сервере.

DELETE – удаляет указанную сущность из базы

Версии API

Для выпуска новых версий и продолжения поддержки старых клиентов

Ограничения REST

Единый интерфейс

Четыре принципа единого интерфейса:

1. Основан на ресурсах
2. Манипуляции над ресурсами через представления
3. Само-документируемые сообщения
4. Hypermedia as the Engine of Application State (HATEOAS). Клиенты предоставляют статус через содержимое body, параметры строки запроса, заголовки запросов и запрашиваемый URI (имя ресурса).

Ограничения REST

Отсутствие состояний (Stateless)

Кеширование ответа (Cacheable)

Клиент-сервер (Client-Server)

Многоуровневая система (Layered System)

"Код по требованию" (Code on Demand - опционально)

Достоинства

Масштабируемость обеспечивается:

- Простотой унифицированного интерфейса;
- Открытостью компонентов к возможным изменениям для удовлетворения изменяющихся потребностей;
- Прозрачностью связей между компонентами системы для сервисных служб;
- Переносимостью компонентов системы путем перемещения программного кода вместе с данными;
- Надёжностью, выражающейся в устойчивости к отказам на уровне системы при наличии отказов отдельных компонентов, соединений или данных.

Инструменты

- Swagger
- Postman
- curl

Как работает в приложении?