

REST

Грушевский Ю.В.

REST

REST (от англ. Representational State Transfer — «передача состояния представления») — архитектурный стиль взаимодействия компонентов распределенного приложения в сети.

Автор идеи и термина Рой Филдинг 2000г.

REST на сегодняшний день практически вытеснил все остальные подходы, в том числе дизайн основанный на SOAP и WSDL

Что нам дает REST

- ❑ Масштабируемость
- ❑ Общность интерфейсов
- ❑ Независимое внедрение компонентов
- ❑ Промежуточные компоненты, снижающие задержку, усиливающие безопасность

Когда использовать REST?

- ❑ Когда есть ограничение пропускной способности соединения
- ❑ Если необходимо кэшировать запросы
- ❑ Если система предполагает значительное масштабирование
- ❑ В сервисах, использующих AJAX

Преимущества REST

- ❑ Отсутствие дополнительных внутренних прослоек, что означает передачу данных в том же виде, что и сами данные
- ❑ Каждая единица информации (ресурс) однозначно определяется URL.
- ❑ Как происходит управление информацией ресурса — это целиком и полностью основывается на протоколе передачи данных.

HTTP методы

GET – получение (чтение) ресурса

POST – создание ресурса

PUT – обновление ресурса (целиком)

PATCH – обновление ресурса (частичное)

DELETE – удаление ресурса

GET – получение (чтение) ресурса

GET <http://www.example.com/api/v1.0/users> (вернуть список пользователей)

GET <http://www.example.com/api/v1.0/users/12345> (вернуть данные о пользователе с id 12345)

GET <http://www.example.com/api/v1.0/users/12345/orders> (вернуть список заказов пользователя с id 12345)

POST – создание ресурса

POST <http://www.example.com/api/v1.0/customers> (создать новый ресурс в разделе customers)

POST <http://www.example.com/api/v1.0/customers/12345/orders> (создать заказ для ресурса с id 12345)

PATCH и PUT – обновление ресурса

PUT <http://www.example.com/api/v1.0/users/12345> (обновить данные пользователя с id 12345)

PUT <http://www.example.com/api/v1.0/users/12345/orders/98765> (обновить данные заказа с id 98765 для пользователя с id 12345)

DELETE – удаление ресурса

DELETE <http://www.example.com/api/v1.0/customers/12345> (удалить из customers ресурс с id 12345)

DELETE <http://www.example.com/api/v1.0/customers/12345/orders/21> (удалить у ресурса с id 12345 заказ с id 21)

Коды состояний HTTP

1xx – информационные

100 – Continue

2xx – успешные

200 – OK

201 – Created

204 – No Content

3xx – перенаправления

301 – Moved Permanently

307 – Temporary Redirect

4xx – клиентские ошибки

400 – Bad Request

401 – Unauthorized

403 – Forbidden

404 – Not Found

5xx – ошибки сервера

500 – Internal Server Error

502 – Bad Gateway

503 – Service Unavailable

Полезные ссылки

<https://medium.com/@andr.ivas12/rest-простым-языком-90a0bca0bc78>