

Zápasíme s REST API

Lukáš Křečan

REST API Architect

GoodData



Něco o mě

- GoodData REST API architekt
- Před tím několik let v korporacích
 - SOAP-WS
 - Spring WS Test
- Java programátor
- blog.krecan.net

Agenda

- Co je to REST?
- Kde mu začínají docházet síly?
- Jak to řešit?

Otázky?

REST v GoodData

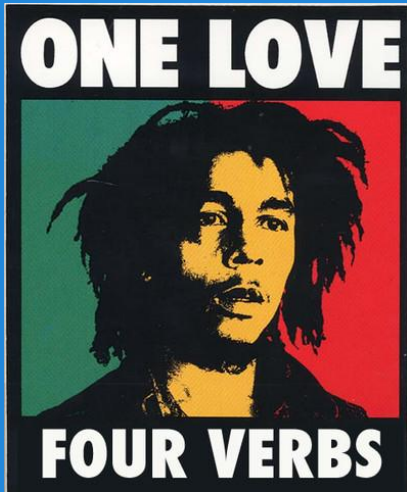
- Tlustý JS klient
- Interní API

REST JE DOBRÝ

ale ne jednoduchý

REST je filozofie

ne standard

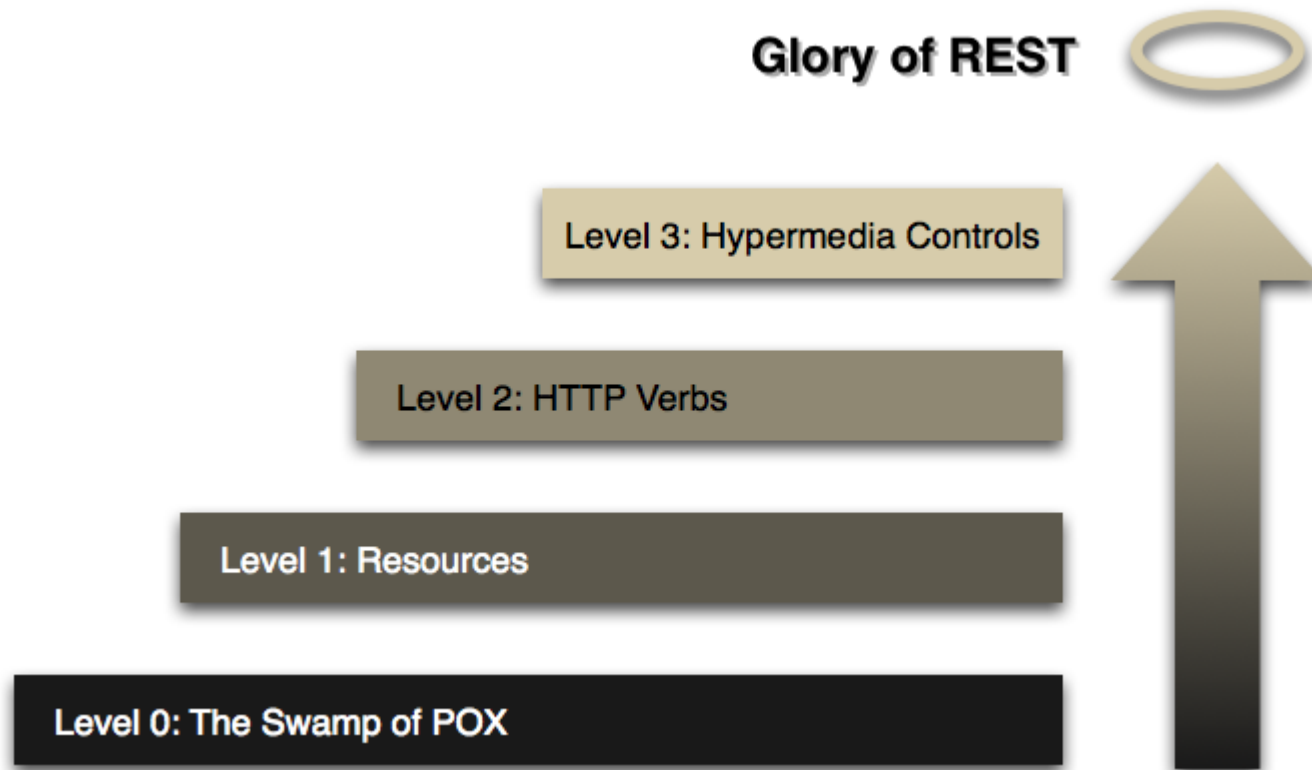


Make Web not RPC!

