

HTTP, REST webové služby

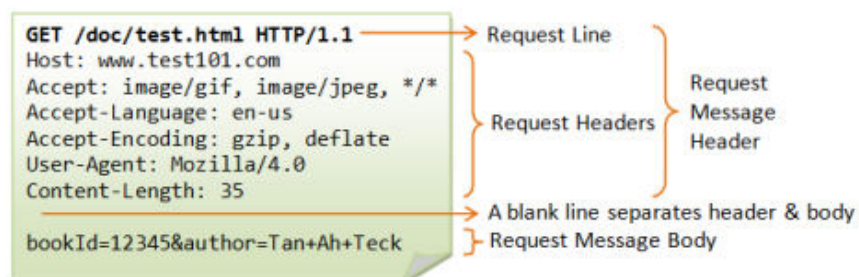
- **URI = Uniform Resource Identifier**
 - o řetězec znaků identifikující nějaký zdroj
 - o např. <http://www.fel.cvut.cz/cz/education/>
- **HTTP = Hypertext Transfer Protocol**
 - o aplikační protokol pro distribuované hypermedia systémy
 - o přenos hypertextových dokumentů
 - o základ komunikace WWW
- **REST = Representational State Transfer**
 - o architekturní styl pro distribuované prostředí (HTTP)

HTTP

- klient-server protokol aplikační vrstvy
- většinou přes TCP/IP spojení
- rozšiřitelný (podpora videí, obrázků,...)
- bezstavový
- cacheable
- vyžaduje spolehlivý transportní protokol (ne UDP)

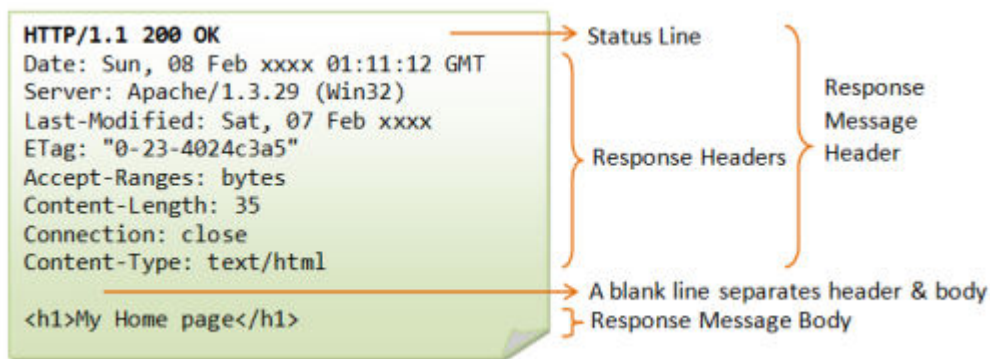
HTTP request

- message header
 - o request line – identifikace HTTP metody, URI a verze protokolu
 - o request headers
- message body



HTTP response

- message header
 - o status line – identifikace verze protokolu a status code odpovědi
 - o response headers
- message body



HTTP headers

Typical, often used HTTP headers

	Request	Response
Content	<ul style="list-style-type: none"> • Content-Type • Content-Length • Content-Encoding • Accept 	<ul style="list-style-type: none"> • Content-Type • Content-Length • Content-Encoding
Caching	<ul style="list-style-type: none"> • If-Modified-Since • If-Match 	<ul style="list-style-type: none"> • Last-Modified • ETag
Miscellaneous	<ul style="list-style-type: none"> • Cookie • Host • Authorization • User-Agent 	<ul style="list-style-type: none"> • Set-Cookie • Location

HTTP metody

- **safe** = nemodifikují zdroj (jeho reprezentaci); mohou být cacheovány
- **idempotent** = stále stejný výsledek, nehlédě na počet volání

GET

- načtení zdroje z požadované URI
- může mít vedlejší efekty, ale spíše ne
- může být podmíněčný či částečný (If-Modified-Since, Range...)

HTTP Method	Safe	Idempotent
GET	✓	✓
PUT	✗	✓
DELETE	✗	✓
POST	✗	✗

POST

- požadavek na vytvoření nového zdroje z daného body
- může být použit i k aktualizaci
- měl by dostat odpověď 201, stav a lokaci vytvořeného zdroje
- např.: POST /questions

PUT

- požadavek na uložení entity na daném URI
- když zdroj neexistuje, server jej může vytvořit
- většinou se používá k aktualizaci
- např.: PUT /questions/{ID}

DELETE

- požadavek na smazání zdroje na dané URI
- nemusí být okamžité

HTTP Response Status Codes

- **1xx** – informační; transfer-protocol level; moc se nepoužívá
- **2xx = SUCCESS**
 - o **200 OK** = požadavek úspěš; většinou obsahuje data
 - o **201 Created** = vrátí Location header nového zdroje
 - o **202 Accepted** = server obdržel požadavek a zpracovává ho
 - o **204 No Content** = požadavek úspěš; nic k vrácení; např. po DELETE
- **3xx = REDIRECTION**
 - o **304 Not Modified** = zdroj nebyl modifikován; lze použít cached verzi
- **4xx = CLIENT ERROR**
 - o **400 Bad Request** = špatná syntax
 - o **401 Unauthorized** = vyžadována autorizace
 - o **403 Forbidden** = nedostatečná oprávnění
 - o **404 Not Found**
 - o **405 Method Not Allowed** = nepodporuje danou HTTP metodu
 - o **409 Conflict** = konflikt zdroje s daty klienta
 - o **415 Unsupported Media Type**
- **5xx = SERVER ERROR**
 - o **500 Internal Server Error** = chyba zpracování požadavku
 - o **501 Not Implemented**

