



# THE LORD OF API DESIGNS

BY

ARNAUD LAURET

@APIHANDYMAN

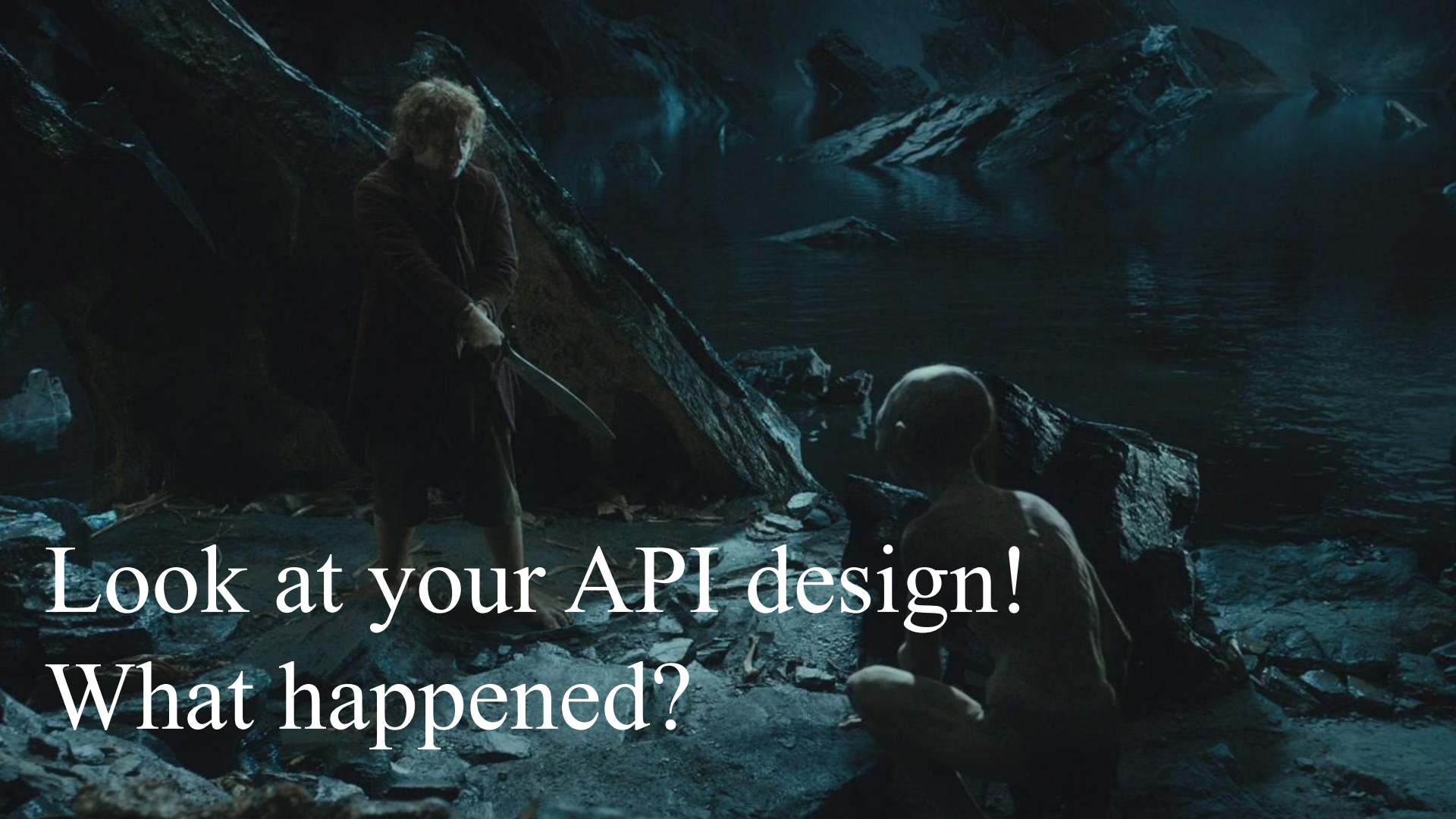
[APIHANDYMAN.IO](https://APIHANDYMAN.IO)

[APISTYLEBOOK.COM](https://APISTYLEBOOK.COM)



*Chapter I*

THE SHADOW OF  
API DESIGN



Look at your API design!  
What happened?



My precious  
API design!  
What have we  
done?





