



API Asia Conference 2019

Connect, Collaborate & Transform

2nd October 2019,
Shangri-La Colombo, Sri Lanka

APIs for Humans



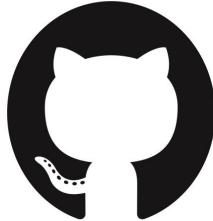
Uli Hitzel
API Evangelist
Axway



Developer Advocate



/in/uhitzel



/u1i/slides



IBM



Y!





cosmicrenaissance

1,257 posts

194 followers

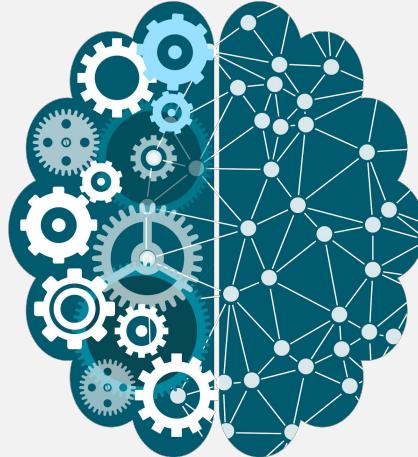
615 following

Just some guy, really.

+65

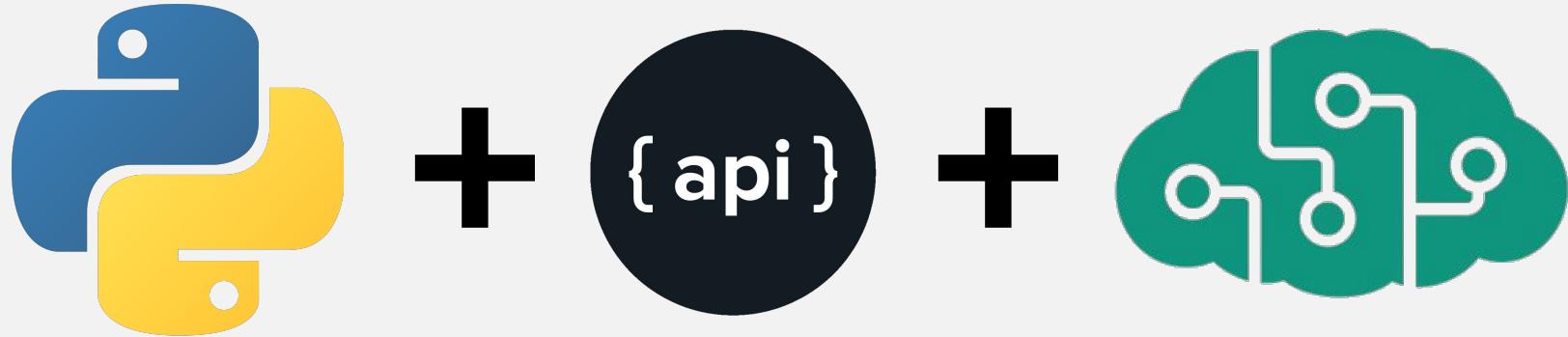
City life, scenery, food.

Camera: Samsung Galaxy S7/S9



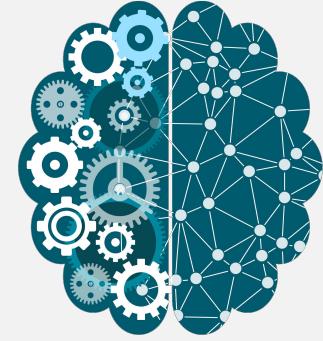
{ api }

A word cloud visualization showing various nouns and adjectives in different colors (blue, green, orange, red) scattered across the page. The words are organized into several large, bold letters: 'clock' (top left), 'outdoor' (top center), 'city' (top right), 'water' (center), 'train' (left), 'tower' (bottom left), 'plate' (bottom center), 'view' (bottom right), and 'field' (far right). Each letter contains a cluster of related words, such as 'sign' and 'green' under 'outdoor', or 'boat' and 'walking' under 'water'. Three green circles are overlaid on the word cloud, containing the words 'bench', 'view', and 'plate' respectively.



