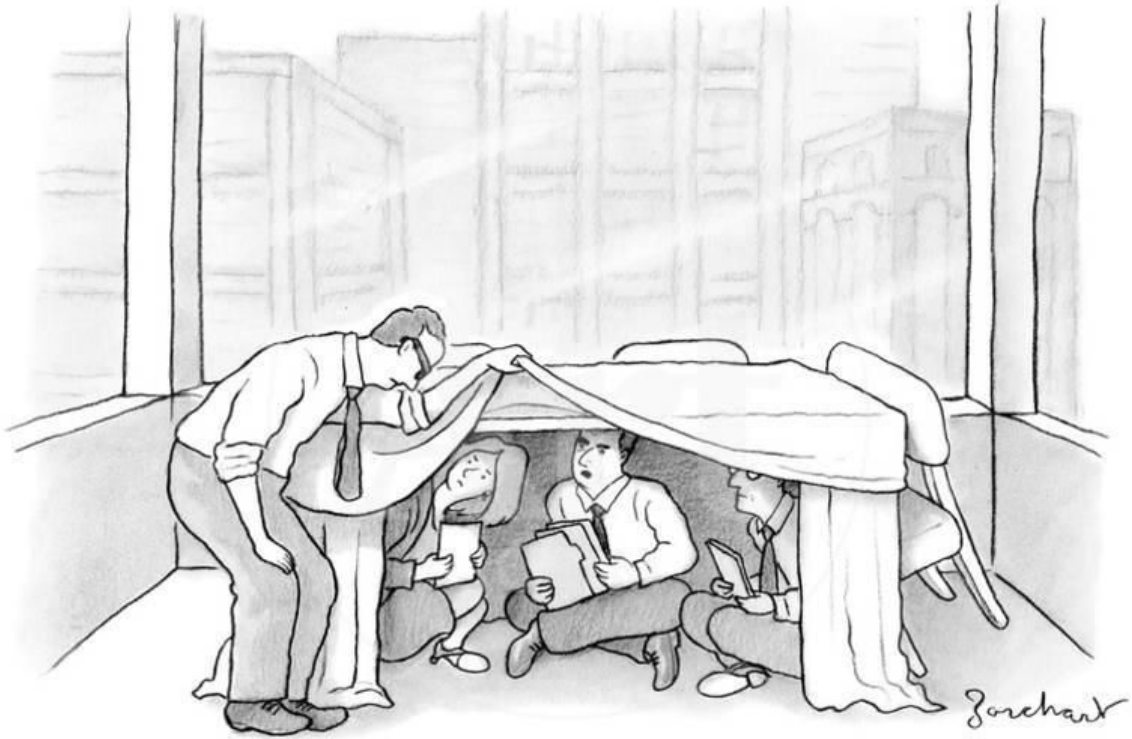
Auto-generating **REST API clients**

Enabled by OpenAPI (Swagger)

Alan Gross

https://www.linkedin.com/in/ajpgross/
https://github.com/ajpgross/open-api-client-gen-example



"This is highly confidential, so, yes, we built a little fort."

REST API definition

*“An API, or application programming interface, is **a set of rules that define how applications or devices can connect** to and communicate with each other”*

“A REST API is an API that conforms to the design principles of the REST, or representational state transfer architectural style”

*“REST APIs communicate **via HTTP requests to perform standard database functions like creating, reading, updating, and deleting records**”*