First API

But after that  
the nightmare  
started







First API  
updated

More and  
more public  
and private  
APIs!







Second API

Third API ...



Fourth API ...







... Nth API



What's the  
problem with  
these APIs?



/heroes/{bilbo}

/characters/heroes/{bilbo}

/api/hero/{bilbo}





“name”:

“FirstnameLastname”:

“character-name”:

“character\_name”:

“fnLn”:

“Baggins, Bilbo”

“Bilbo Baggins”

“bilbo baggins”

“BILBO BAGGINS”

“B. Baggins”

URL

Naming

Case

Data structures

Security

Parameters

Error messages

HTTP status codes

Behaviour

...



My precious!  
Too much  
differences!  
Too much  
inconsistency!





Inconsistency  
Between APIs  
And inside a single one





But what's the  
problem with  
inconsistency?





Why do you almost  
always know how to  
open a door?





Why Gandalf hates  
Moria's door?

One does not  
simply use  
inconsistent  
APIs



A person with curly hair, wearing a dark, textured jacket, is looking down. In the background, another person's hand is visible on their shoulder. The scene is set outdoors with a blurred background of trees and foliage.

Using each new endpoint or  
API is a totally new adventure



Users and  
their software  
hate inconsistent  
APIs



So much  
time lost!  
So much  
users lost!  
So much  
money lost!







Is that the  
work of a  
single API  
designer?



At the  
beginning  
yes...



A warrior with long blonde hair, wearing a silver helmet and red and brown armor, is shown in a dynamic pose. He is holding a sword aloft in his right hand and has a wide, shouting expression on his face. The background is a hazy, rocky landscape. The text "Let's forget what I've just doooooone!!!!" is overlaid on the left side of the image in a white, serif font.

Let's  
forget what I've  
just doooooone!!!!

... But we  
needed more  
and more API  
designers and  
things get  
even worse!





A cinematic scene featuring three orcs from 'The Hobbit' franchise. They are gathered around a campfire at night, cooking in a large metal pot. The orc in the center is actively stirring the pot with a long-handled spoon. The other two orcs are looking on, one with a focused expression and the other with a more neutral, perhaps skeptical, look. The scene is lit by the warm glow of the fire, creating a sense of intimacy and shared activity. The background is dark and out of focus, suggesting a forest setting.

Endless discussions



Repeated endless discussions





A cinematic scene featuring three knights on horseback in a field of tall, dry grass. The knight in the center is wearing ornate plate armor and a helmet, riding a white horse and holding a sword aloft. To his left, a knight with a beard and a simple helmet rides a grey horse. To his right, a third knight with a helmet and a cross on his forehead rides a dark horse. All three knights are smiling and appear to be celebrating. The background is a soft-focus field of tall grass under a warm, golden light.

Let's  
reinvent  
the wheeeeeel!

My precious!  
More  
inconsistency!  
And so much  
time loosed!





