



Pragmatic DataAccess Modules

with OpenAPI

Speaker: **Sven Graziani** (Technical Frontend Lead @ Anexia SI)



Image courtesy: Colin Moldenhauer @unsplash.com



Contents

- * Overview
- * Status Quo
- * Pragmatic approach
- * Q&A

Terminology



Angular



DataAccess Module

Naming borrowed from the Domain Driven Design Universe

Terminology



Winkelabfangjäger

Google Translate ...

aka.

Http Interceptors

Terminology



(previously Swagger)

Terminology



Swagger
by REBORN SWAGGERS

<https://petstore3.swagger.io/api/v3/openapi.json> [Explore](#)

Swagger Petstore - OpenAPI 3.0 1.0.0 OAS3

<https://petstore3.swagger.io/api/v3/openapi.json>

This is a sample Pet Store Server based on the OpenAPI 3.0 specification. You can find out more about Swagger at <http://swagger.io>. In the third iteration of the pet store, we've switched to the design first approach! You can now help us improve the API whether it's by making changes to the definition itself or to the code. That way, with time, we can improve the API in general, and expose some of the new features in OAS3.

Some useful links:

- [The Pet Store repository](#)
- [The source API definition for the Pet Store](#)

[Terms of service](#)
[Contact the developer](#)
[Apache 2.0](#)
[Find out more about Swagger](#)

Servers

/api/v3

[Authorize](#)

pet Everything about your Pets

Find out more: <http://swagger.io>

PUT

/pet Update an existing pet

▼

POST

/pet Add a new pet to the store

▼

GET

/pet/findByStatus Finds Pets by status

▼

GET

/pet/findByTags Finds Pets by tags

▼

pet Everything about your Pets

PUT

/pet Update an existing pet

POST

/pet Add a new pet to the store

GET

/pet/findByStatus Finds Pets by status

GET

/pet/findByTags Finds Pets by tags

GET

/pet/{petId} Find pet by ID

POST

/pet/{petId} Updates a pet in the store with form data

DELETE

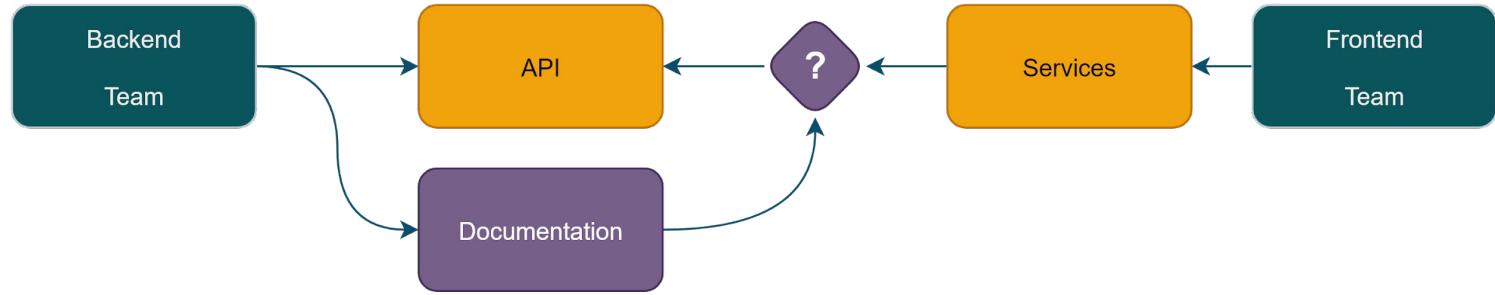
/pet/{petId} Deletes a pet

POST

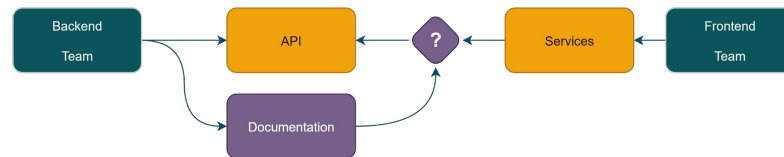
/pet/{petId}/uploadImage uploads an image