RESTful web services

- architektonický styl, ne standard
- pro distribuované hypermedia systémy
- používají se HTTP 1.1 metody (HTTP verbs) → podpora Uniform Interface omezení
- definován 6 principy / architekturními omezeními
- Systém/API, který principy dodrží, je RESTful

1.) Client-server

2.) Uniform interface (jednotné rozhraní)

- o resource-based
- o manipulace se zdroji skrz reprezentace
- o samovysvětlující zprávy
- o hypermedia as the engine of app state

3.) Bezstavové interakce

4.) Cacheable

5.) Layered systém

6.) Code on demand (nepovinné)

HTTP GET

```
GET /eshop/rest/categories HTTP/1.1
Host: localhost:8080
Accept: application/json
Cache-Control: no-cache
```

```
HTTP/1.1 200
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Content-Type: application/json; charset=UTF-8

[{"id": 2,
  "name": "CPU"}, {"id": 7,
  "name": "Graphic card"}, {"id": 11,
  "name": "RAM"}]
```

HTTP verbs – POST

```
POST /eshop/rest/categories HTTP/1.1
Host: localhost:8080
Content-Type: application/json
Cookie: EAR_JSESSIONID=18162708908C126C0BA5A3D3081CCAC9
Cache-Control: no-cache
```

```
{
  "name": "Motherboard"
}
```

```
HTTP/1.1 201
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Location: http://localhost:8080/eshop/rest/categories/151
```

HTTP verbs – PUT

```
PUT /eshop/rest/products/8 HTTP/1.1
Host: localhost:8080
Content-Type: application/json
Cookie: EAR_JSESSIONID=18162708908C126C0BA5A3D3081CCAC9
```

```
{
  "id":8,
  "name":"MSI GeForce GTX 1050 Ti 4GT OC",
  "amount":50,
  "price":4490.0,
  "categories":[{"id":7,
    "name":"Graphic card"}],
  "removed":false
}
```

```
HTTP/1.1 204
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
```

HTTP verbs – DELETE

```
DELETE /eshop/rest/products/8 HTTP/1.1
Host: localhost:8080
Cookie: EAR_JSESSIONID=18162708908C126C0BA5A3D3081CCAC9
Cache-Control: no-cache
```

```
HTTP/1.1 204
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
```

HTTP Verb	CRUD	Collection (e.g. /categories)	Specific Item (e.g. /categories/{id})
POST	Create	201 Created ¹	405 Method Not Allowed /409 Conflict ³
GET	Read	200 OK, list of categories	200 OK, single category/404 Not Found ⁴
PUT	Update/Replace	405 Method Not Allowed ²	200 OK/ 204 No Content /404 Not Found ⁴
PATCH	Update/Modify	405 Method Not Allowed ²	200 OK/ 204 No Content /404 Not Found ⁴
DELETE	Delete	405 Method Not Allowed ²	200 OK/ 204 No Content /404 Not Found ⁴

Table : Recommended return values of HTTP methods in combination with the resource URIs.

Jmenné konvence

- zdroje by měly být označeny jako podstatná jména, nikoliv slovesa či akce
- plurál, pokud možno
- URI by mělo dodržovat předvídatelnou a hierarchickou strukturu

Correct usages

POST /customers/12345/orders/121/items

GET /customers/12345/orders/121/items/3

GET|PUT|DELETE /customers/12345/configuration

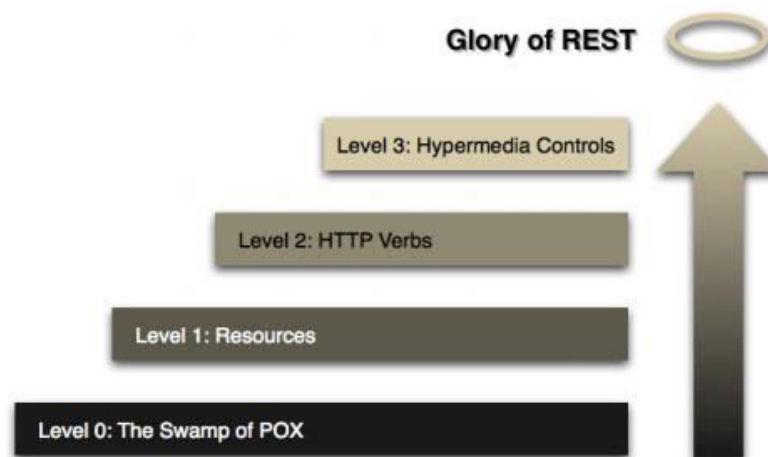
Anti-patterns

GET /services?op=update_customer&id=12345&format=json

PUT /customers/12345/update

The Richardson Maturity Model

- provides a way to evaluate compliance of API to REST constraints



HATEOAS (Hypermedia as the Engine of Application State)

- finální úroveň RMM
- klient potřebuje malou či žádnou znalost API
- potřebuje jen rozumět hypermedia
- server nabídne v response adresy pro další endpointy
 - o řekne nám, co můžeme dělat dál a jakou pro to potřebujeme URI
- často obtížně implementovatelné

Linked Data (propojená data)

- způsob publikace strukturovaných dat umožňující je propojit s ostatními daty
- založené na idei Webu, tentokrát machine-readable
- oproti webu používají jazyk RDF pro ukládání dat s popisem jejich sémantického významu
- např. znalostní systémy
- zdroje mají globální identifikátory