

A photograph of two children racing a shopping cart down a paved path. The child in the cart is wearing a blue helmet with white stars and goggles, and is leaning forward with an open mouth. The child pushing the cart is wearing an orange helmet and goggles, and is leaning back, pushing the cart forward. The background is a blurred field of dry grass under a bright sky. A semi-transparent white box is overlaid on the bottom half of the image, containing the text "API Boundries".

API Boundries

Sander Elias

Lead architect @Scully

Senior principal engineer @HeroDevs

GDE Web Technology

GDE Angular

Dutch Angular Group founder

Active in online communities

Available for a chat



What makes a good API?

What makes a good API?

1. Simplicity!

What makes a good API?

1. Simplicity!
2. Simplicity!

What makes a good API?

1. Simplicity!
2. Simplicity!
3. Simplicity!

Yeah, Yeah, but what is simplicity?

1. less is more.

Yeah, Yeah, but what is simplicity?

1. less is more.



Yeah, Yeah, but what is simplicity?

1. less is more.

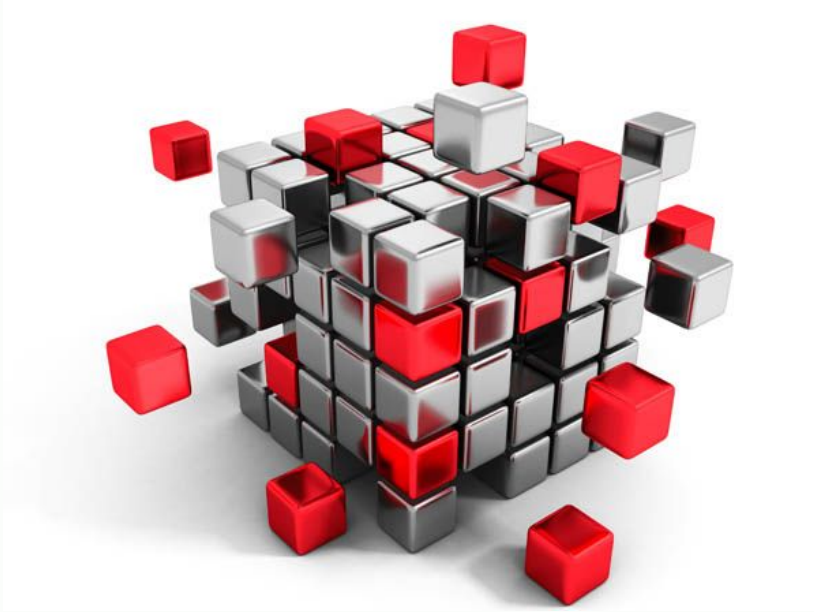


Yeah, Yeah, but what is simplicity?

2. Simple data types

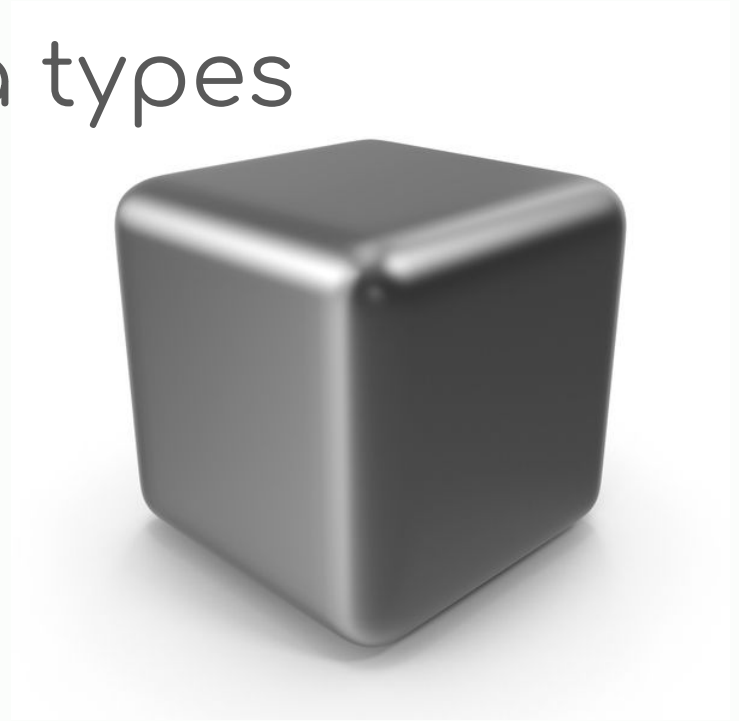
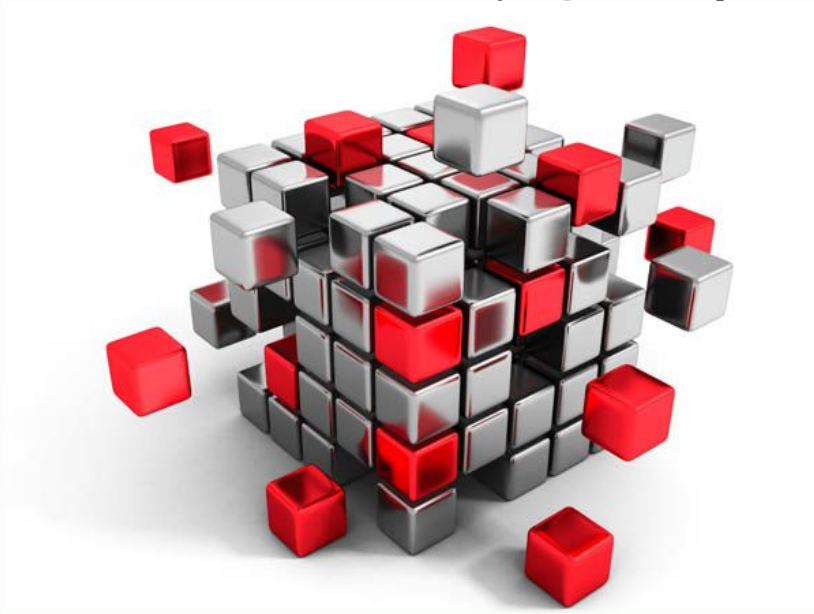
Yeah, Yeah, but what is simplicity?

