Database Systems Lab
REST

Christian Rauch

(changed: August 28, 2024)

Representational State Transfer

REST is an architecural pattern for stateless client-server communication. It is simple, scalable, and based on HTTP and related web standards. Everything is treated as a resource identified by a URI.

```
GET /api/products
GET /api/products?category=shoes&brand=adidas
GET /api/products/280030
GET /api/users/u3013
GET /api/users/u3013/orders
GET /api/users/u3013/orders/43209-339/shipping/4
```

HTTP methods are used to indicate the operation to apply.

Read	→	GET
Create	→	POST
Update (partially)	→	PUT (PATCH)
Delete	→	DELETE

RESTful services are often prefixed with /api.

Read, Create, Update, Delete

The HTTP request header contains the request line, the host, and further meta data (e.g., user-agent: ..., accept: application/json).

```
POST /api/products      HTTP/1.1
host: 127.0.0.1:8080
Content-Type: application/json
```

The HTTP request body contains the payload (usually JSON).

```
{
  "name": "Converse Chucks classic",
  "price": 59.0,
  "categories": ["sneakers", "streetwear"]
}
```

POST, PUT, and PATCH are similar. GET and DELETE often need no body.

```
  GET    /api/products/28003
  DELETE /api/users/u3013/orders/43209-339
```

Backend Request Handling

```python
@app.route('/api/products/', methods=['POST'])
def add_product():
  data    = request.get_json()
  ps      = ProductService()
  product = ps.create_product(
    session['usr'], data['name'], data['price'])
  return jsonify(product), 201

@app.route('/api/orders/<oid>',
           methods=['GET', 'DELETE'])
def get_order(oid: int):
  ps = ProductService()
  if request.method == 'GET':
    order = ps.get_order(session['usr'], oid)
    return jsonify(order), 200
  else:
    ps.delete_order(session['usr'], oid)
    return jsonify({'deleted': oid}), 200
```

Frontend Response Handling

```
async function addProduct(name, price) {
  const response = await fetch(
    '/api/products/', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json'
      },
      body: JSON.stringify({
        name: name, price: price
      })
    });

  if (!response.ok) { ... return; }

  const product = await response.json();
  console.log('Product added:', product);
  ...
}
```

Conclusion

Thank you for your attention!