mndwrk

Leveraging UX

In Developing People Centered APIs

Anthony Ejiogu

UX Designer

ExxonMobil

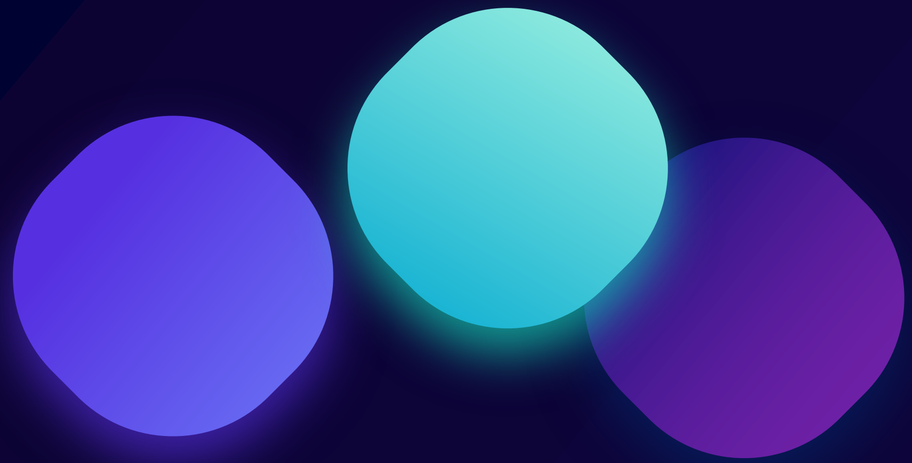
Topics covered

- 01 What is User Centered Design
- 02 Why UX Design In API Matters
- 03 Factors Behind API User Experience
- 04 Why Your API UX Is Bad