Status Quo

Status Quo



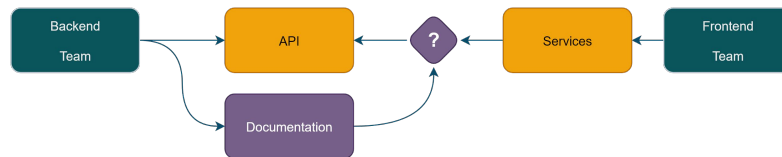
Status Quo



DEMO time

// todo: add error handling

Status Quo



CONS

- Time consuming
- Typos, **repetitive** work
- Interfaces tracking by hand
- lack of flexibility

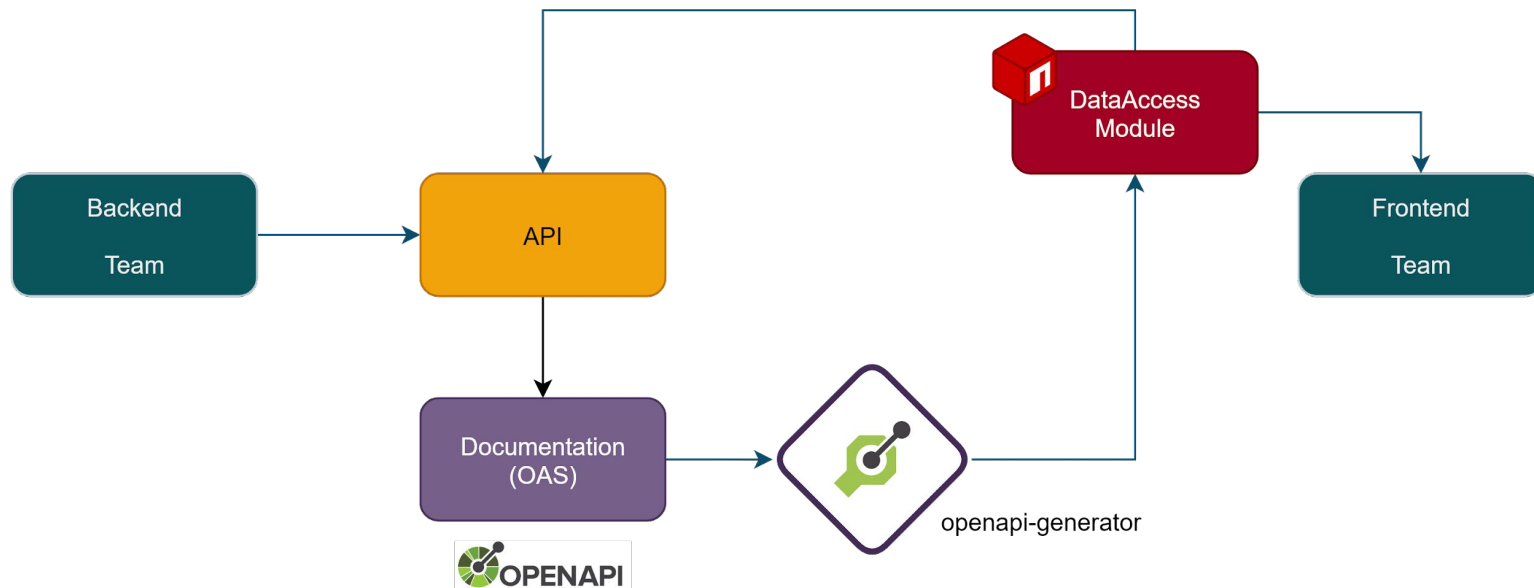
Pros

- feeling of control
- feeling of having something to do

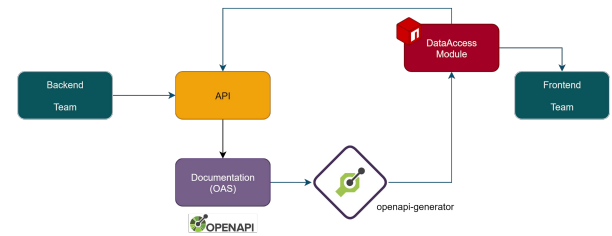
(eg. okey the next 2 Weeks i'm gonna type the 300+ Endpoints... --)