ART, NOT ROUTINE



Richardson Maturity Model



Resource URI

- Resource je objekt
- Každý resource má URI
 - Je to jeho ID (reference)
- Podstatná jména, množné číslo

/pets

/pets/123

/owners

/owners/456

Výhody URI

- Routing
- Logování
- Autorizace
- Cache

Monitoring

Key Public Resource SLAs

resource ↕	Tu08	Tu09	Tu10	Tu11	Tu12	Tu13	Tu14
/gdc/account/login POST 200	Green	Green	Green	Green	Yellow	Red	Red
/gdc/account/profile/X GET 200	Green	Yellow	Red	Red	Red	Red	Red
/gdc/account/profile/X/projects GET 200	Green	Green	Green	Green	Green	Green	Green
/gdc/account/token GET 200	Green	Yellow	Yellow	Red	Red	Red	Red
/gdc/app/account/bootstrap GET 200	Green	Green	Green	Green	Green	Red	Red
/gdc/app/projects/X/execute POST 201	Green	Green	Green	Green	Green	Green	Green
/gdc/app/projects/X/log POST 204	Green	Yellow	Yellow	Red	Red	Red	Red
/gdc/md/X/dataResult/X GET 200	Green	Green	Green	Green	Red	Red	Red
/gdc/md/X/dataResult/X GET 202	Green	Green	Green	Green	Red	Red	Red
/gdc/md/X/dataResult/X GET 204	Green	Green	Green	Green	Yellow	Red	Red

Cvičení

- Jakou URI by měl mít seznam mazlíčků daného majitele?

Udružujte mělkou hierarchii

~~/owners/789/pets~~

~~/owners/789/pets/123~~

/pets?owner=789

/pets?vet=555&owner=789

Slovesa jen když to nejde jinak

- `/convert?from=CZK`
`&to=USD&amount=100`

Metody alias slovesa

	POST (C)	GET (R)	PUT (U)	DELETE (D)
<code>/pets</code>	Vytvoř nový	Vrať seznam	X*	Vymaž seznam*
<code>/pets/123</code>	X	Vrať jeden	Aktualizuj / vytvoř*	Smaž

Výhody

- Jasná sémantika
 - GET - bezpečný
 - PUT, DELETE - idempotentní
 - POST - nebezpečný
- Vím co mě čeká
- Proxy a cache s tím umí pracovat

Nevýhody

- Ne všechno je CRUD
- GET parametry jenom v URL
 - omezení na délku (??)
 - GET s tělem (fůůůj)
 - POST tunneling
- Hromadné operace
 - Nastav slevu u všech majitelů z daného města
 - Vrať mi detail uživatelů s těmito ID

Svatý grál - Univerzální Knihovna

	POST	GET	PUT	DELETE
<code>/thing}s</code>	Vytvoř nový	Vrať seznam	X	Vymaž seznam
<code>/thing}s/123</code>	X	Vrať jeden	Aktualizuj / vytvoř	Smaž

- Framework na serveru
- Knihovna na klienta

Detaily nejsou
standardizované

Judská Lidová Fronta



http://www.youtube.com/watch?v=gb_qHP7VaZE

Detaily jako

- Formát kolekcí
- Stránkování
- Formát odkazů
- URI šablony
- HATEOAS

Frakce

- JSON API <http://jsonapi.org/format/>
- Collection+JSON <http://amundsen.com/media-types/collection/examples/>
- Siren <https://github.com/kevinswiber/siren>
- HAL http://stateless.co/hal_specification.html
- AtomPub <http://atompub.org/>
- ...