05 Design APIs with UX Design Approach

06 Qualities of Good API Design

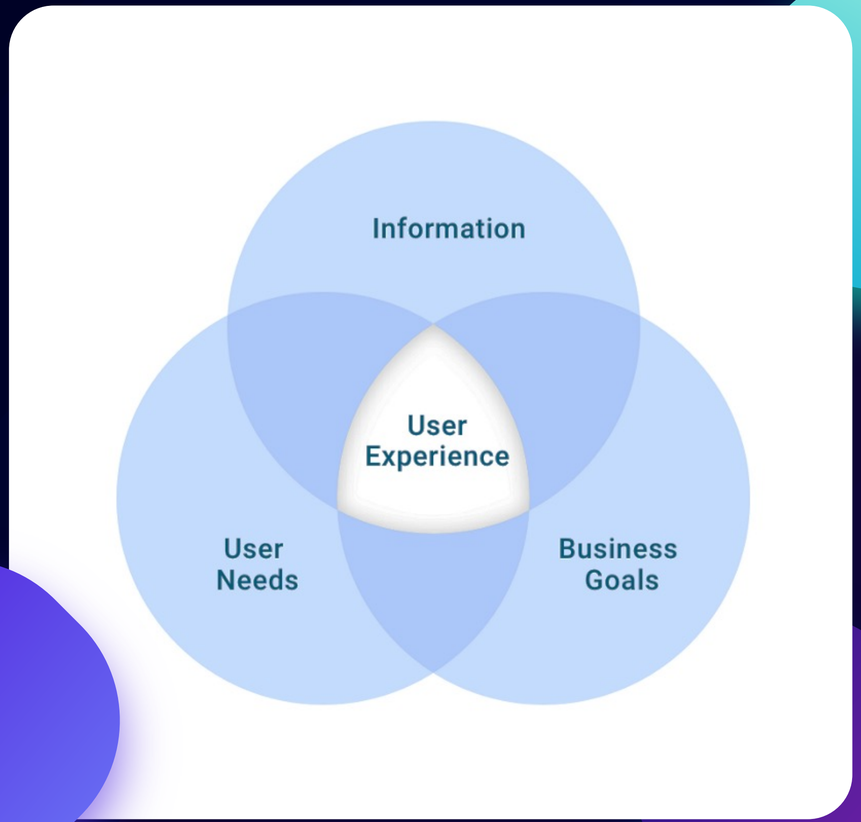
07 Conclusion



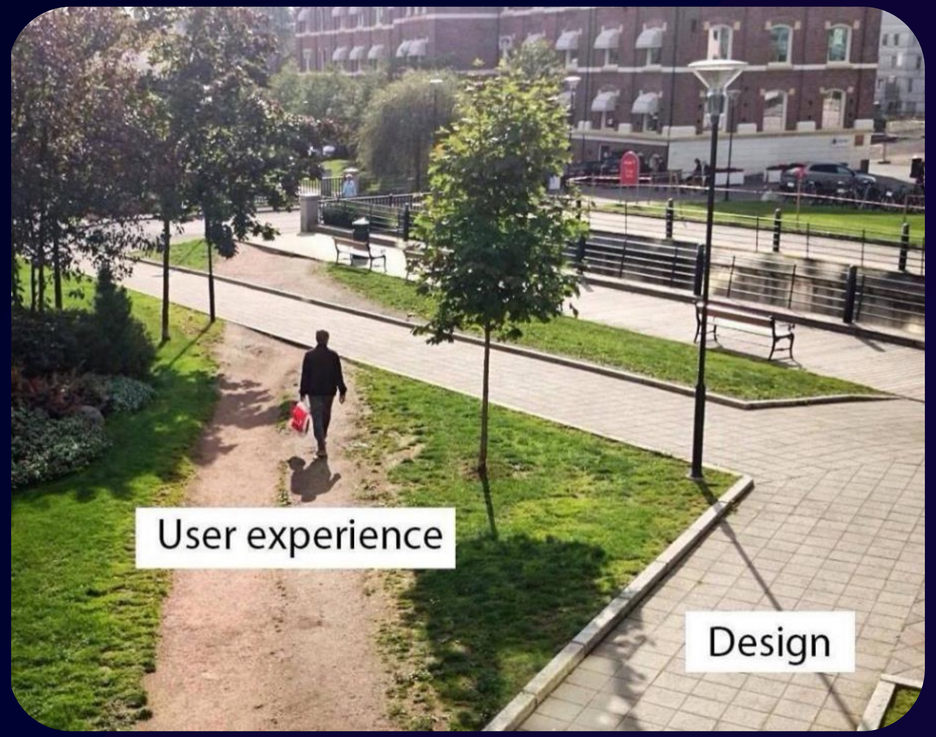
What is User Centered Design?

#01

User Centered Design



User Experience Design



What is UX Design in API Matters?

#02

Consequences of Poor API Design

- Increased long-term costs and losses
- Difficulties in documentation
- Unexpected downtimes
- A lack of consistency, which can hurt the developer experience
- Inconsistencies that lead to Bigger Errors
- Loss of trust

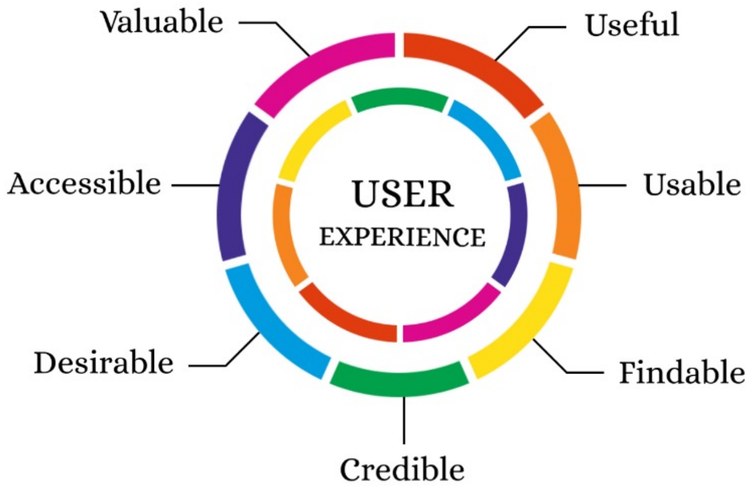
Importance of Good API Design

- Helps build better implementation
- Facilitates Continuous Development
- Improves Developer Experience
- Saves Time and Money in Implementation
- Improves Sustainability of the API

