

Building the API – Implementing Advanced Requirements



Kevin Dockx

@KevinDockx | <http://blog.kevindockx.com/>



The API needs to support data shaping to allow including associated resources in one call

GET



api/expensegroups/1

expensegroup

GET



api/expensegroups/1/expenses/1

expense

GET



api/expensegroups/1/expenses/2

expense

GET



...

expense

GET



api/expensegroups/1
?fields=expenses,...

expensegroup

expense



The API needs to support data shaping to allow selection of specific resource fields

GET



api/expensegroups/1

expensegroup

id

userid

title

description

expensgroupstatusid

expenses

GET



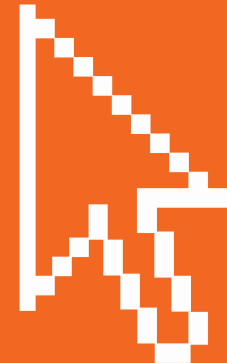
api/expensegroups/1?fields=id

expensegroup

id

Data Shaping, Part 1

Learn how to allow selection of
specific resource fields



Data Shaping, Part 2

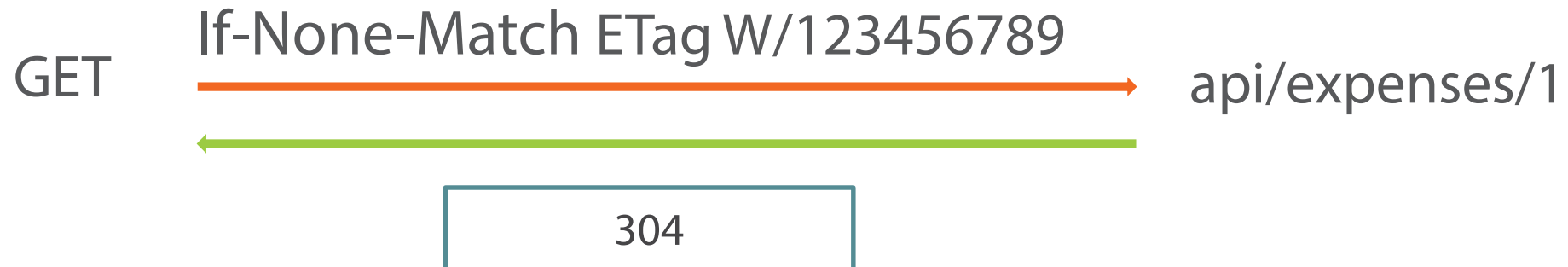
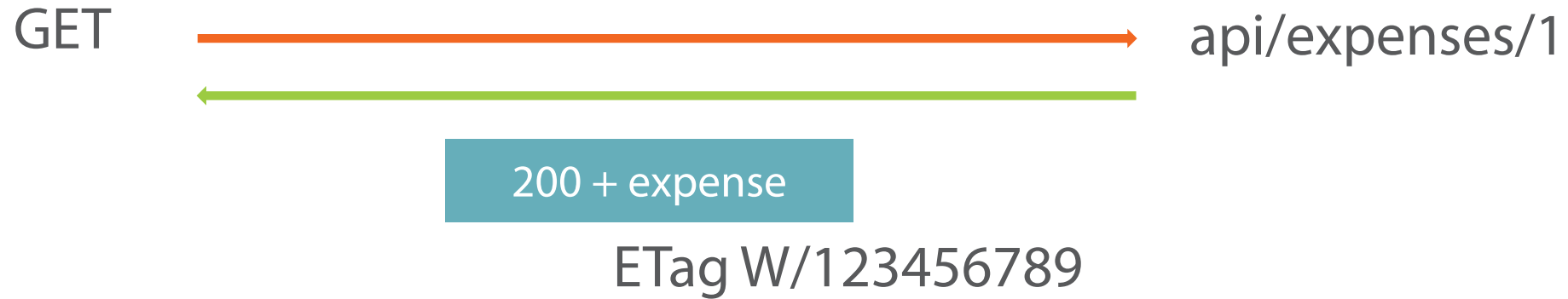
Learn how to include associated resources in one call



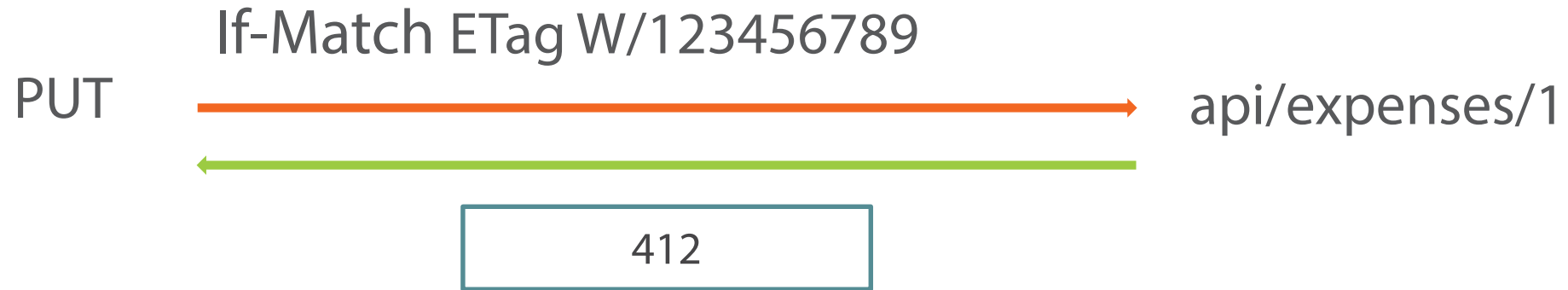


The API needs to support caching

Caching: ETags



Caching: ETags



- Quite a heavy burden on the server...
- ... and on the client as well
- NuGet package: **CacheCow.Server**