Reprezentace

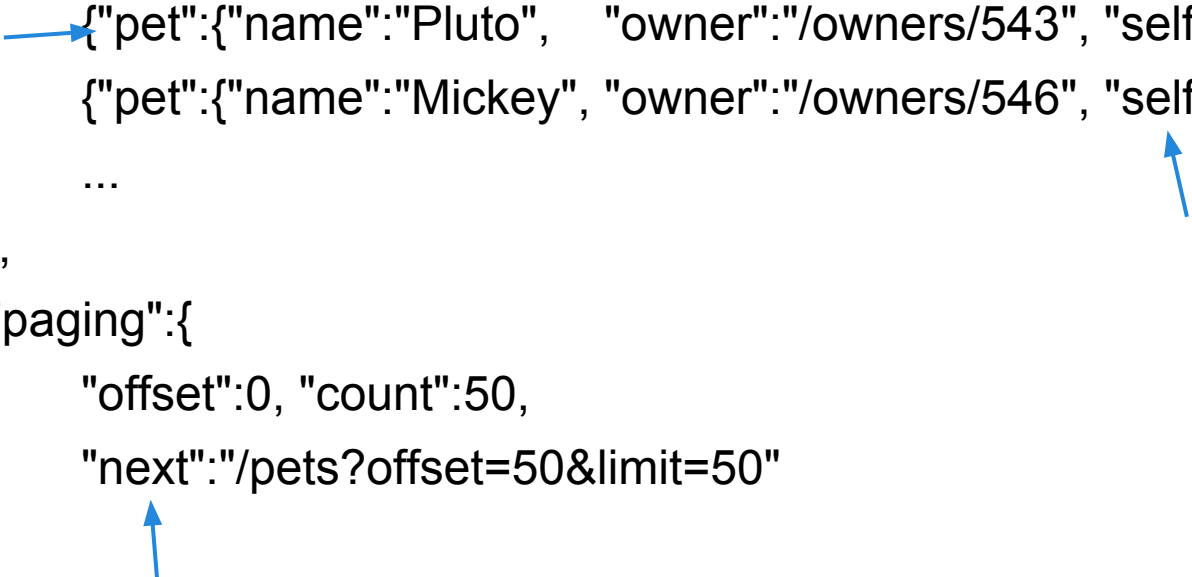
- Stejná pro PUT, POST, GET, list
GET /pets/123

```
{
  "pet": {
    "name": "Pluto",
    "owner": "/owners/543",
    "self": "/pets/123"
  }
}
```

Seznam

GET /pets

```
{
  "pets":{
    "items":[
      →{"pet":{"name":"Pluto", "owner":"/owners/543", "self":"/pets/1" }},
        {"pet":{"name":"Mickey", "owner":"/owners/546", "self":"/pets/2" }},
        ...
    ],
    "paging":{
      "offset":0, "count":50,
      "next":"/pets?offset=50&limit=50"
    }
  }
}
```



Cvičení

- Jak zobrazit všechny mazlíčky a k nim jméno a příjmení majitele?

Možnosti

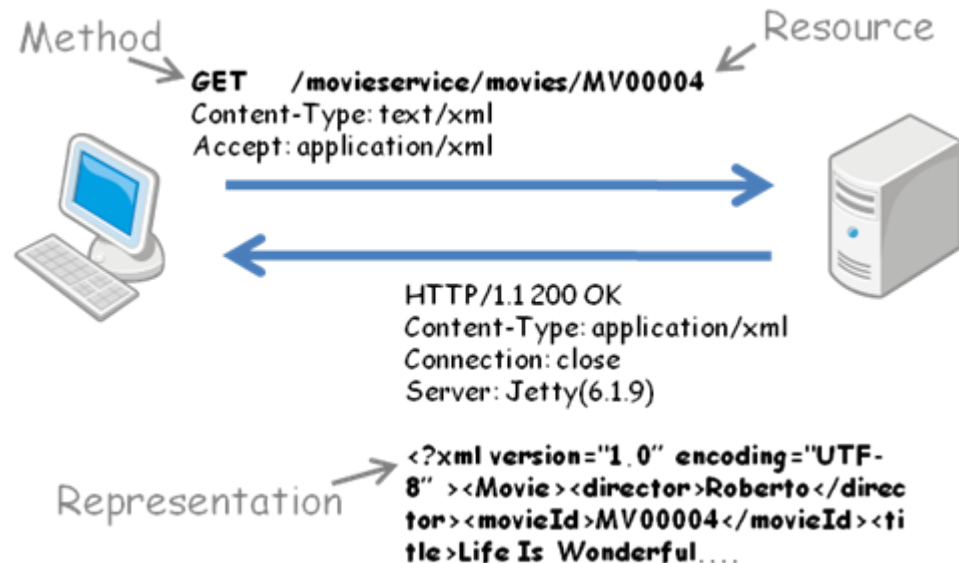
- Čisté řešení (nezávislé resources)
 - Snadný update
 - Oddělené služby
 - N+1 problém
 - Sideloading
- Informace o majiteli v entitě **pet**
 - Ošklivé
 - Co všechno zahrnout?
 - Co se změnami?
 - Cache?
 - `/pets?fields=name,owner`