What is a REST API? | IBM

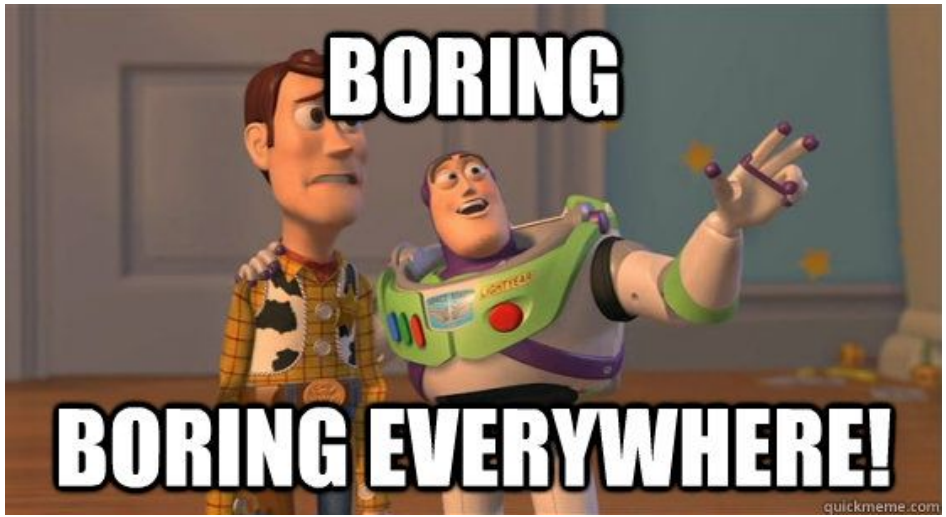
OpenAPI background

“An OpenAPI file allows you to describe your entire API, including:

- *Available **endpoints** (/users) and operations on each endpoint (GET /users, POST /users)*
- *Operation **parameters** Input and output for each operation*
- ***Authentication** methods*
- *Contact information, license, terms of use and other information.”*

About Swagger Specification | Documentation | Swagger

Coding a connection between a REST client and a REST API is tedious.



Problem space

Not DRY

- Hand-coding AJAX requests on the frontend
- Rebuilding interfaces, enums, and other models in TS

Not error-resistant

- Every line of code written is another opportunity for error
- The REST API's semantic versioning goes out the window

Not fun

- Not leveraging the full power of OpenAPI
- You can't automate it, which isn't fun …or can you?

Solution A

Ditch REST

GraphQL solves at least some of the listed problems

But switching isn’t always an option

Solution B

The monorepo

Super in-fashion right now, but it comes with scale challenges

You can avoid redundant models, but you *have to* be using compatible backend and frontend tech

Monorepos | Fireship | YouTube

Solution C

Keep REST and automate

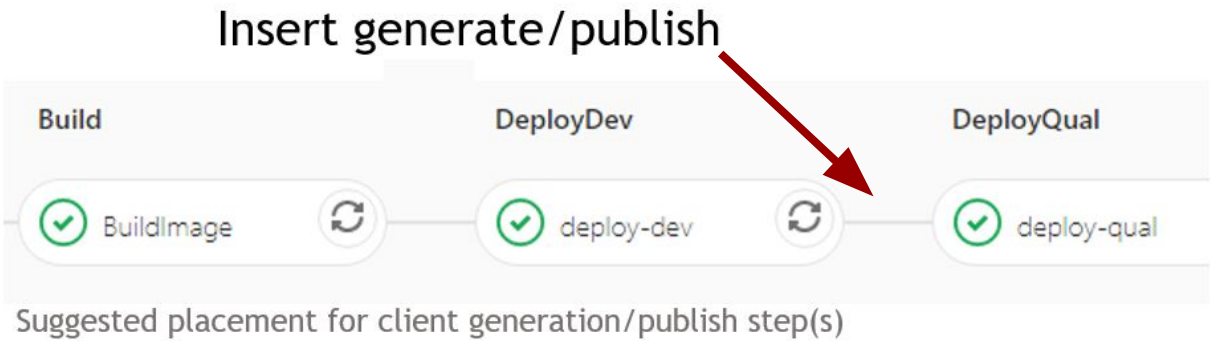
OpenAPI already models your whole REST API as JSON for you