Factors Behind API User Experience

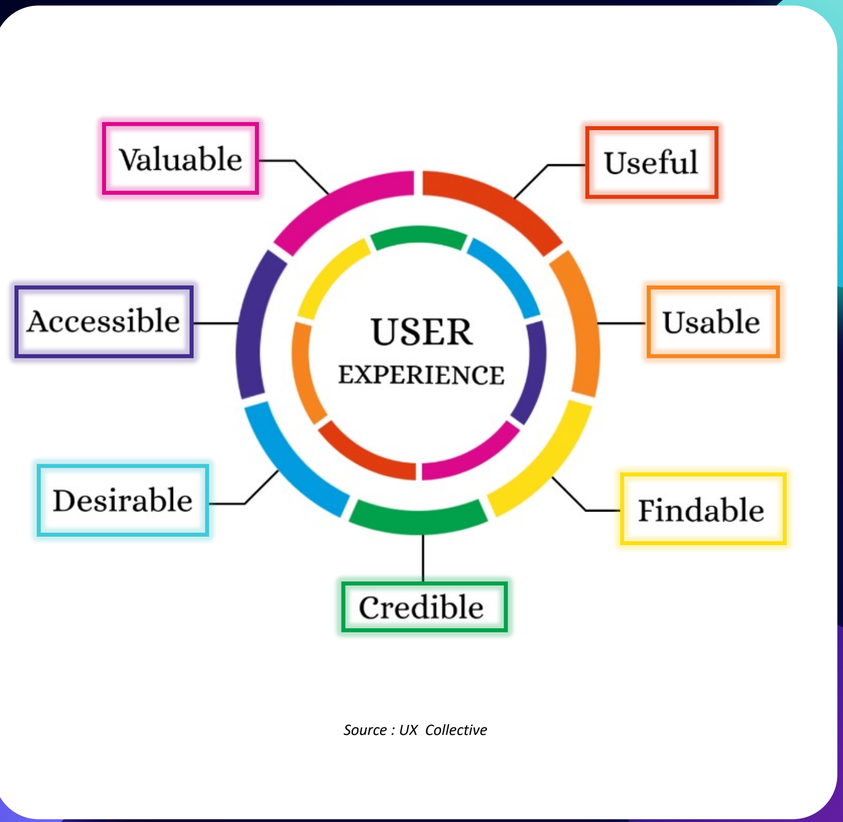
#03

Factors Influencing User Experience



Source : UX Collective

Factors influencing API User Experience



Source : UX Collective

Why your API UX Is Bad

#04

Your API documentation was written without considering the users