Caching

Learn how to support caching at API level



Breaking REST with Output Caching

- *“Why is there no OutputCache support for Web Api?”*



- It's not according to REST principles (no HTTP Cache), but it does put less burden on the client
- NuGet package: **Strathweb.CacheOutput.WebApi2**



The API mustn't break when
changes are rolled out

Versioning

Through the URI

- Version number in URI
- URI maps to method on controller
- But...
 - Shouldn't a URI to the same resource stay the same?
 - ... and if the resource changes, isn't that essentially *another* resource?

```
HTTP GET  
http://domain/api/expenses/1
```

```
Accept: application/json
```

```
HTTP GET  
http://domain/api/v2/expenses/1
```

```
Accept: application/json
```

Versioning

Through content negotiation

- Custom Content Type in the Accept Header
- vnd.myservice.v2+json

```
HTTP GET  
http://domain/api/expenses/1
```

```
Accept: application/json
```

```
HTTP GET  
http://domain/api/expenses/1
```

```
Accept:  
application/vnd.myservice.v2+json
```

Versioning

Through a custom request header

- Custom header added to request, containing the version number

```
HTTP GET  
http://domain/api/expenses/1
```

```
Accept: application/json
```

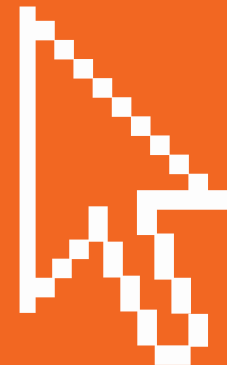
```
HTTP GET  
http://domain/api/expenses/1
```

```
Accept: application/json
```

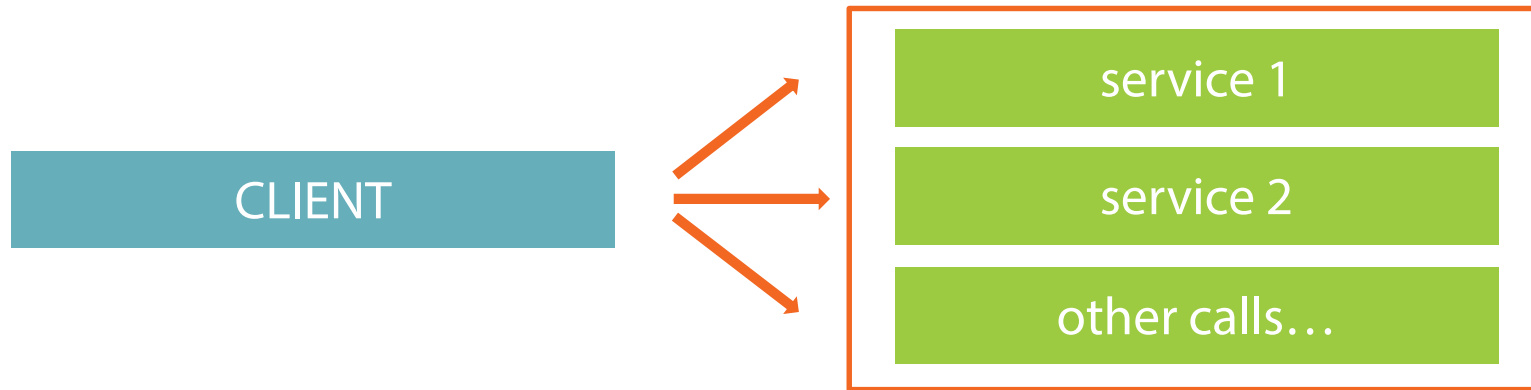
```
api-version: 2
```


Versioning

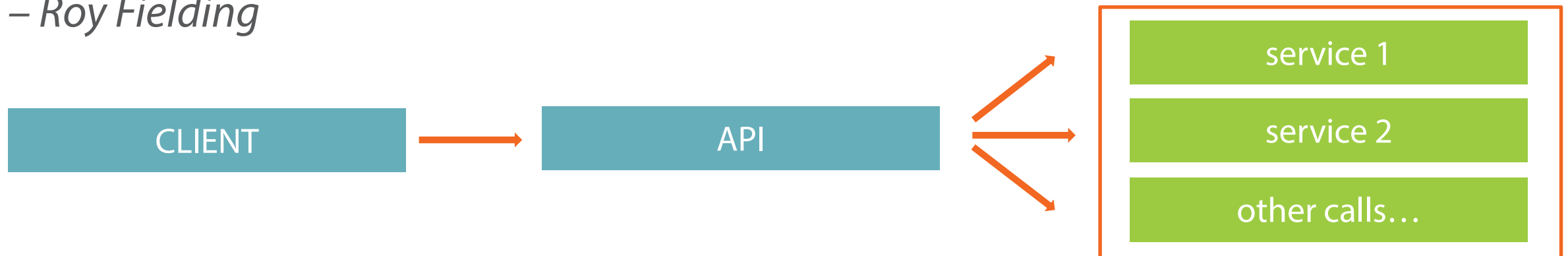
Learn how to implement different versioning strategies



What About Transactions?



“A REST transaction is an oxymoron”
– Roy Fielding



Summary



Data shaping allows better control over what's returned

Caching can greatly increase performance, but comes at a price

Versioning is essential

REST transactions are an oxymoron