



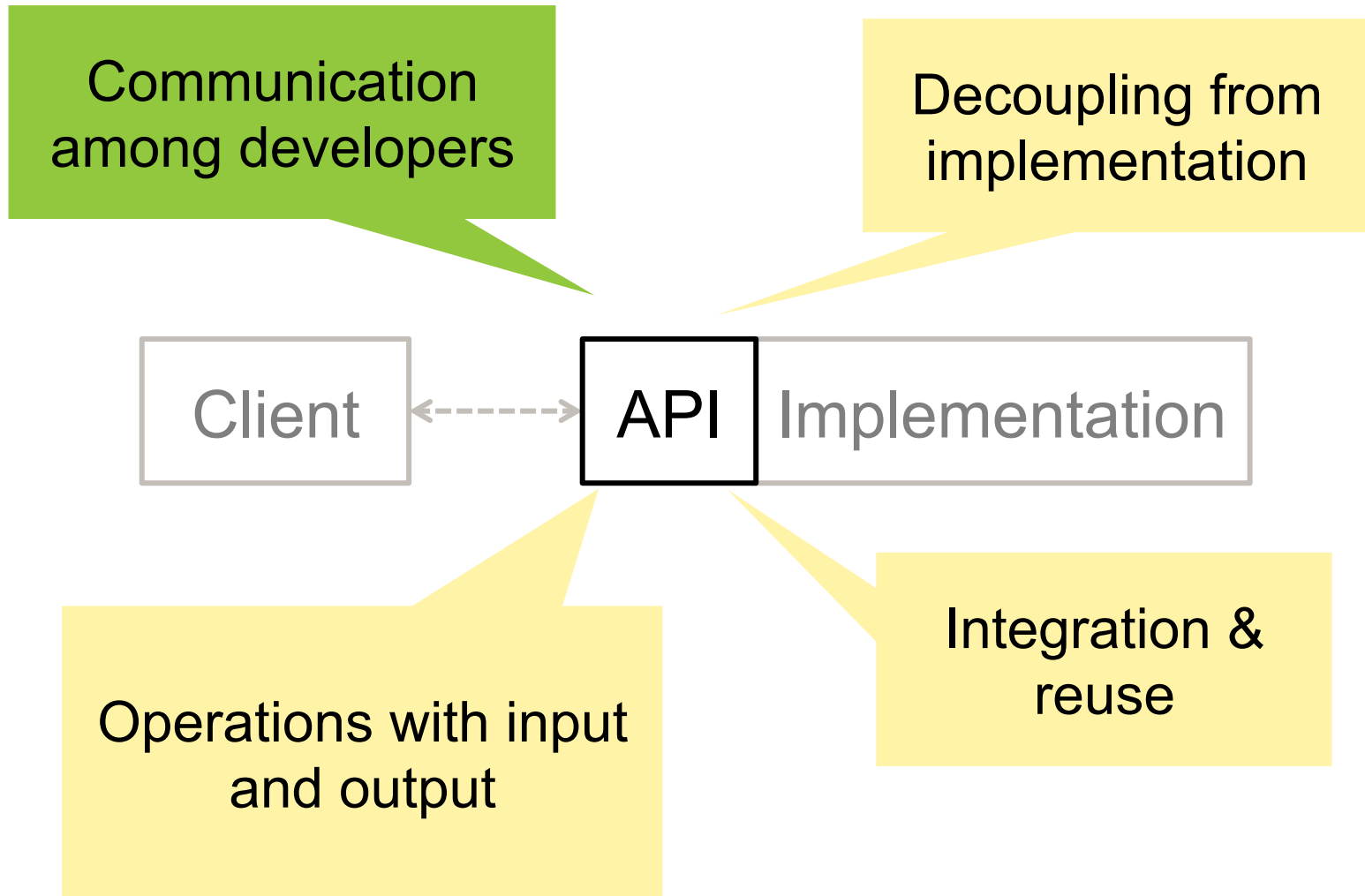
**Click 'engage'
to rate sessions
and ask questions**