A bad example: Relying on auto-generated documentation

Error codes not well documented

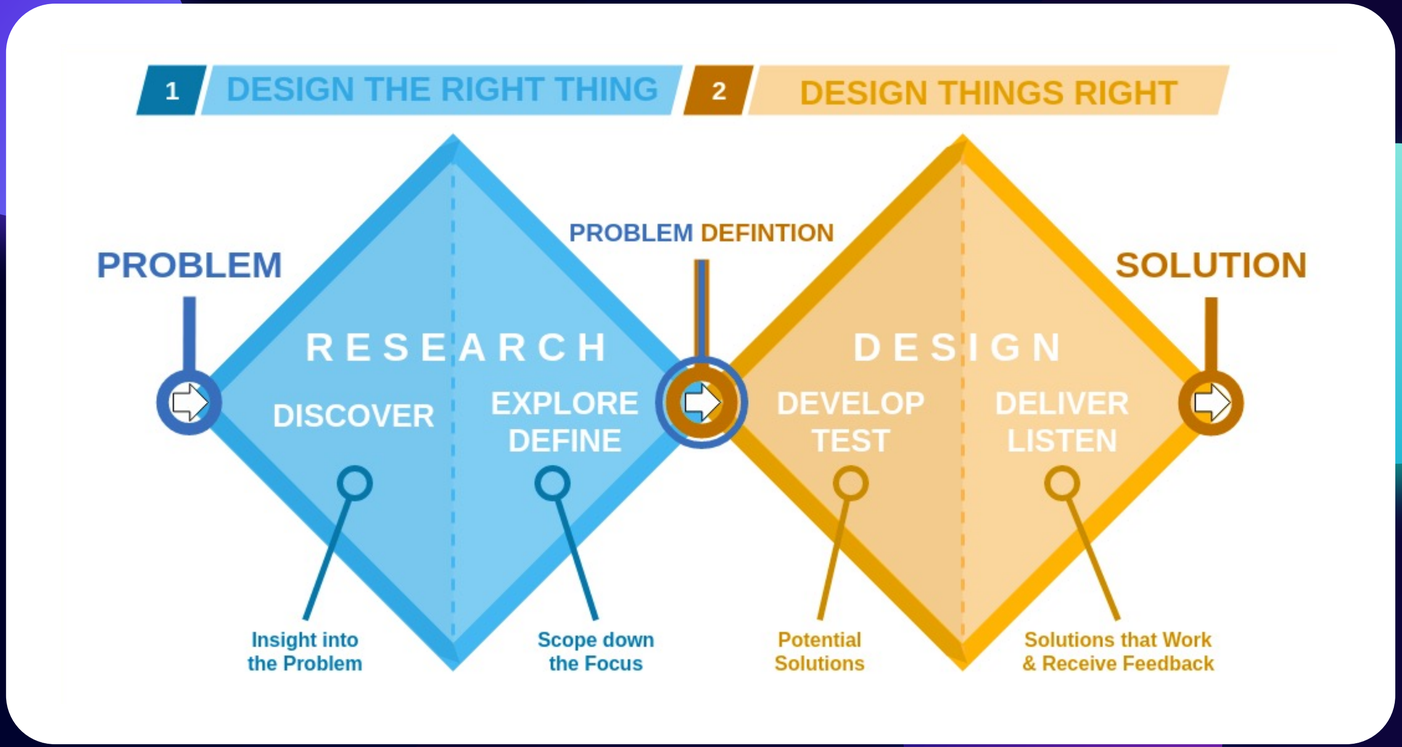
Your API authentication process is too complex for humans

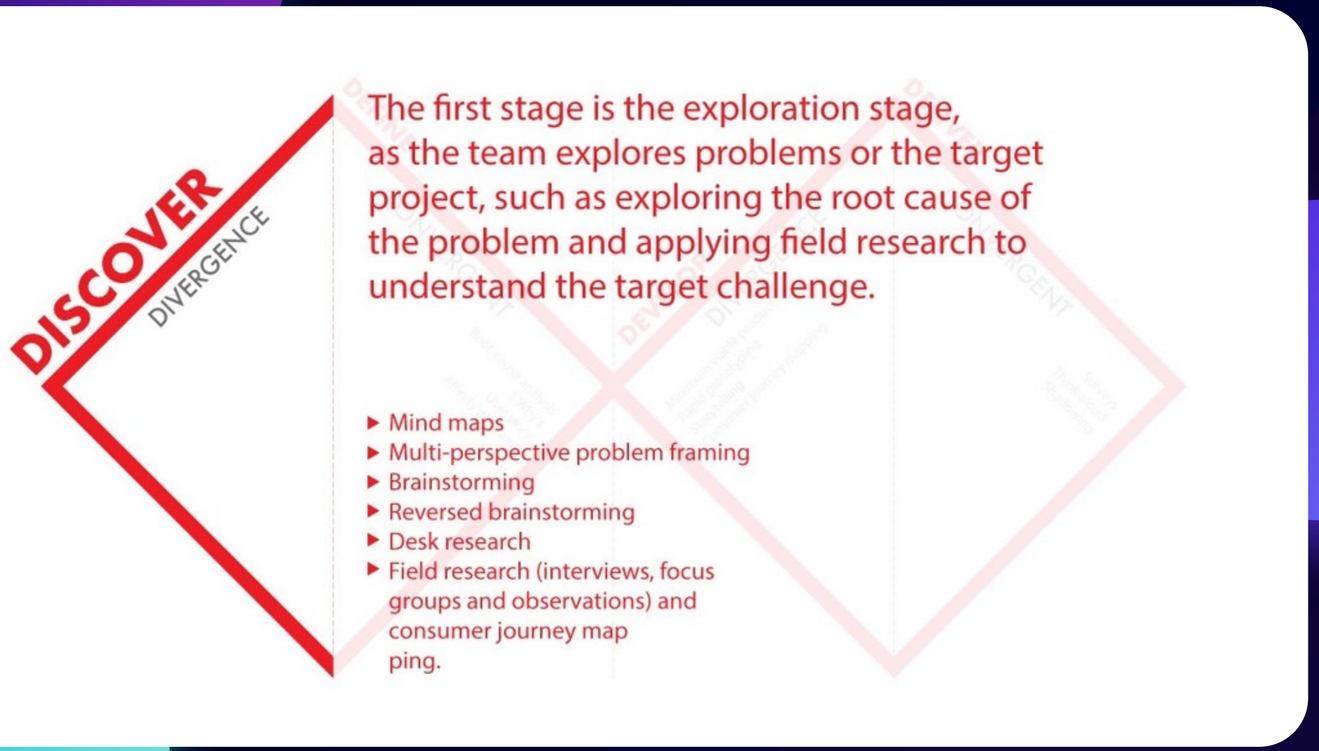
API consumers are more likely to leave if you API has a complex authentication process.