A generation tool can read that JSON and write the code

OpenAPI Generator

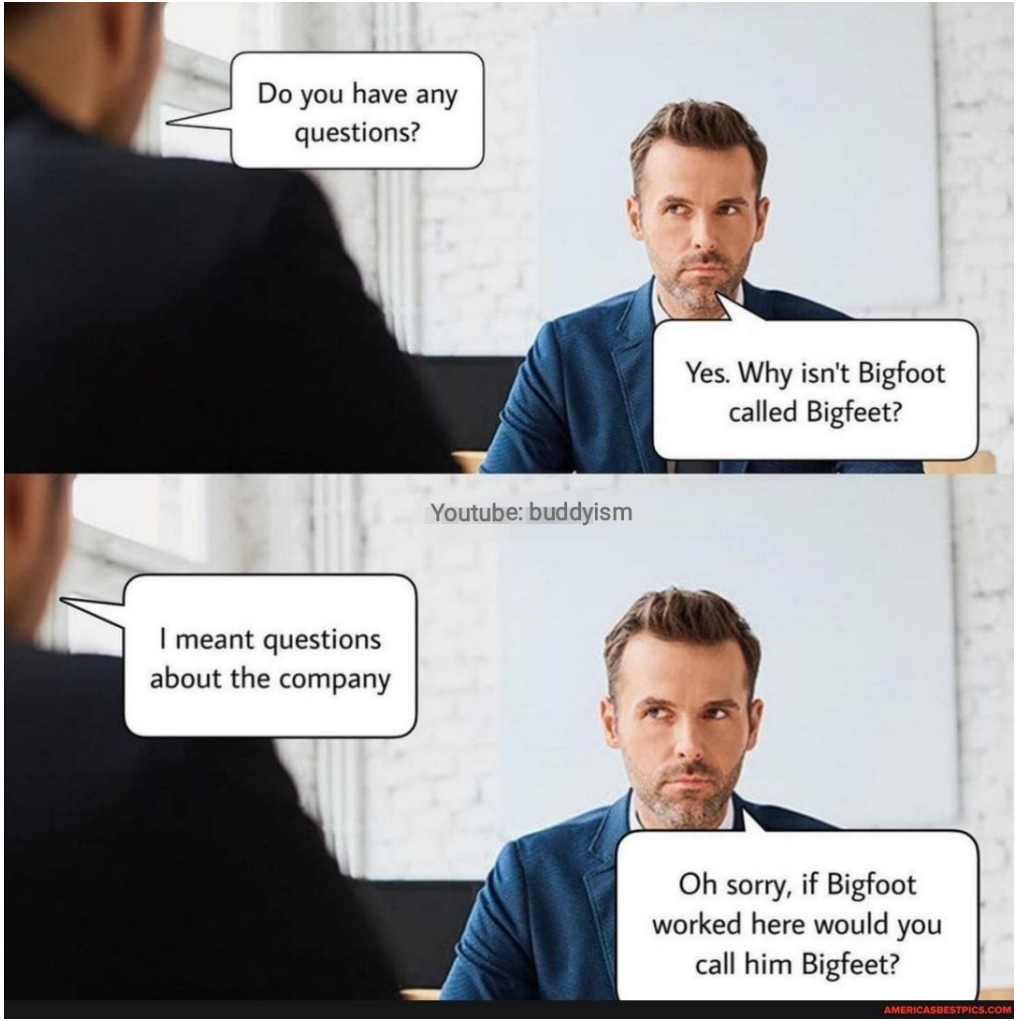
Solution C in depth

1. Feed an endpoint to the JSON file associated with your REST API's OpenAPI spec to the generator
   a. In the example, we use OpenAPI Generator in Docker
   b. If you need to build before publishing, remember to! (for instance, anything with TS)
2. Push to a package repo (on prem or otherwise) for consumption
3. Automate steps 1 and 2 as a job or pipeline (doing this manually gets old)



Other applications

- Connecting 2 *microservices* together (not limited just to font to back-end)
- Modify your e2e tests less
    - For small changes, drop in a new package and just hit play


Do you have any questions?
Yes. Why isn't Bigfoot called Bigfeet?
Youtube: buddyism
I meant questions about the company
Oh sorry, if Bigfoot worked here would you call him Bigfeet?
AMERICASBESTPICS.COM

Demo!