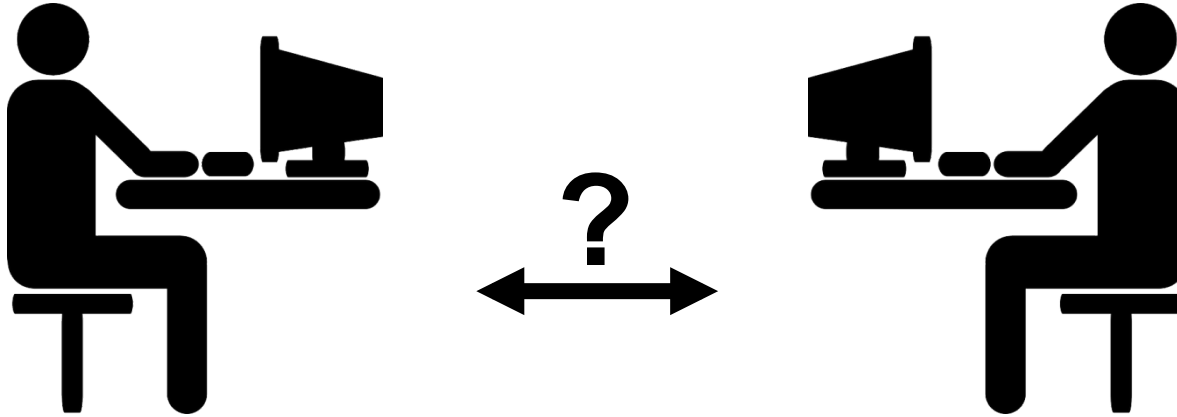
Web APIs that Developers Love

Kai Spichale @kspichale





Developer Perspective



- ▶ APIs are intended for developers
- ▶ Use standard HTTP
- ▶ Document precisely & with examples

URI Design for Web APIs

Unique identifier of resources

`/answers/42`

Could be cryptic

`/xc487xdjfnngx71dd`

Client should not create links by
himself

Server provides links

Hypermedia links (HATEOAS)

```
"links": [  
  {  
    "href": "https://api.sandbox.paypal.com/v1/  
            payments/payment/PAY-2XR80EBWOSA",  
    "rel": "self",  
    "method": "GET"  
  }  
]
```

Should You Care about URI Design?

No!

REST does not require proper URI design

Yes!

Many so-called „REST“ APIs ignore hypermedia

Readable URIs help developers to understand the API

Advantages of URI Design

Readable

`https://api.github.com/users/mojombo/followers`

Hackable

`https://api.github.com/users`

`https://api.github.com/users/kspichale`

Stable

No broken links

HTTP 301 «Moved Permanently»

Best Practices for URIs

Short

Easier to read

Abbreviations can be counterproductive

Consistent

Patterns, names, case

Plural nouns

Strict lowercase

Hyphen

Don't Be Dogmatic – Pseudo File Endings

File of a web app

`http://www.ics.uci.edu/~fielding/pubs/
dissertation/top.htm`

Resource of a
Web API

`http://userstream.twitter.com/1.1/user.json`
(There is no content negotiation anyway.)

Don't Be Dogmatic – Trailing Slashes

`/foo/bar` Unix file

`/foo/bar/` Unix directory

`/books/`

`/books/42`

`/books/42/chapters`

`/books/42/chapters/1`

Be consistent:
Avoid trailing
slashes

Don't Be Dogmatic – Trailing Slashes

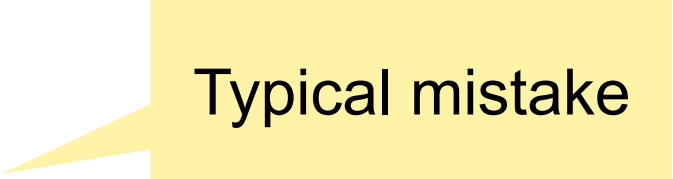
Directory resource /

Provides navigation links to begin the interaction with the API

`http://api.example.com/`

vs.