Microsoft

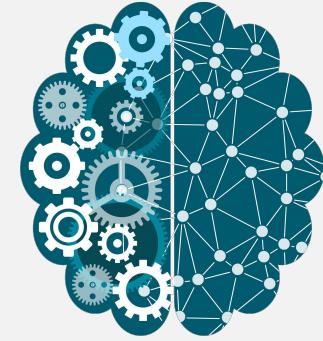
Cognitive Services



Text: "a close up of a pizza"

Confidence: 0.9246481371443355

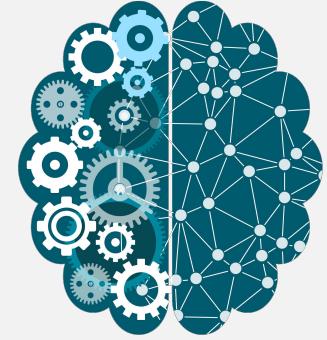
Tags: "pizza" "food" "table" "sitting" "cheese" "plate" "top" "piece" "slice" "toppings" "close" "eaten" "sauce" "covered" "wooden" "large" "pepperoni" "white" "pan"



Text: "a cup of coffee"

Confidence: 0.8527612488821656

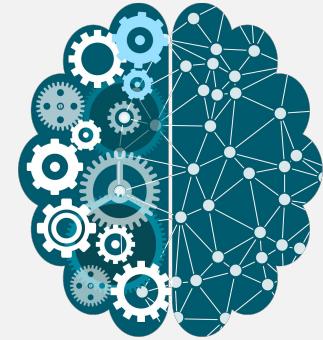
Tags: "cup" "table" "coffee" "food" "sitting" "drink" "plate" "sandwich" "breakfast"
"spoon" "glass" "close" "donut" "hot" "white" "phone" "soup"



Text: "a wooden table next to a window"

Confidence: 0.8365439206950018

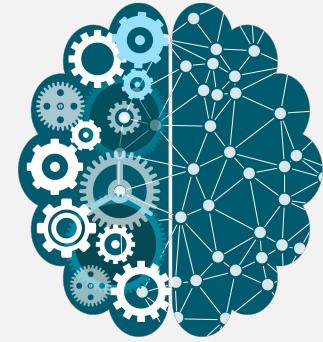
Tags: "table" "indoor" "window" "room" "sitting" "chair" "small" "desk" "wooden" "laptop" "lit" "top" "light" "counter" "computer" "living" "kitchen"



Text: "a sunset over the ocean"

Confidence: 0.8065587199349568

Tags: "outdoor" "water" "beach" "sunset" "ocean" "sun" "man" "pier" "track"
"board" "top" "large" "red" "standing" "surfing" "walking" "boat" "holding" "flying"



Text: "a bunch of bananas"

Confidence: 0.37608016747604295

Tags: "animal" "covered" "sitting" "old" "side" "large" "water" "lot" "many" "top"
"table" "bunch" "street" "fire" "bird" "snow" "parking" "group" "river" "ocean" "field"

Key Points

- 1 – Build APIs:
Design First, Code Later

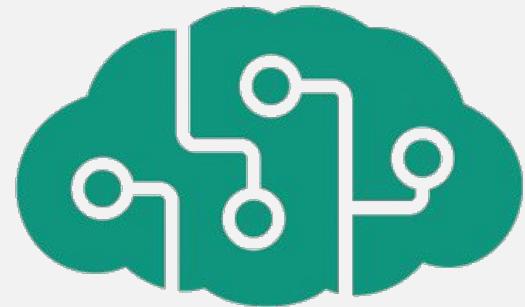
- 2 – Manage APIs:
Divide & Conquer

- 3 – API Adoption: Developer
Experience is Everything



You've just seen an example of this

- API abstracts something really complex
- Quick Results
- Easy to use → More Users → More Sample Data → AI learns → Better Product