Pragmatic approach

Pragmatic approach

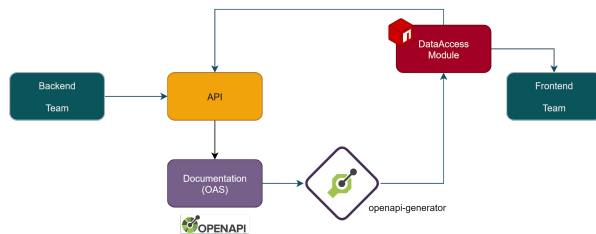


Pragmatic approach



DEMO time
// The Pet Store

Pragmatic approach



CONS

- we must modify generated code (Really?)
- the backend must provide an OAS (Really?)

Pros

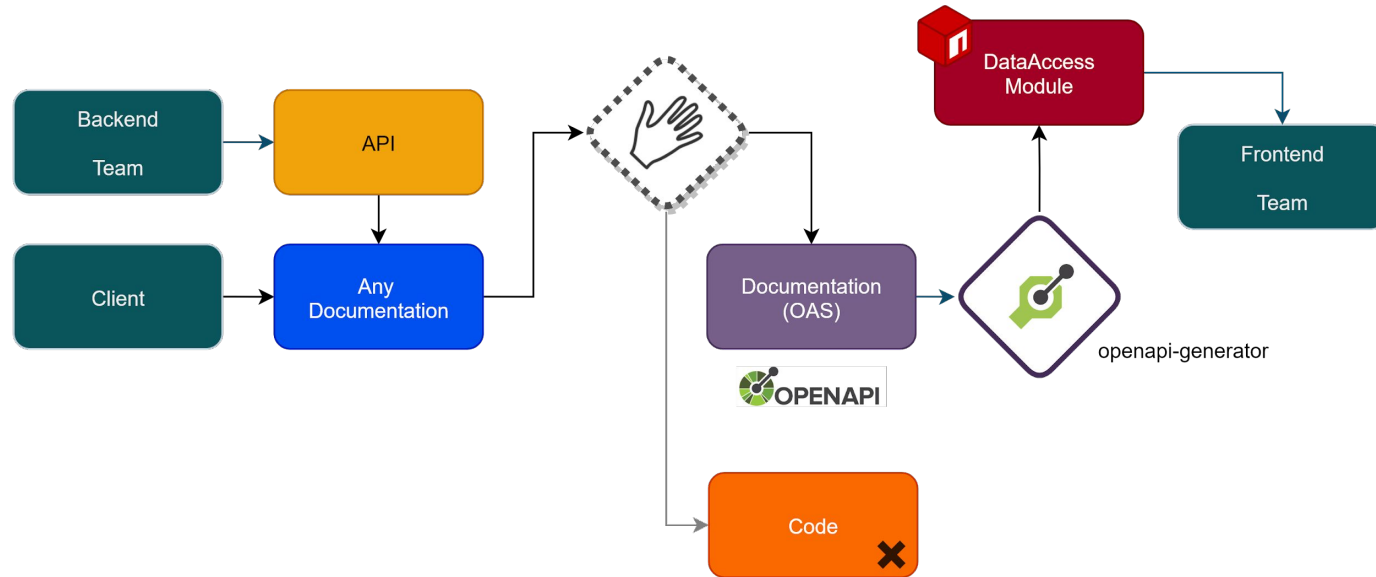
- time saving
 - descriptions
 - interfaces
- versioning on a module based level
- reduce typos
- flexibility
- generic
- easy migration
- portability
- ...