`http://api.example.com`



Typical mistake

1:* Relations

Hierarchical URI pattern

`:collection/:item/:sub-collection/:sub-item`



POST `/groups/g1/persons`

GET `/groups/g1/persons/p1`

DELETE `/groups/g1/persons/p1`

n:m Relations



GET /groups/g1

```
{
  "id": "p1",
  "persons": [ "p1", "p2" ]
}
```

GET /persons/p1

```
{
  "id": "g1",
  "groups": [ "g1", "g2" ]
}
```

Caching?

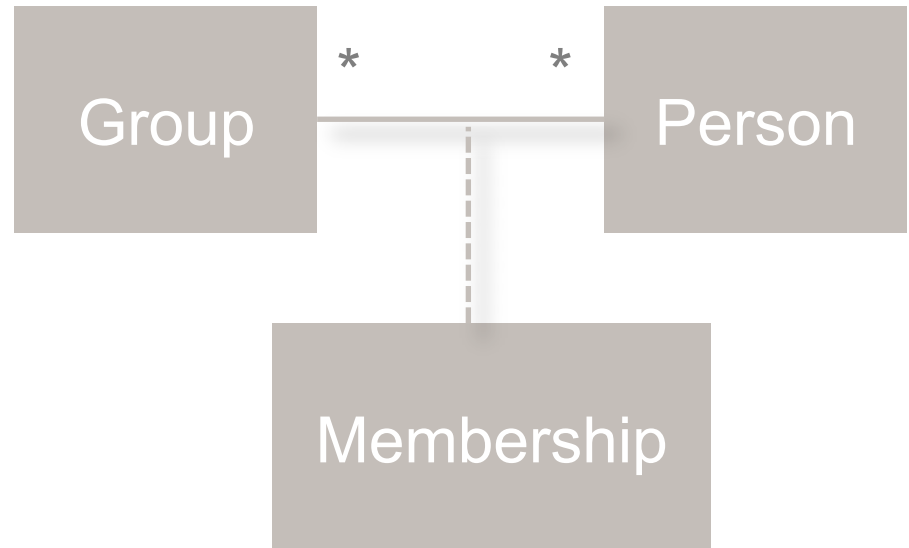
Which one is leading?

n:m Relations with Subscriptions

```
POST /memberships
{
  "person_id": "p1",
  "group_id": "g1",
}
```

```
GET      /membership/mem-p1-g1
```

```
DELETE  /membership/mem-p1-g1
```



Caching!

Atomicity!

Additional
properties!

Best Practices for IDs – Prefixes

Makes URI more readable

`/agents/007` vs. `/agents/agent007`

`/persons/83j9x` vs. `/persons/per-83j9x`

`https://jira.spring.io/browse/DATAREST-516`

Best Practices for IDs – Sequential IDs

`https://jira.spring.io/browse/DATAREST-516`

Guessable (advantage):

There must be a ticket 515

Hackable (advantage):

Quick navigation to another ticket

Best Practices for IDs – Sequential IDs

`http://api.super-kewl-app.com/users/u-314`

Guessable (disadvantage):

There must be 313 other users

Hackable (disadvantage):

Potential security issue

No 1:1 relation between database key and public ID

Best Practices for IDs – Slug IDs

`http://stackoverflow.com/questions/47427/why-
-do-some-websites-add-slugs-to-the-end-of-urls`

Natural key

Created by URL encoding

Search Engine Optimization

Read content in link before clicking on it

Partial Responses

Partial Responses – Paging



Partial Responses – Web Linking (RFC5899)

```
"links": [  
  {  
    "href": "http://api.example.com/products?page=1&count=5",  
    "rel": [ "previous" ],  
    "method": "GET"  
  },  
  {  
    "href": "http://api.example.com/products?page=2&count=5",  
    "rel": [ "self" ],  
    "method": "GET"  
  },  
  {  
    "href": "http://api.example.com/products?page=3&count=5",  
    "rel": [ "next" ],  
    "method": "GET"  
  }  
]
```