Sideloading

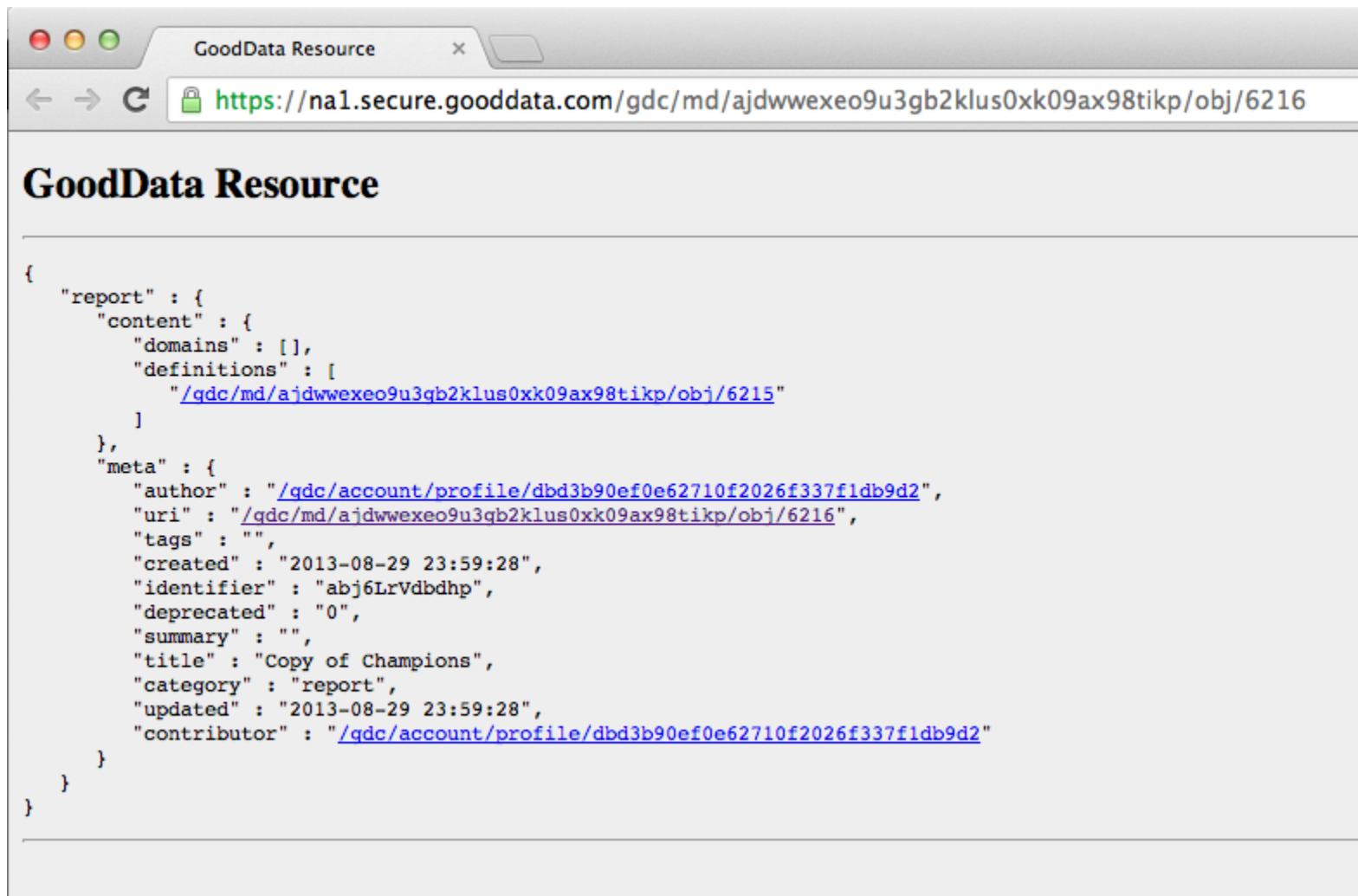
```
{ "pets": [  
  "pet": {  
    "name": "Pluto",  
    "owner": "/owners/543",  
    "self": "/pets/1"  
  },  
  ...  
],  
"owners": [  
  { "owner": { "name": "John Doe",  
    "pets": "/pets?owner=543"}, "self": "/owners/543"},  
  ...  
] }
```

Reprezentace - jaký formát?

