Contents

REST

REpresentational **S**tate **T**ransfer

- Uses **HTTP** protocol
- *Stateless* communication (no memory about prior requests)
- Scalable
- Resource is identified by **URL** e.g. `http://example.com/customers/1234`
- Data Exchange can use *text*, *XML*, *JSON*, *HTML*

Resource Linkage

Example: Order 1234: 23 products with id 44533 ordered by customer with id 4711

```
{
    "orderId":      1234,
    "amount":       23,
    "productRef":   "http://example.com/products/44533",
    "customerRef":  "http://example.com/customers/4711"
}
```

Standard Method Call Semantics

method	path	body	semantics
GET	order		return all orders
GET	order/1		return order with ID 1
POST	order	{"product": "milk", "amount": 2}	create new order based on body data
PUT	order/1	{ "product": "milk", "amount": 13 }	update order with ID 1 based on body data
DELETE	order/1		delete order with ID 1

method	path	body	semantics
PATCH	order/1	{ "amount": 13 }	partially update order with ID 1 based on body data

HTTP methods

- **GET**: retrieve information
- **POST**: submit data (often create resource)
- **PUT**: update resource
- **DELETE**: delete resource
- **PATCH**: partially update resource (only send changed parts)

Most Important Status Codes

- **100 range**: information
- **200 range**: indicates **success**
- **300 range**: redirects
- **400 range**: indicates **client error**
- **500 range**: indicates **server error**

Error Code	Meaning
200	**OK**
201	**Created**
204	**No Content**
400	**Bad Request**
401	Unauthorized
403	Forbidden
404	**Not Found**
409	**Conflict**
500	Internal Server Error
501	Not Implemented
503	Service Unavailable

Server Interface As Table

operation	HTTP method	path	request body	reply body	status codes
getOrder(int id): Order	GET	order/:id		Order	200, 404

operation	HTTP method	path	request body	reply body	status codes
createOrder(order: Order): Order	POST	order	Order	Order	201
getOrdersFiltered(date): Order[]	GET	order/[?date=:date&...]			