Partial Responses – Web Linking (RFC5899)

```
"links": [  
  {  
    "href": "http://api.example.com/products?page=1&count=5",  
    "rel": [ "first" ],  
    "method": "GET"  
  },  
  {  
    "href": "http://api.example.com/products?page=23&count=5",  
    "rel": [ "last" ],  
    "method": "GET"  
  }  
]
```


Partial Responses – Streams of Data

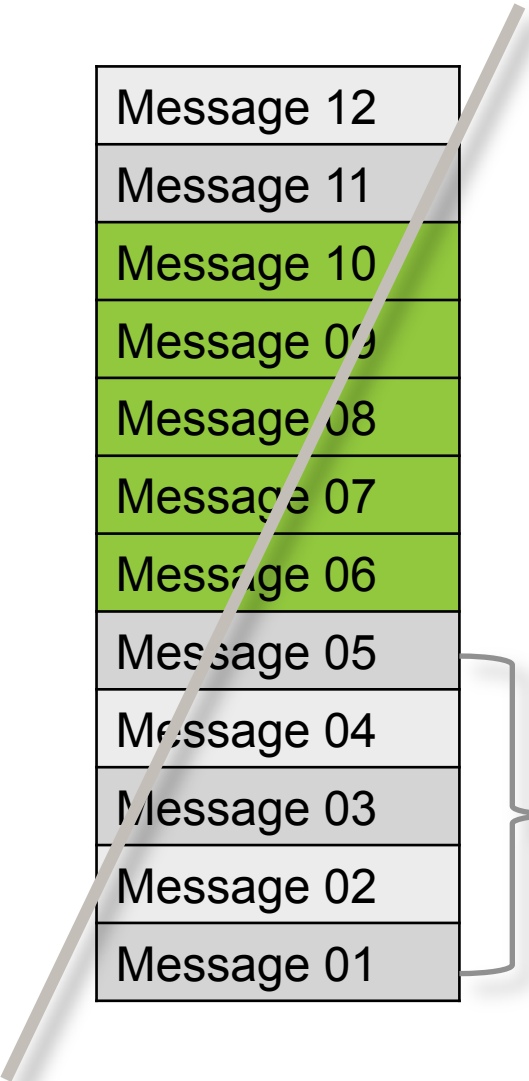
Message 10
Message 09
Message 08
Message 07
Message 06
Message 05
Message 04
Message 03
Message 02
Message 01

page=1
count=5

Message 12
Message 11
Message 10
Message 09
Message 08
Message 07
Message 06
Message 05
Message 04
Message 03
Message 02
Message 01

