

# Краткая шпаргалка по запросам REST API

---

## URL запроса

<https://www.happy-path.com/contact/2?phone>

<https://> = протокол

[www.happy-path.com/](http://www.happy-path.com/) = домен

[contact/](http://www.happy-path.com/contact/) = путь

[2](http://www.happy-path.com/contact/2) = параметр пути

[?phone](http://www.happy-path.com/contact/2?phone) = параметр запроса.

## HTTP-глаголы

**GET** – получает запись из базы данных

**POST** – добавляет запись в базу данных

**PUT** – заменяет запись в базе данных новой записью

**PATCH** – изменяет существующую запись в базе данных

**DELETE** – удаляет запись из базы данных

## Параметры

**Параметр пути:** является частью пути

Примеры:

- [contacts/2](#) вернет контакт с id 2
- [company/3/contacts/4](#) вернет контакт с id 4 в компании с id 3.

**Параметр запроса:** присоединяется к пути

Начинается со знака вопроса.

Может образовывать цепочки через амперсанд.

Может запрашивать определенный ресурс через знак равенства.

Примеры:

- [contacts/2?phone](#) вернет номер телефона контакта
- [contacts/2?phone&email](#) вернет номер телефона и электронную почту контакта.
- [contacts/?lastname=Smith](#) вернет контакты по фамилии Смит.

# Краткая шпаргалка по запросам REST API

---

## Заголовки

### Заголовки запросов:

**Accept** – типы медиа, разрешенные в ответе

**Accept-Charset** – наборы символов, разрешенные в ответе

**Accept-Encoding** – кодировка, разрешенная в ответе

**Accept-Language** – язык, разрешенный в ответе

**Authorization** – авторизационные данные для запроса

**Cache-Control** – определяет, разрешено ли кэширование

**Content-Length** – длина тела запроса

**Content-Type** – тип тела запроса

**Cookie** – куки, ранее возвращенная сервером для сохранения запроса

**User-Agent** – ПО, при помощи которого осуществлен запрос – например, браузер

### Заголовки ответов:

**Cache-Control** – определяет, можно ли кэшировать запрос, и как надолго

**Content-Encoding** – тип кодировки в данных

**Content-Language** – язык ответа

**Content-Length** – длина ответа

**Content-Type** – формат ответа

**Date** – дата и время ответа

**Location** – определяет, был ли запрос перенаправлен, или где создавался ресурс

**Set-Cookie** – куки, которую можно использовать для хранения данных

## Коды ответов

### Уровень 100: информационный

**100 Continue** – сервер получил заголовки запроса, и клиент теперь должен передать тело запроса.

### Уровень 200: успешный

**200 OK** – запрос был успешным.

**201 Created** – запрос создал новый ресурс.

**202 Accepted** – запрос был принят, но обработка еще не завершена.

**204 No Content** – запрос был обработан и вернул пустой ответ.

### Уровень 300: перенаправление

**301 Moved Permanently** – ресурс был перемещен в место, указанное в заголовке ответа.

# Краткая шпаргалка по запросам REST API

---

## Уровень 400: клиентская ошибка

**400 Bad Request** – запрос имеет неверный формат или невалиден.

**401 Unauthorized** – у клиента нет нужной авторизации.

**403 Forbidden** – у клиента нет прав для доступа к ресурсу.

**404 Not Found** – запрошенный ресурс не найден.

**405 Method Not Allowed** – неверный HTTP-глагол.

**409 Conflict** – запрос вызывает конфликт ресурсов друг с другом.

## Уровень 500: серверная ошибка

**500 Internal Server Error** – возникло неожиданное состояние.

**502 Bad Gateway** – сервер получил плохой ответ от вышестоящего сервера.

**503 Service Unavailable** – сервер недоступен или перегружен.

**504 Gateway Timeout** – сервер не получил своевременный ответ от вышестоящего сервера.

## Формат JSON

### Пара имя-значение

```
"name": "Spot"
```

### Объект – группа пар "имя-значение"

```
{  
  "name": "Spot",  
  "type": "dog",  
  "age": "5"  
}
```

### Массив: группа объектов

```
"pets": [  
  {  
    "name": "Spot",  
    "type": "dog",  
    "age": "5"  
  },  
  {  
    "name": "Fluffy",  
    "type": "cat",  
    "age": "3"  
  }  
]
```