2. Simple data types



Yeah, Yeah, but what is simplicity?

2. Simple data types



Yeah, Yeah, but what is simplicity?

3. Consistency

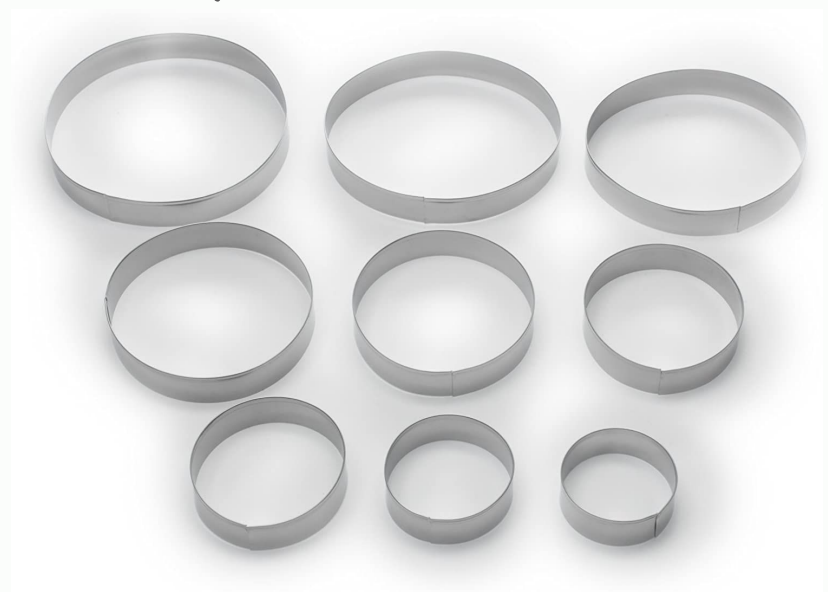
Yeah, Yeah, but what is simplicity?

3. Consistency



Yeah, Yeah, but what is simplicity?

3. Consistency



But I'm building Angular components

Not an backend???

Yes!!!
I know!

Components have an API too!

What makes up the API of a component?

What makes up the API of a component?

- The @Input()
 - How we name them
 - What data do they accept
- The @Output()

What makes up the API of a component?

- Updating the screen is an implied side-effect
- Sub-component interaction

(those are out of scope for this talk)

DEMO-TIME

DEMO-TIME

Well, almost....

First, some background

```
2  type RelationId = string;
3
4  export interface Relation {
5      id: RelationId;
6      /** more properties */
7  }
8  export interface Product {
9      id: string;
10     /** more properties */
11     creator: RelationId;
12 }
13 export interface Order {
14     id: string;
15     products: { productId: string, handler: RelationId }[];
16     transporter: RelationId;
17     processor: RelationId;
18 }
```

DEMO-TIME?

(dude that app needs some visual design!!)

(dude that app needs some visual design!!)

Yes, I'm aware, but not the point of the talk.. :-D

Show && tell.

Take-aways:

- Complex api leads to complex code
- Getting data up-front might not always be the best idea
- Imagine you will have 1.000.000 of each object.
- A bit of “smart” in lower level components pays off
- Simple component API's help you keeping logic in services
- They reduce the total amount of code

Thank You

A photograph of a rocket launch. The rocket is ascending vertically, leaving a large, billowing cloud of white smoke and orange fire at its base. It is flanked by four tall, slender service towers. The launch is taking place on a coastal area with green vegetation and a body of water in the foreground. The sky is a clear, deep blue.

API boundaries