processed

Partial Responses – Streams of Data



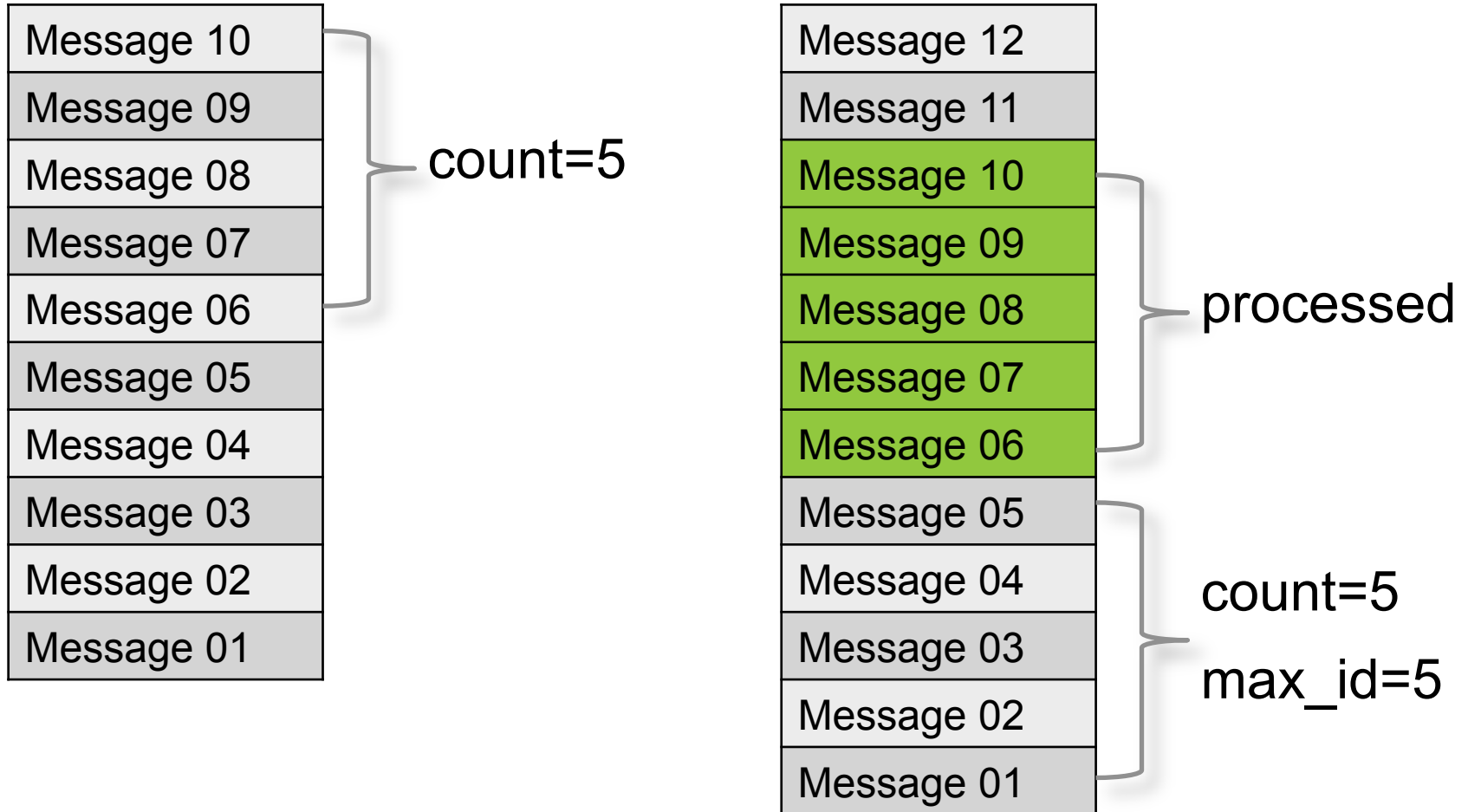
Message 12
Message 11
Message 10
Message 09
Message 08
Message 07
Message 06
Message 05
Message 04
Message 03
Message 02
Message 01

