



Design Practices

REST API Design, Development & Management

<http://www.acloudfan.com>

raj@acloudfan.com

Slides/Discussion from a course that covers the
A to Z of RESTful API. More information.

Updated: Dec 27th, 2016

Discounted access to the courses:

raj@acloudfan.com
<http://www.acloudfan.com>



<https://www.udemy.com/ibm-bluemix/?couponCode=BLUE100>

Coupon Code = **BLUE100**



<https://www.udemy.com/rest-api/?couponCode=REST100>

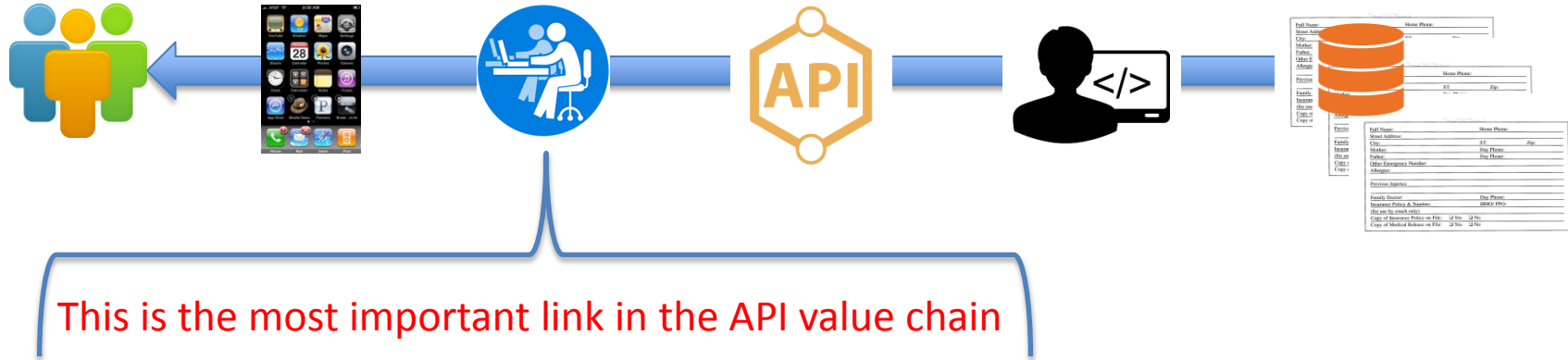
Coupon Code = **REST100**

PS:

- For latest coupons & courses please visit: <http://www.acloudfan.com>
- Enter to **WIN Free access** – please visit: <http://www.acloudfan.com/win-free-access>

API Value Chain

raj@acloudfan.com
<http://www.acloudfan.com>



Design your API with the needs of App Developers in mind



Resource identity, names, actions



<http://www.acloudfan.com>

raj@acloudfan.com

Summary of a course that covers the A to Z of RESTful API. More information.

[UDEMY Link With Coupon Discount](#)



API endpoint or URL for the resource



`https://api.walmartlabs.com/v1/items`

Resource



Base URL



Version





/items /items/{id}



/people /people/{id}



/programmes /programmes/{id}



API Operation \neq CRUD



U B E R

/estimates/price



/friendships/lookup



/search?query

Summary



1. Create a simple base URL ; subdomain ; separate domain
2. Resources names should be nouns; Use of plurals suggested
3. Actions can be verbs may be standalone

<http://api.acme.com/search>

4. URI Pattern for associations; avoid nesting over 3 levels

<http://api.acme.com/vacations/{id}/reviews/{review-id}>



CRUD Operations



<http://www.acloudfan.com>

raj@acloudfan.com

Part of a course that covers the A to Z of RESTful API. More information.