Con: we must modify generated code

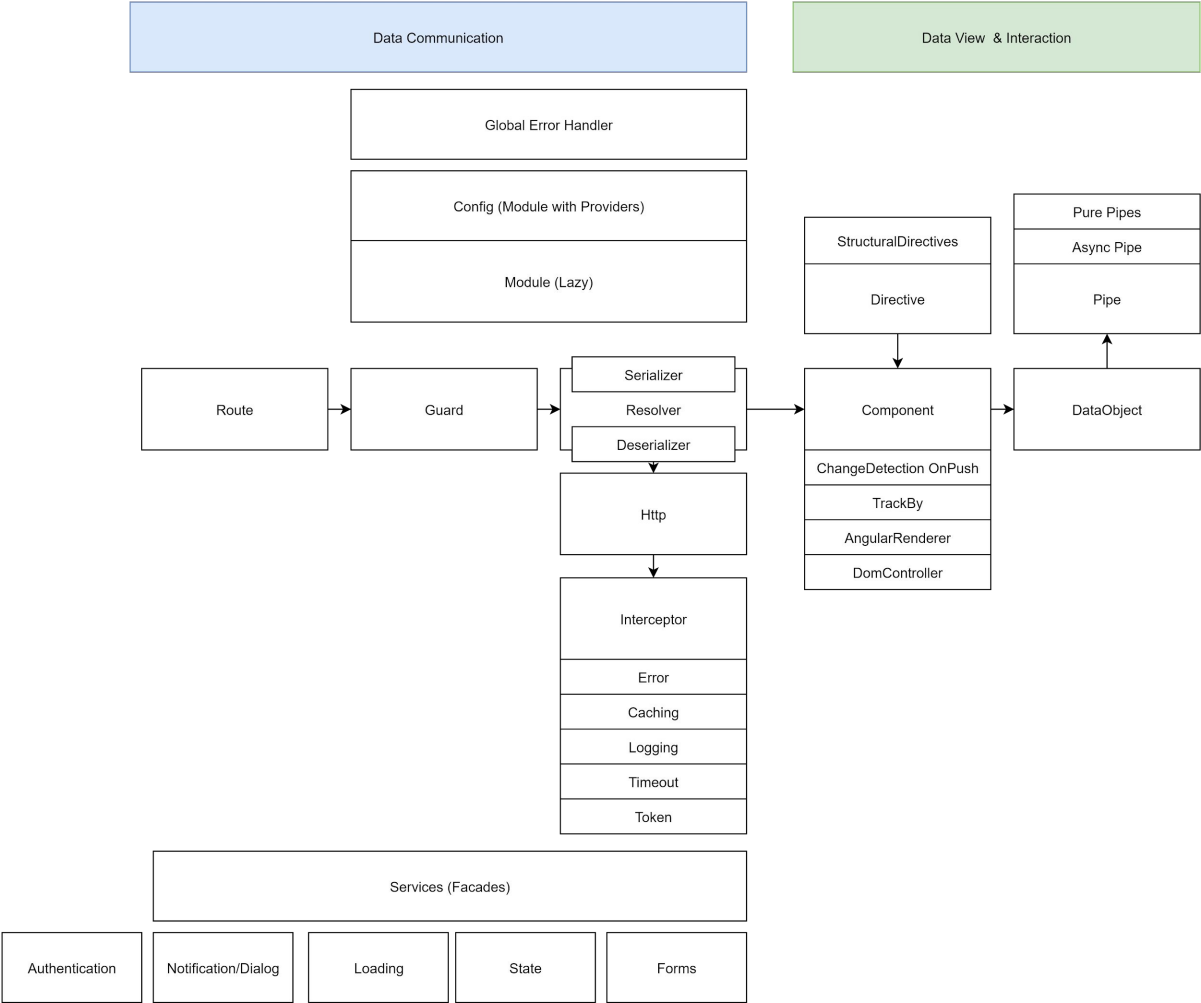
Http Interceptors

Http Context

Con: the backend/client must provide an OAS



In a Nutshell



Q&A

Thank you

Special thanks goes to André Prochiner.