page=2
count=5

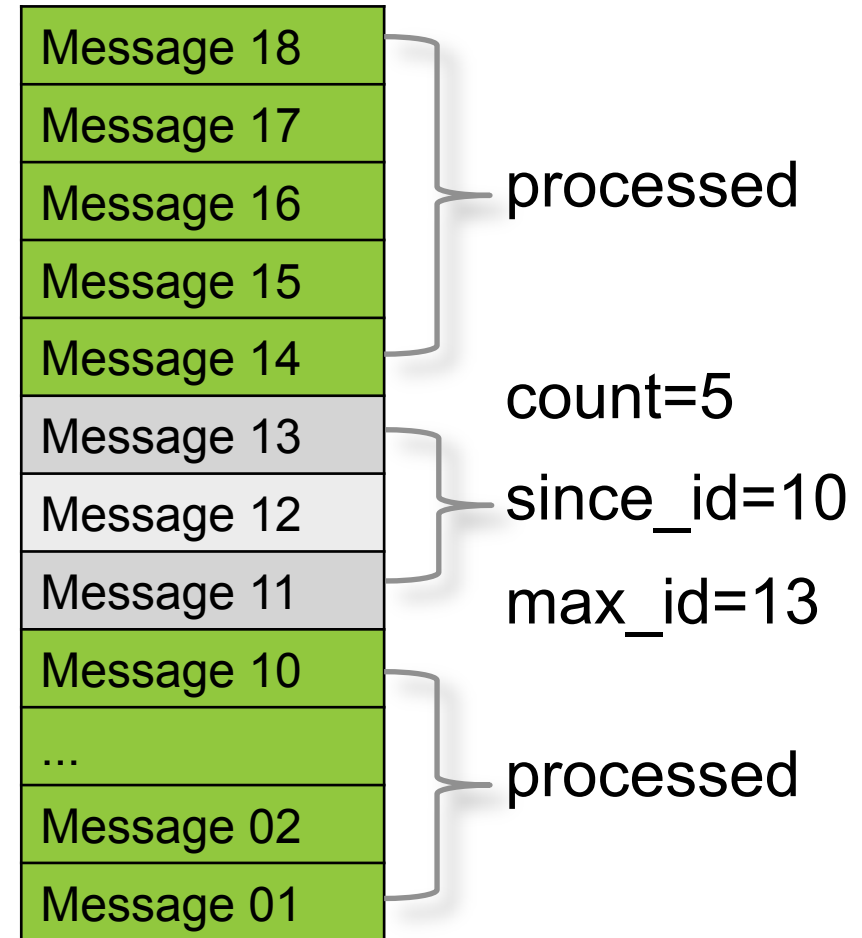
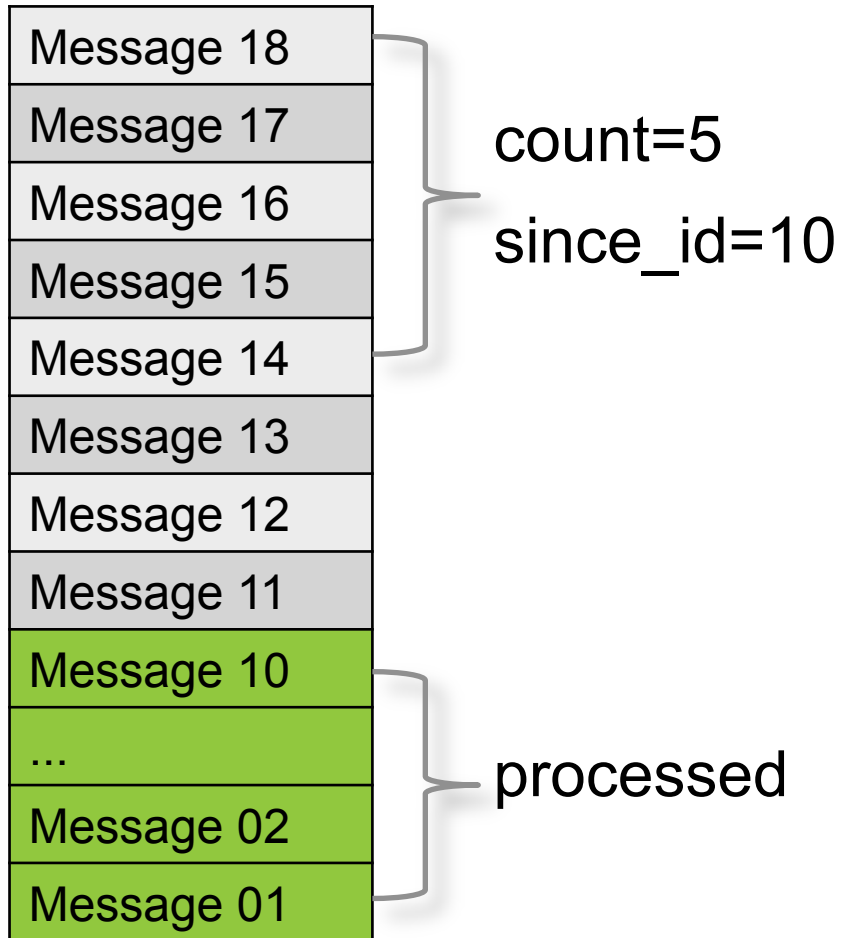
Message 12
Message 11
Message 10
Message 09
Message 08
Message 07
Message 06
Message 05
Message 04
Message 03
Message 02
Message 01