[UDEMY Link With Coupon Discount](#)



POST

Creates a resource

/vacations

/vacations/{id}/reviews

Success

Code = 201

- May return a link (id) to new resource in Location header
Location: <http://api.acme.com/vacations/121/reviews/2>
- May return the new object

Failure

- | | | |
|-----|---------------------|----------------------------------|
| 4xx | Bad Request | e.g., 400 missing required field |
| 5xx | Issue in processing | e.g., 503 database unreachable |



GET

Reads/Retrieves a resource collection or specific resource

/vacations

/vacations/{id}

Success

Code = 200 OK

- Send back the response in requested format

GET http://api.acme.com/vacations/121

Failure

4xx Bad Request

e.g., 404 Resource not found

5xx Issue in processing

e.g., 500 Internal Server Error



PUT

Updates all attributes of existing resource – effectively replace
Can also CREATE if client provide ID (not suggested)

PATCH

Modifies parts of an existing resource.

PATCH <http://api.acme.com/vacations/121?validTill=3/1/2017>

Success

Code = 200

Code = 204 No Content

Code = 201 Created

- Resource in body of response – OPTIONAL
- No need to send the Link – OPTIONAL

Failure

4xx Bad Request

e.g., 404 Resource Not Found

5xx Issue in processing

e.g., 503 database unreachable



DELETE

Deletes the resource

/vacations

/vacations/{id}

Success

Code = 200

Code = 204

- May return deleted resource in the response body
- No returned content = 204

Failure

4xx Bad Request

e.g., 404 Resource Not Found

5xx Issue in processing

e.g., 503 database unreachable



API Response

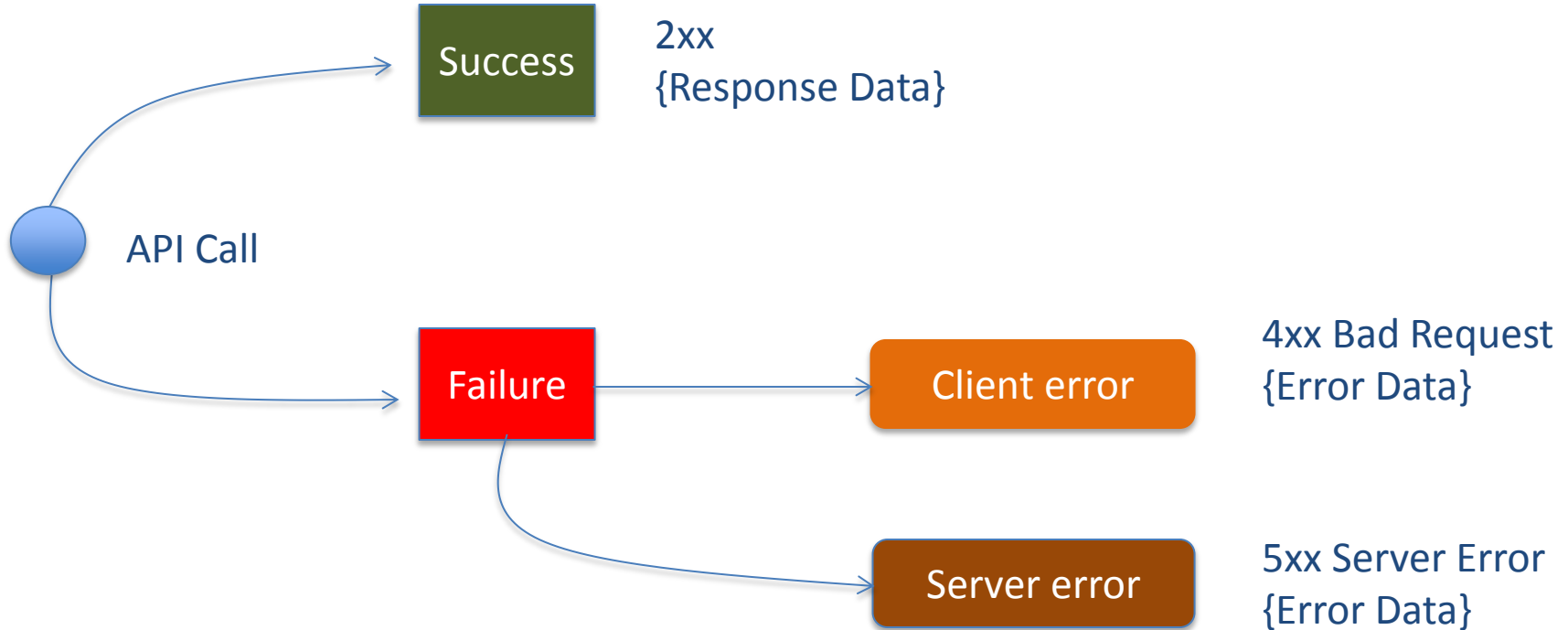


<http://www.acloudfan.com>

raj@acloudfan.com

Part of a course that covers the A to Z of RESTful API. More information.

[UDEMY Link With Coupon Discount](#)





- Over 70 status code that are hard to remember
- Decide on the code that you will use for your API
- Common ones

200 OK

400 Bad Request

500 Server Error

201 Created

404 Not Found

401 Unauthorized

403 Forbidden

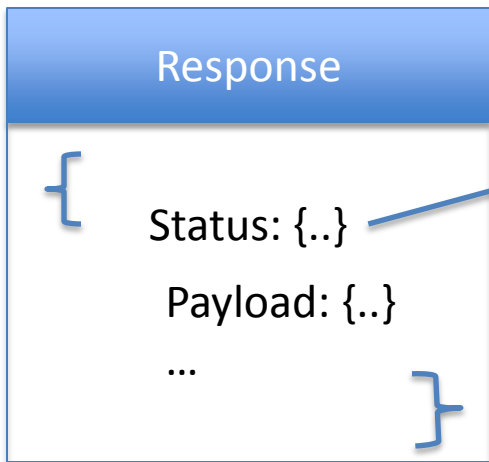