 Microsoft
Cognitive Services



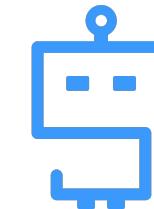
Design First

- Start with the Interface
- Use API Design Guides
- Quick Prototyping & Mocking

```
1  swagger: "2.0"
2  info:
3    description: "This is a sample server for
4      out more about Swagger at [http://swagger.io]
5      or on [irc.freenode.net, #swagger].
6      For this sample, you can use the api key
7      authorization filters."
8  version: "1.0.0"
9  title: "Swagger Petstore"
10 termsOfService: "http://swagger.io/terms/"
11 contact:
12   email: "apiteam@swagger.io"
13 license:
14   name: "Apache 2.0"
15   url: "http://www.apache.org/licenses/LICENSE-2.0.html"
16 host: "petstore.swagger.io"
17 basePath: "/v2"
18 tags:
19   - name: "pet"
20     description: "Everything about your pet"
21     externalDocs:
22       description: "Find out more"
23       url: "http://swagger.io"
24   - name: "store"
25     description: "Access to Petstore operations"
26   - name: "user"
27     description: "Operations about users"
28     externalDocs:
29       description: "Find out more about users"
30       url: "http://swagger.io"
```

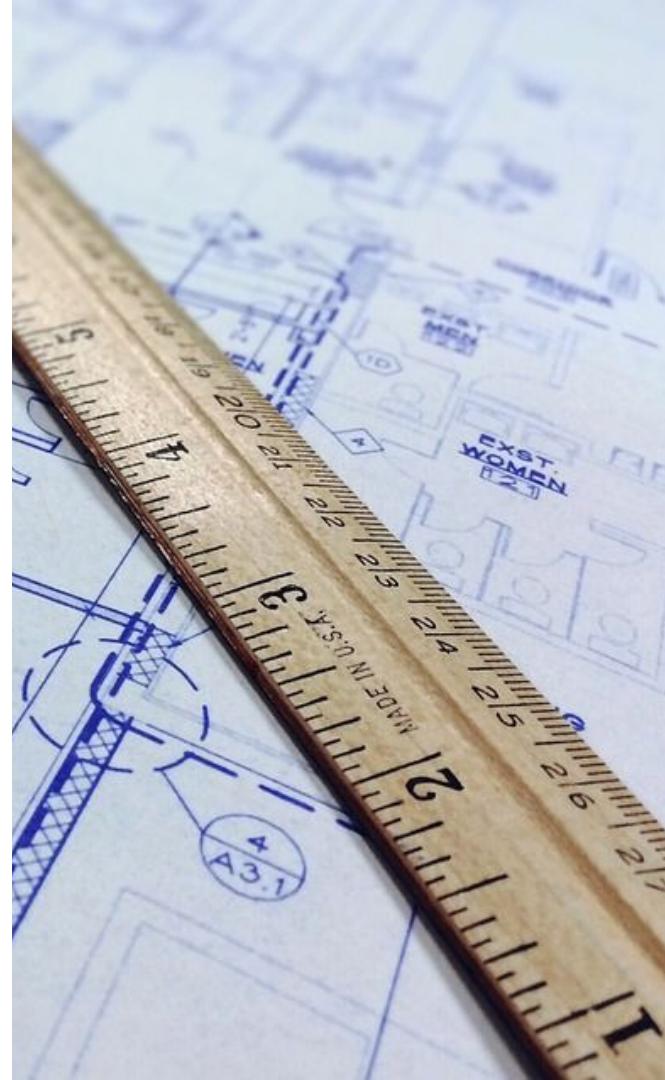
Contract First

- What can I do?
- What do I get?
- Data Models
- Authentication
- License



Be Strict about This

C REATE	→	HTTP POST /resource + Payload 201 + Location
R EAD	→	HTTP GET /resource 200 + Payload
U PDATE	→	HTTP PUT /resource/{id} + Payload 200/204
D ELETE	→	HTTP DELETE /resource/{id} 200/204