page=2
count=5

Partial Responses – Cursoring



Partial Responses – Cursoring



Siren – Embedding vs. Linking

```
{  
  _links: {  
    "self": { "href": "/books/42" },  
    "author": { "href": "/people/douglas-adams" }  
  }  
}
```

Client may resolve links to sub-entities

Siren – Embedding vs. Linking

```
{
  "_links": {
    "self": { "href": "/books/42" },
    "author": { "href": "/people/douglas-adams" }
  },
  "_embedded": {
    "author": {
      "_links": { "self": { "href":
                          "/people/douglas-adams" } },
      "name": "Douglas Noel Adams",
      "born": "March 11, 1952",
      "died": "May 11, 2001"
    }
  }
}
```

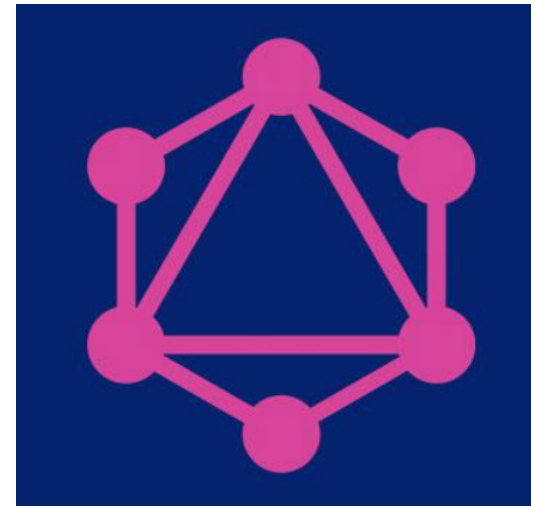
GraphQL – A Data-Fetching API

A GraphQL query is sent as a string to a server to define the shape of the returned representation

Developed by Facebook

<http://graphql.org>

Strongly typed



GraphQL – A Data-Fetching API

```
{  
  author(id:douglas-adams) {  
    name  
  }  
}
```

Server can return exactly what a client asks for and no more:

```
{  
  "author": {  
    "name": "Douglas Noel Adams"  
  }  
}
```


GraphQL Selections

Fields

```
{  
  id  
  firstname  
  lastname  
}
```

Complex Data

```
{  
  releaseDate {  
    month  
    year  
  }  
}
```

Arguments

```
{  
  book(id:42) {  
    id  
    title  
    coverPic(size:100)  
  }  
}
```

GraphQL Selections

Using the spread operator (...) and fragments

```
fragment bookFields on Book {  
  id  
  title  
  coverPic(size:100)  
}
```

```
query bookQuery {  
  book(id:42) {  
    ...bookFields  
  }  
}
```

GraphQL Mutations

```
Mutation {  
  likeBook(bookId: 42) {  
    book {  
      name  
      likeCount  
    }  
  }  
}
```

API Changes

GET /books/42

Response:

```
{  
  "id": 42,  
  "title": "The Hitchhiker's Guide to  
           the Galaxy",  
  "price": "7.95"  
}
```

API Changes

GET /books/42

Response:

```
{  
    "id": 42,  
    "title": "The Hitchhiker's Guide to  
             the Galaxy",  
    "bruttoPrice": "7,95",  
    "nettoPrice": "6,12"  
}
```