415 Unsupported
media

An Alternate Approach to Error Response

raj@acloudfan.com
<http://www.acloudfan.com>



- Always sends back HTTP Status = 200 OK



```
// Meant for the developer
text:message,
timestamp:new Date(),
// POST, GET ....
method:httpMethod,
// Endpoint information
endpoint:endpointInformation,
// An array of all errors
errors : errorList,
// OPTIONAL -
// Use only during development
payload: receivedPayload
```

```
{
  code:7002,
  text:"Required field vacation 'number of nights' is missing",
  hints:["Please check that user has provided a number (between 1 & 31)"],
  info:"http://developer.acme.com/error#RequiredFields"
},
```

```
{
  code:7001,
  text:"Required field vacation 'name' is missing",
  hints:["Please check that user has provided the non null value for 'name'"],
  info:"http://developer.acme.com/error#RequiredFields"
}
```

facebook



Handling Change



<http://www.acloudfan.com>

raj@acloudfan.com

Summary/Part of a course that covers the A to Z of RESTful API. More information.

[UDEMY Link With Coupon Discount](#)



- Adding a new operation or resource

GET /hotels

- Adding optional parameters or properties

GET /vacations?limit=3

- Change the HTTP verb or methods

PUT /vacations

POST /vacations

- Delete an operation

GET /vacationsByDestination

Non Breaking

Breaking



- Avoid changes : is the change really adding value?
 1. Eliminate or minimize impact on app developers
 2. Provide planning opportunity to the app developers
 3. Support backward compatibility (if possible)
 4. Provide support to app developers with the changes
 5. Minimize change frequency e.g., once per 6 months

Version your API right from day#1



API Versioning



<http://www.acloudfan.com>

raj@acloudfan.com

Summary/Part of a course that covers the A to Z of RESTful API. More information.

[UDEMY Link With Coupon Discount](#)

Version Information

raj@acloudfan.com
<http://www.acloudfan.com>



HTTP Header

x-myapi-version: 1.2

Query
parameter

/posts?version=1.2

facebook.

