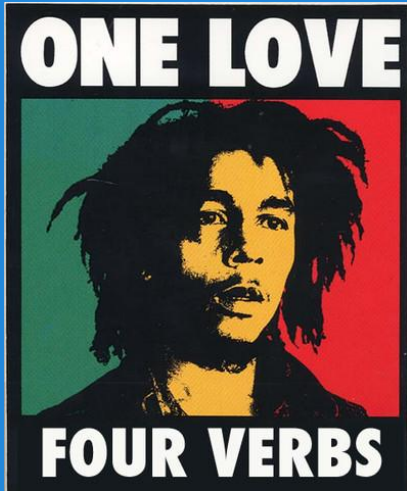


# REST APIs

Lukáš Křečan

REST API Architect

GoodData



Make Web no RPC!

# Resource

/pets/123 % supports GET, PUT, DELETE

```
{
  "pet": {
    "name": "Pluto",
    "owner": "/owners/543",
    "self": "/pets/1"
  }
}
```

# What about list?

/pets % supports POST, GET

```
{
  "pets":{
    "items":[
      {"pet":{"name":"Pluto", "owner":"/owners/543", "self":"/pets/1" }},
      {"pet":{"name":"Mickey", "owner":"/owners/543", "self":"/pets/2" }},
      ...
    ],
    "paging":{
      "offset":0,
      "count":50,
      "next":"/pets?offset=100&limit=50"
    }
  }
}
```

# Visits?

```
{
  "pet": {
    "name": "Pluto",
    "owner": "/owners/543",
    "self": "/pets/1",
    "visits": [
      {"visit": {"date": "2012-12-01", "note": "Another long text2"}},
      {"visit": {"date": "2011-12-01", "note": "This is long text1"}},
      ...
    ]
  }
}
```

# Better

```
{  
  "pet": {  
    "name": "Pluto",  
    "owner": "/owners/543",  
    "self": "/pets/1",  
    "visits": "/pets/1/visits"  
  }  
}
```

# REST Highlights

- Every “thing” has an URI
- Link things together
- Use standard HTTP methods
- HTTP caches
- Simple

# REST Downsides

- No documentation standard
- Inconsistent best practices
- Mismatch between HTTP standard and your needs



# Resources

[http://blog.apigee.com/detail/announcement\\_new\\_ebook\\_on\\_web\\_api\\_design](http://blog.apigee.com/detail/announcement_new_ebook_on_web_api_design)