API Changes

GET /books/42

Response:

```
{  
  "id": 42,  
  "title": "The Hitchhiker's Guide to  
           the Galaxy",  
  "price": "7.95",  
  "bruttoPrice": "7,95",  
  "nettoPrice": "6,12"  
}
```

GraphQL Introspection API

```
GET /books/42
```

```
query BookIntrospection {  
  __type(name: "Book") {  
    name  
    fields(includeDeprecated: true) {  
      name  
      isDeprecated  
      deprecationReason  
    }  
  }  
}
```

GraphQL Introspection API

```
{ "data": {  
  "__type": {  
    "name": "Book",  
    "fields": [  
      {  
        "name": "id",  
        "isDeprecated": false,  
        "deprecationReason": null  
      },  
      {  
        "name": "price",  
        "isDeprecated": true,  
        "deprecationReason":  
          "Use 'bruttoPrice' instead."  
      },  
    ]  
  }  
}
```


Compatible Extension

Old client:

```
query bookQuery {  
  book(id:42) {  
    id  
    title  
    price  
  }  
}
```

New client:

```
query bookQuery {  
  book(id:42) {  
    id  
    title  
    bruttoPrice  
  }  
}
```

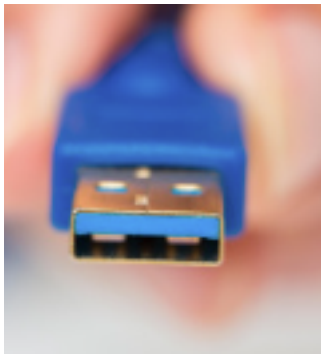
Data Versioning vs. Language Versioning

Backward compatibility



USB 3.0 ports support
USB 2.0 sticks

Forward compatibility



USB 2.0 ports support
USB 3.0 sticks

	Consumer	Producer
--	-----------------	-----------------

Backward Compatibility

Adding new resources and HTTP methods

Extensible data formats

Requests: new optional elements can be omitted by clients

Responses: new elements can be problematic for clients

A yellow callout box with a pointed top, containing the text "Forward compatibility".

Forward
compatibility

Forward Compatibility

Postel's law (robustness principle)

"Be conservative in what you send,
be liberal in what you accept."

Error Handling

Black box from user's perspective

Common problems:

Error tunneling through 200 «OK»

Generic 500 «Internal Server Error»

HTTP status codes

1xx Informational

4xx Client Error

2xx Success

5xx Server Error

3xx Redirection

Most Common HTTP Error Codes

400 «Bad Request»

404 «Not Found»

403 «Forbidden»

500 «Internal Server Error»

503 «Service Unavailable»

504 «Gateway Timeout»

Proprietary Error Codes

Twitter API

```
{
  "errors" : [
    {
      "message" : "Sorry, that page does not exist",
      "code" : 34
    }
  ]
}
```

Error Codes

In addition to descriptive error text, error messages contain machine-parseable codes. While the text for an error message may change, the codes will stay the same. The following table describes the codes which may appear when working with the API:

Code	Text	Description
32	Could not authenticate you	Your call could not be completed as dia
34	Sorry, that page does not exist	Corresponds with an HTTP 404 - the specified resource was not found.
64	Your account is suspended and is not permitted to access this feature	Corresponds with an HTTP 403 — the access token being used belongs to a suspended user and they can't complete the action you're trying to take

<https://dev.twitter.com/overview/api/response-codes>

Proprietary Error Codes

Twilio API

```
{  
  "status": 400,  
  "message": "No 'To' number is specified",  
  "code": 21201,  
  "more_info": "http://www.twilio.com/docs/←  
                errors/21201"  
}
```

Error Reference

[Overview](#)[Error Code Listing](#)[App Monitor Triggers](#)

ERROR - 21201

No 'To' number specified

You attempted to initiate an outbound phone call, but you did not specify what number to call in the 'To' POST parameter



<https://www.twilio.com/docs/api/errors/21201>



Please

**Remember to
rate session**

Thank you!

Kai Spichale @kspichale