URL

/v1/products



U B E R

Multiple Version Support

Key points

raj@acloudfan.com

<http://www.acloudfan.com>



1. Support at least 1 previous version for a period of time
 - E.g., 3 months
2. Mark the previous version as deprecated
 - For new apps developers can access only the latest version
3. Publish a roll out plan in advance
4. Manage changelog that clearly shows the reason for new version



Caching



<http://www.acloudfan.com>

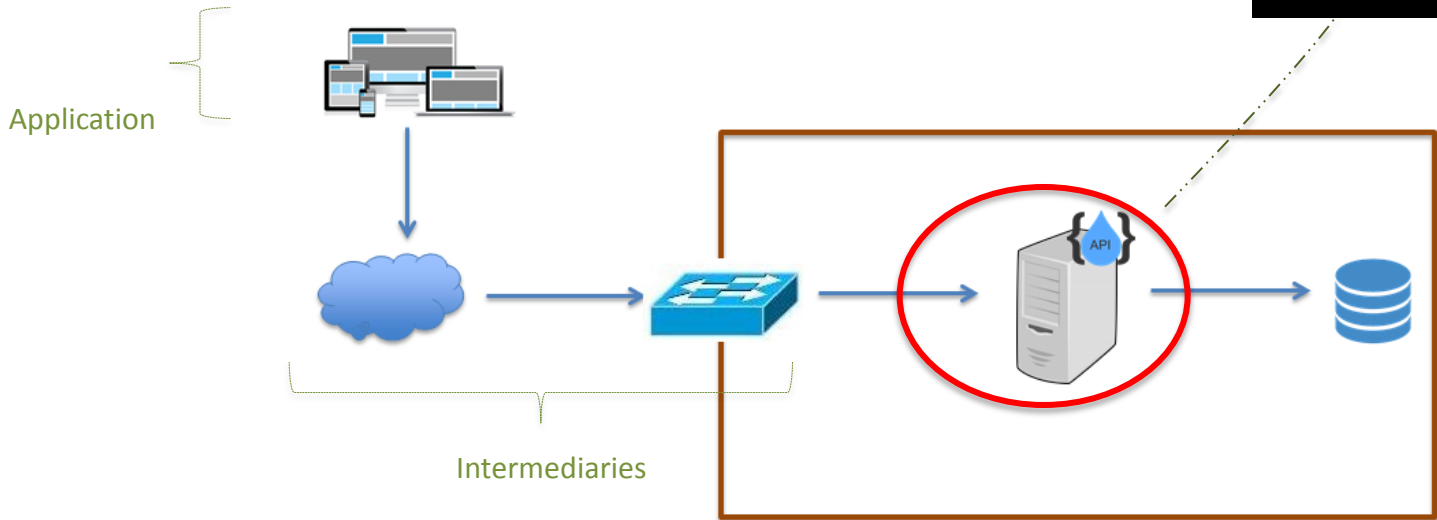
raj@acloudfan.com

Summary/Part of a course that covers the A to Z of RESTful API. More information.

[UDEMY Link With Coupon Discount](#)

- 1. Which component should control the caching?
- 2. What to cache? Who can cache?
- 3. For how long is the cached data valid?

HTTP Cache Control Directives



Summary



1. Benefits

- Enhances performance
- Leads to higher scalability

2. Data to cache depends on

Speed of change

Time sensitivity

Security

3. Design decisions

- Who can cache?
- For how long?



HTTP Cache-Control



<http://www.acloudfan.com>

raj@acloudfan.com

Summary/Part of a course that covers the A to Z of RESTful API. More information.

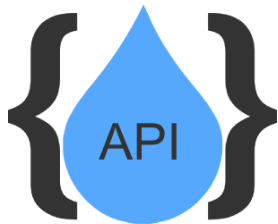
[UDEMY Link With Coupon Discount](#)

Cache-Control



Cache-Control : “ directive-1, directive-2, ...”