Design APIs with UX Design Approach

#05

‘Double Diamond’ Process



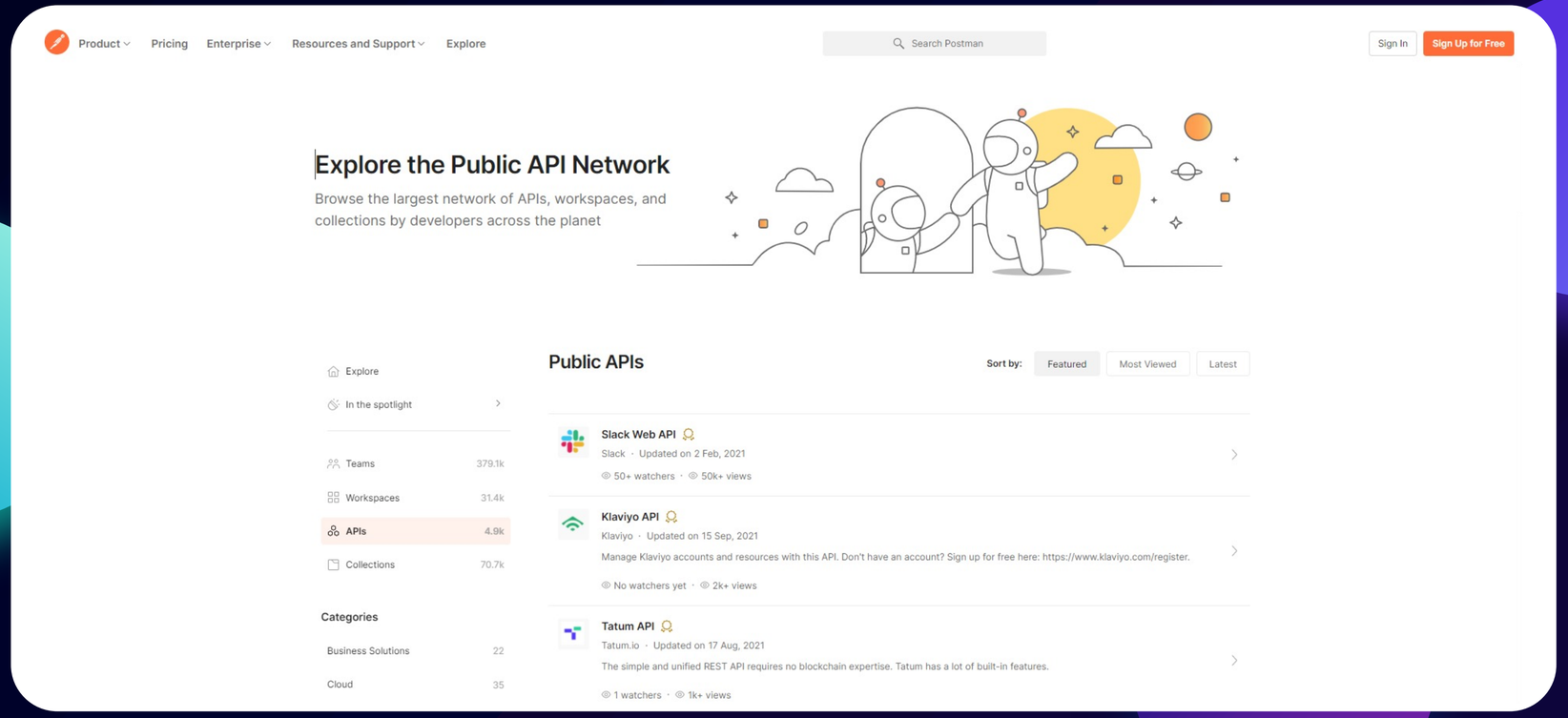


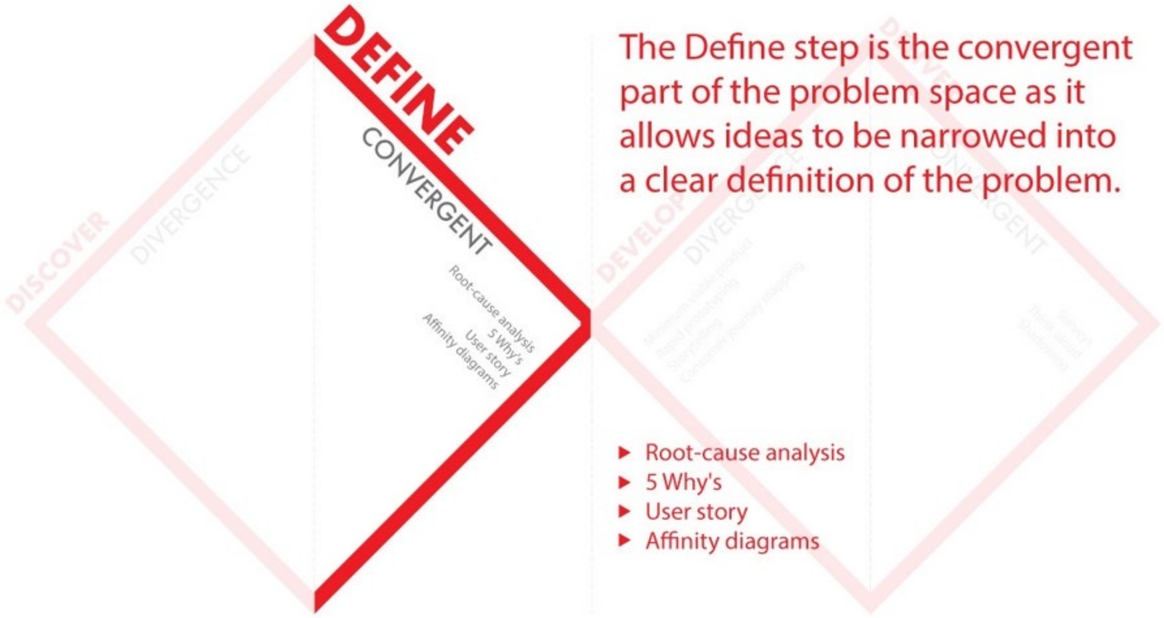
Problem Space

Designing The Right Thing

Leverage Other API Networks

Designing The Right Thing





Problem Space

Designing The Right Thing

Develop is the first step in the solution space. In this step, the team builds prototypes of the solution and tests it involving the end-user in the testing process

- Minimum viable product
- Rapid prototyping
- Storytelling
- Consumer journey mapping

Solution Space

Designing Things Right

Solution Space

Designing Things Right

Design Process Summary

DISCOVER

Explore the problem with your team

Define

Clearly describe the problem need to be addressed

Develop

Create the solution prototype, test and iterate to improve the prototype.

Deliver

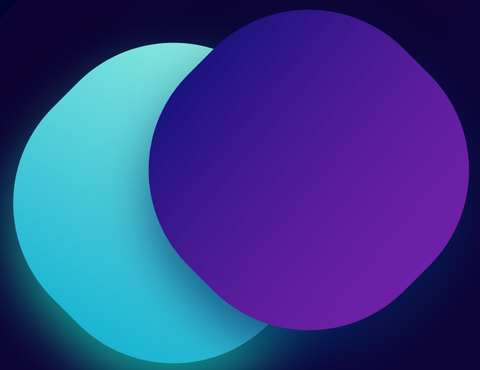
Deliver the final product to the consumer and get feedback that will be used to improve the product in the future.