Use & Share Design Guidelines

The API Guidelines are split into two main parts:

- General Guidelines
- API type-specific Guidelines
 - REST APIs Guidelines
 - Kafka Guidelines

The general guidelines section discusses the core principles relevant to any kind of API. The API type-specific section further defines the guidelines specific to a given architectural style or API technique (such as REST, Kafka or GraphQL APIs).

How to read the Guidelines

These Guidelines are available for online reading at [GitBook](#) its source can be found on [GitHub](#).

The CAPITALIZED words throughout these guidelines have a special meaning:

- 1 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
- 2 "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in
- 3 this document are to be interpreted as described in RFC2119.



apistylebook.com

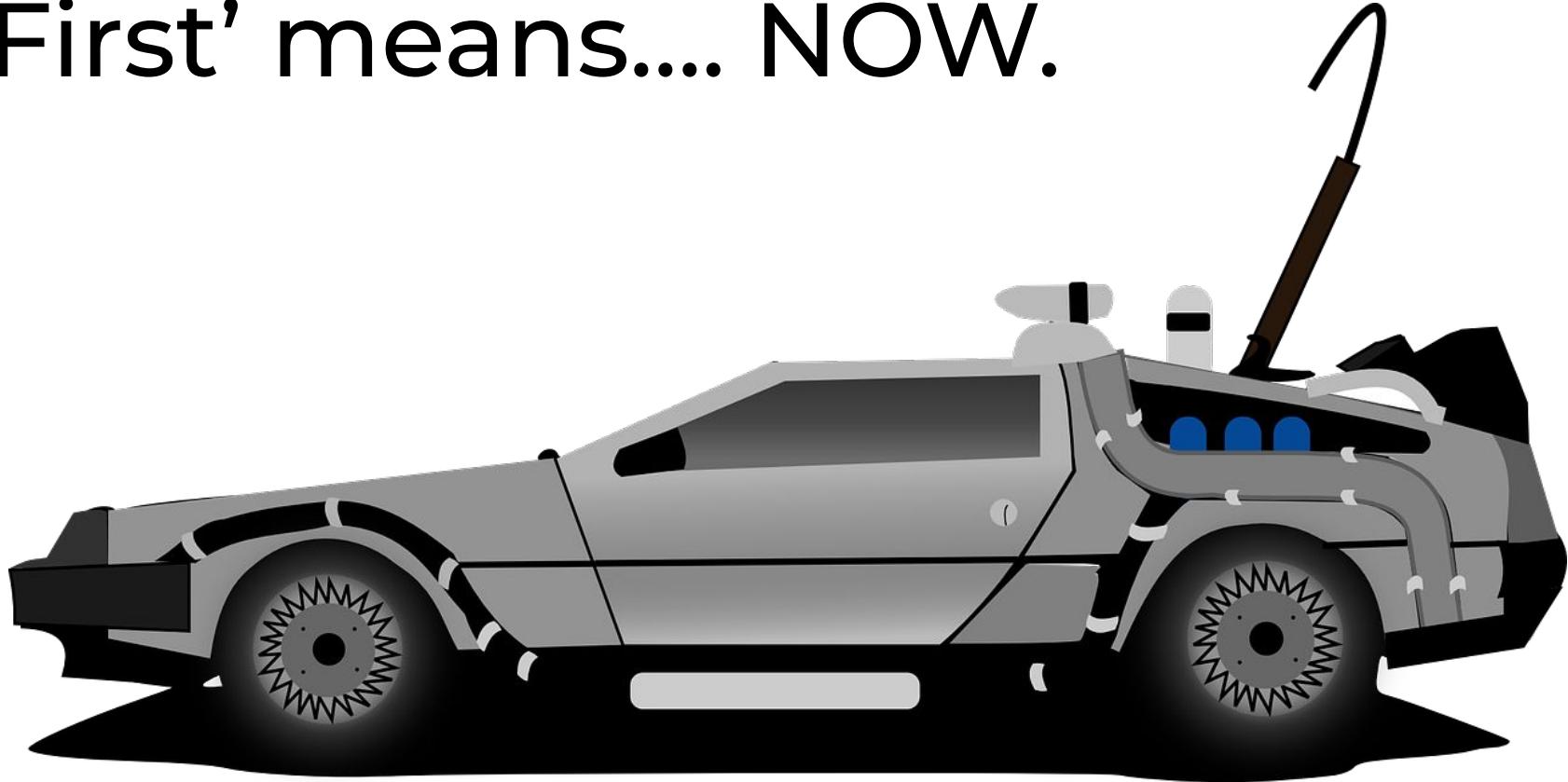


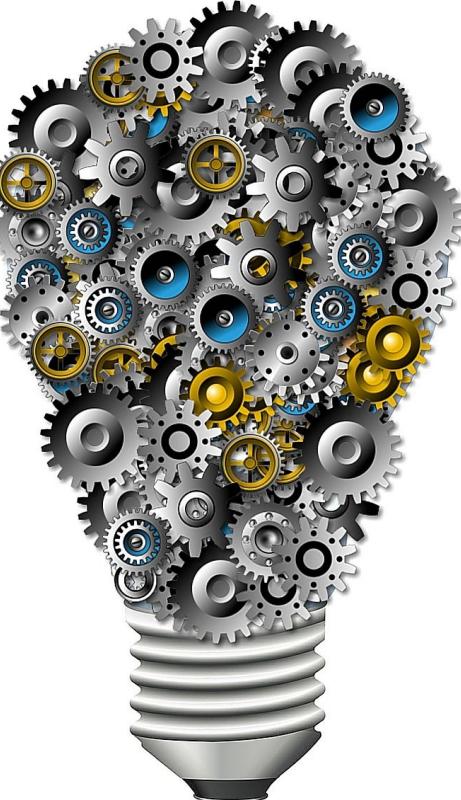
Iterations

Clean API Specs

- Generate Docu + SDK
- Mock with 'Semantics'
- Test
- Refine & Repeat

'First' means.... NOW.