Cache-Control : “private, max-age=60”



Response HTTP Header
Cache-Control: “.....”

- Who can cache the response?
- For how long?
- Under what conditions?



Request HTTP Header
Cache-Control: “.....”

- Override the caching behavior
- Protect sensitive data from caching



1. Take advantage of caching especially for high volume API
2. Consider *no-store* and *private* for sensitive data
3. Provide the validation tag (**ETag**) especially for large responses
4. Carefully decide on the optimal *max-age*



Partial Response



<http://www.acloudfan.com>

raj@acloudfan.com

Summary/Part of a course that covers the A to Z of RESTful API. More information.

[UDEMY Link With Coupon Discount](#)



- Better performance & optimized resource usage
 - CPU, Memory, Bandwidth
- API consumer controls the granularity
- Common API version for all consumers
 - E.g., to support multiple devices, use cases – form factors



/people:(id, first-name, last-name)



/friends?fields=id,name,picture



/pins?fields=id,link,creator(first_name)



Pagination



<http://www.acloudfan.com>

raj@acloudfan.com

Summary/Part of a course that covers the A to Z of RESTful API. More information.

[UDEMY Link With Coupon Discount](#)

1. Cursors for Pagination



- **Cursor** = “Control structure that enables traversal of records”
- Cursor based pagination considered most efficient

Cursor=“Random string”

Points to a specific item

Item=10299



facebook.

<https://developers.facebook.com/docs/graph-api/using-graph-api/#paging>

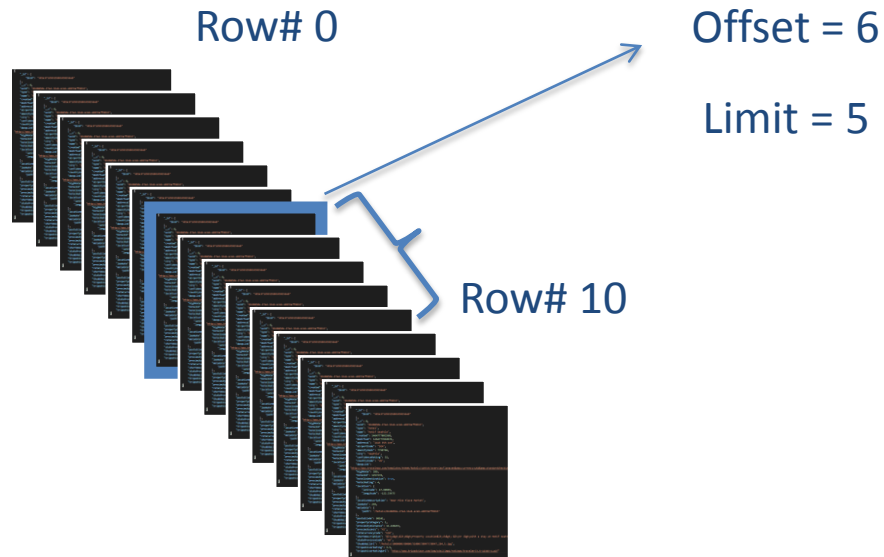
2. Offset based pagination

raj@acloudfan.com
<http://www.acloudfan.com>



- Commonly used approach

GET ../hotels?offset=6&limit=5



3. HTTP Link Header

raj@acloudfan.com
<http://www.acloudfan.com>



<https://api.github.com/search/code?q=addClass>

- Use of Link header *a.k.a. web linking*

<https://tools.ietf.org/html/rfc5988>

HTTP Header

Link to Next page

Link: <https://api.github.com/user/repos?page=3&per_page=100>; rel="next",

<https://api.github.com/user/repos?page=50&per_page=100>; rel="last"

Link to Last page



- Consider pagination for avoiding large data sets
Decide on the default page size. May be different for API(s)
- Support for multiple pagination for different resources