Qualities of A Good API Design

#06

A Well-designed API is/has...

- Easy to learn
- Easy to use, even without documentation
- Appropriate to the audience
- Hard to misuse
- Easy to read and maintain
- Easy to scale
- Meaningful error message



Some Things to Consider When Designing Your APIs

#07

APIs should be designed from the view of the consumer

API should satisfy the requirements.

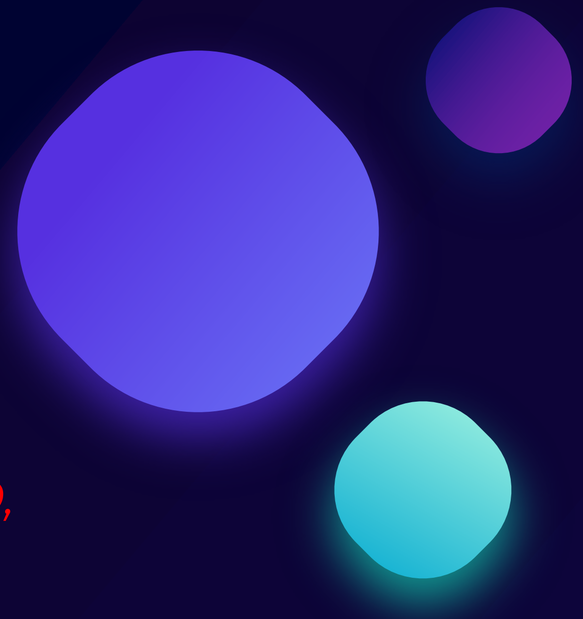
API consumer preferences matter – take them serious. Build simple, clean, clear and approachable APIs – from the perspective of your consumers.

API should do one thing and do it well

Easy to explain functionality”
The functionality should be easy to explain
The names should be clear

Bad; CustomerDynServicesAccName, OMGVMCICD, ENCODING_EDR_ENCAPS

Good: customer, privateKey, accountName, font, date, timeUnit



Use consistent naming and code patterns.

API naming conventions should be internally consistent (If you usually denote counts via the num_* prefix, don't switch to n_* in some places), but also consistent with widely recognized external standards.

API Should Provide Helpful Feedback to The User

Give context to our error messages
Make it meaningful.

Human readable messages that summarize the context, cause, and general solution for the error at hand.

```
< HTTP/1.1 400 Bad Request
< Date: Wed, 31 May 2017 19:01:41 GMT
< Server: Apache/2.4.25 (Ubuntu)
< Connection: close
< Transfer-Encoding: chunked
< Content-Type: application/json
{ "error" : "REQUEST - BR0x0071" }
```

```
< HTTP/1.1 400 Bad Request
< Date: Wed, 31 May 2017 19:01:41 GMT
< Server: Apache/2.4.25 (Ubuntu)
< Connection: close
< Transfer-Encoding: chunked
< Content-Type: application/json
{ "error" : "Bad Request - Your request is missing parameters. Please verify and res
ubmit. Issue Reference Number BR0x0071" }
```

```
HTTP/1.1 200
Date:
Thu, 01 Jun 2017 03:40:55 GMT
Content-Length:
276
Connection:
keep-alive
Content-Type:
application/json; charset=utf-8
Server:
Microsoft-IIS/10.0
X-Content-Type-Options:
nosniff

{"SearchResponse":{"Version":"2.2","Query":{"SearchTerms":"api error codes"},"Error
s":[{"Code":1001,"Message":"Required parameter is missing.","Parameter":"SearchReque
st.AppId","HelpUrl":"http\u003a\u002f\u002fmsdn.microsoft.com\u002fen-us\u002flibrar
y\u002fdd251042.aspx"}]}}
```

Other Good Error Examples

From a GET Request to Bing

Conclusion

#07

Like all design processes, the API design process also needs to be an iterative process.

Thank You!

Anthony Ejiogu

linkedin.com/in/anthonyejiogu