Divide & Conquer

- Do you really need to code?
- Centralized Governance
and Unified Experience
- API Management as
Control Plane

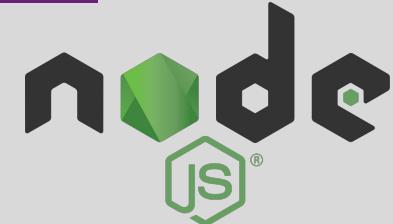
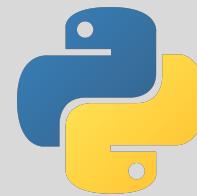
Use

- API Portal
- Swagger/OAS
- API Keys
- OAuth
- HTTP
- JSON
- Sandbox

Manage

- Access Control
- Security
- Quotas
- Transformation
- Service Catalog
- Self Service
- Analytics

Build





Developer Experience

- Who is your Customer?
- Give them Quick Results
- Involve them

3



Products made by Fans

Can you think like a developer, if you're not a developer yourself?

'Hello World' in 10 Minutes

- Instant Access
- Sandbox
- Examples, Tutorials & 'Flavored' Documentation



Involve & Empower your Audience

- Git powered Documentation
→ Pull Requests
- API Portal as your Developer Community
- Meetups, Success Engineers
- Developer Experience → Building Relationships



Takeaways

- 1 – Build APIs:
Design First, Code Later

- 2 – Manage APIs:
Divide & Conquer

- 3 – API Adoption: Developer
Experience is Everything



APIs for Humans