- XML, JSON, HTML, ~~YAML~~
- Content negotiation
 - Accept
 - Content-Type



HTML



The screenshot shows a web browser window with the title "GoodData Resource" and the URL <https://na1.secure.gooddata.com/gdc/md/ajdwwexeo9u3qb2klus0xk09ax98tikp/obj/6216>. The page content is a JSON object representing a report resource.

```
{
  "report" : {
    "content" : {
      "domains" : [],
      "definitions" : [
        "/gdc/md/ajdwwexeo9u3qb2klus0xk09ax98tikp/obj/6215"
      ]
    },
    "meta" : {
      "author" : "/gdc/account/profile/dbd3b90ef0e62710f2026f337f1db9d2",
      "uri" : "/gdc/md/ajdwwexeo9u3qb2klus0xk09ax98tikp/obj/6216",
      "tags" : "",
      "created" : "2013-08-29 23:59:28",
      "identifier" : "abj6LrVdbdhp",
      "deprecated" : "0",
      "summary" : "",
      "title" : "Copy of Champions",
      "category" : "report",
      "updated" : "2013-08-29 23:59:28",
      "contributor" : "/gdc/account/profile/dbd3b90ef0e62710f2026f337f1db9d2"
    }
  }
}
```

Možnosti

- application/xml
- application/json

- application/vnd.petclinic.pets+json

- application/vnd.collection+json
- application/vnd.siren+json
- ...

Odkazy

- Naivní

```
"owner": "/owners/543"
```

- HAL

```
"owner": {"href": "/owners/543"}
```

- Collection+JS

```
"links" : [  
  {"rel" : "owner", "href" : "/owners/543"}  
]
```

- Siren

```
"links": [{  
  "class": [ "item"],  
  "rel": [ "http://x.io/rels/owner" ],  
  "href": "http://api.x.io/owners/543"  
}]
```

HTTP status kódy

- 200 - OK
- 201 - Created
- 202 - Accepted
- 204 - No Content

Chybný požadavek

- 400 Bad Request
- 403 Forbidden
- 404 Not Found
- 405 Method not allowed
- 418 I am a teapot (RFC 2324)

Chyba serveru

- 500 Internal Server Error
- 503 Service Unavailable

Redirect

- 301/302/303/307 - Redirect - který je ten správný?
- Všechno kromě HEAD a GET musí být potvrzeno uživatelem

<http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>

Chybí

- Redirect na POST, PUT, DELETE
- Chyby co nejsou způsobeny ani requestem ani serverem
 - Chyba třetí strany
 - Chybný stav

HTTP status

- Nestačí, musíte dodat další informace
 - kód chyby, zpráva, parametry
- HTTP nebylo navrženo pro aplikace
- Není nutné nad tím moc dumat
 - Klienti to stejně většinou ignorují

Cvičení - asynchronní volání

- REST API na spouštění a čtení výsledků operací, které mohou trvat několik desítek minut.

Dokumentace

- WSDL x pravý RESTafarián dokumentaci nepotřebuje
- Samonavigující se klient

Možnosti

- Apiary.io
- Swagger
- Samodomo

JSON schéma

- Schéma je pro bačkory x schéma se hodí
- JSON schema
- Orderly JSON
- RelaxNG :-)
- Samodomo

Zpětná kompatibilita

- Přidání resource
- Přidání volitelného parametru
- Přidání volitelného prvku do requestu
- Přidání prvku do odpovědi?

Verzování

- Verze v URI `/v1/pets/123`
- URI parametr
- Content negotiation
 - `Accept: application/json;v=2`
 - `Content-Type: application/json;v=2`
- Hlavička
- Neverzovat

Shrnutí

- REST je užitečný
- HTTP protokol je zároveň dar a prokletí
- Nikdo vám neřekne jak to správně udělat
 - musíte si najít vlastní variantu
- Umění, ne rutina
 - rovnováha mezi čistotou, použitelností, rychlostí, jednoduchostí, rozšiřitelností, ...

Zdroje

http://blog.apigee.com/detail/announcement_new_ebook_on_web_api_design