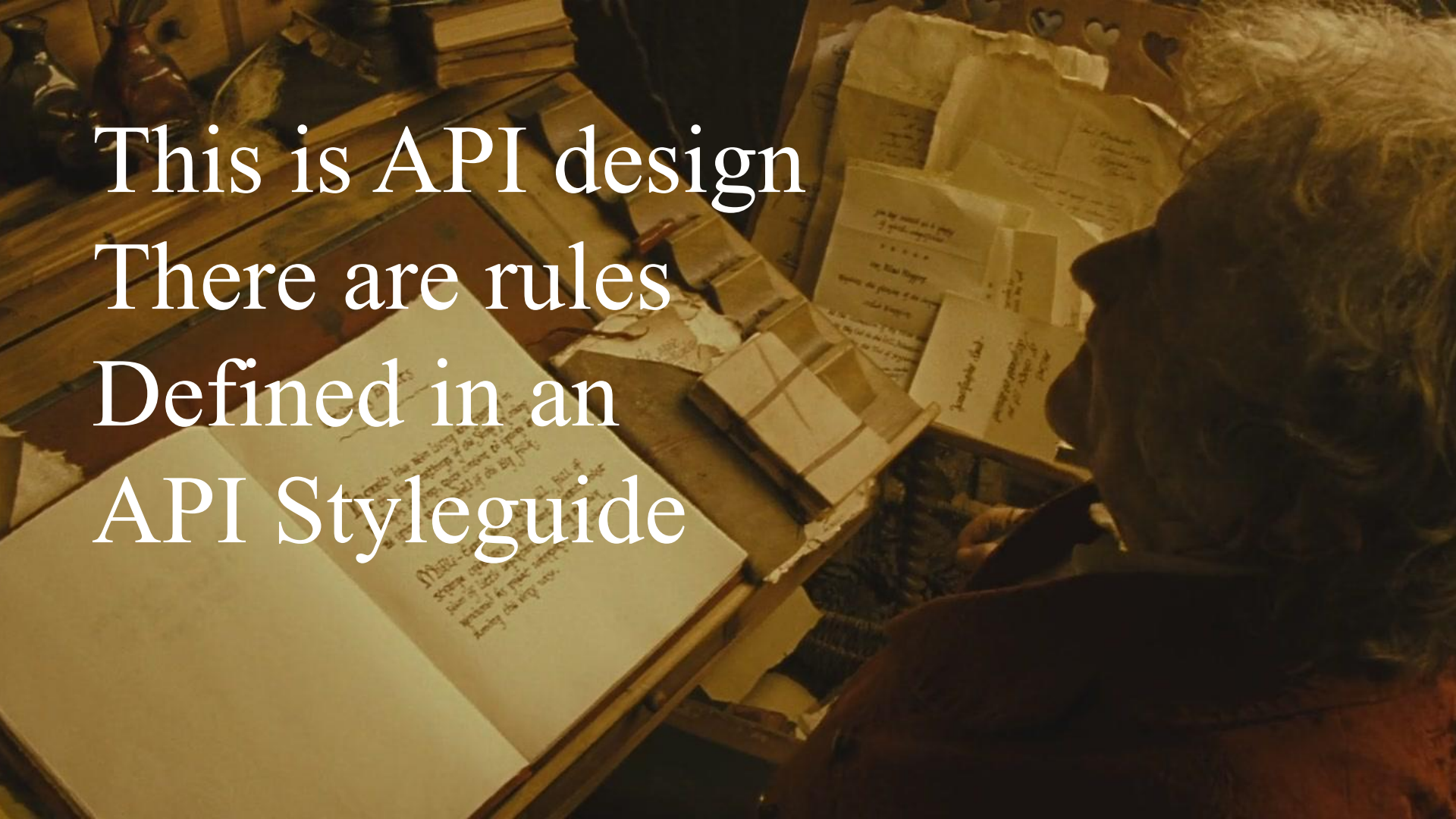
But what can  
we do my  
precious?



## *Chapter II*

# A LONG-EXPECTED API STYLEGUIDE



A person with curly hair is seen from the side, looking down at a desk. The desk is cluttered with various papers, some of which are handwritten and appear to be old or historical documents. There are also some small, folded pieces of paper or cards. A book is open on the desk, showing text on its pages. The overall scene suggests a workspace for research, writing, or editing.

This is API design  
There are rules  
Defined in an  
API Styleguide

A person with curly hair is seen from the side, looking down at a desk. The desk is cluttered with various papers, some of which are handwritten and appear to be old or weathered. A book is open on the desk, showing text that is partially legible. The scene is dimly lit, with a warm, yellowish light. The overall atmosphere is one of quiet study or research.

# How APIs look and behave



A person with curly hair is looking down at a desk. On the desk is an open book with handwritten text in a cursive script. There are also several loose papers and a small box on the desk. The scene is dimly lit, with a warm, yellowish light. The text "How to design APIs" is overlaid in white on the left side of the image.

# How to design APIs

A person with curly hair is seen from the side, looking down at a desk. The desk is cluttered with various papers, some of which are pinned to a corkboard in the background. A book is open on the desk, showing handwritten text. The scene is dimly lit, with a warm, yellowish light. The text "How to create but also update an API" is overlaid on the image in a white, serif font.

How to create  
but also update an API



# How to handle the API lifecycle

A person with curly hair is seen from the side, looking down at a desk. The desk is cluttered with various papers, some of which are handwritten and appear to be drafts or notes. A book is open on the desk, showing text that is partially legible. The scene is dimly lit, with a warm, yellowish light source, possibly a lamp, creating a focused and somewhat somber atmosphere. The person's hand is visible, holding a pen or pencil, suggesting they are actively working or reviewing the documents.



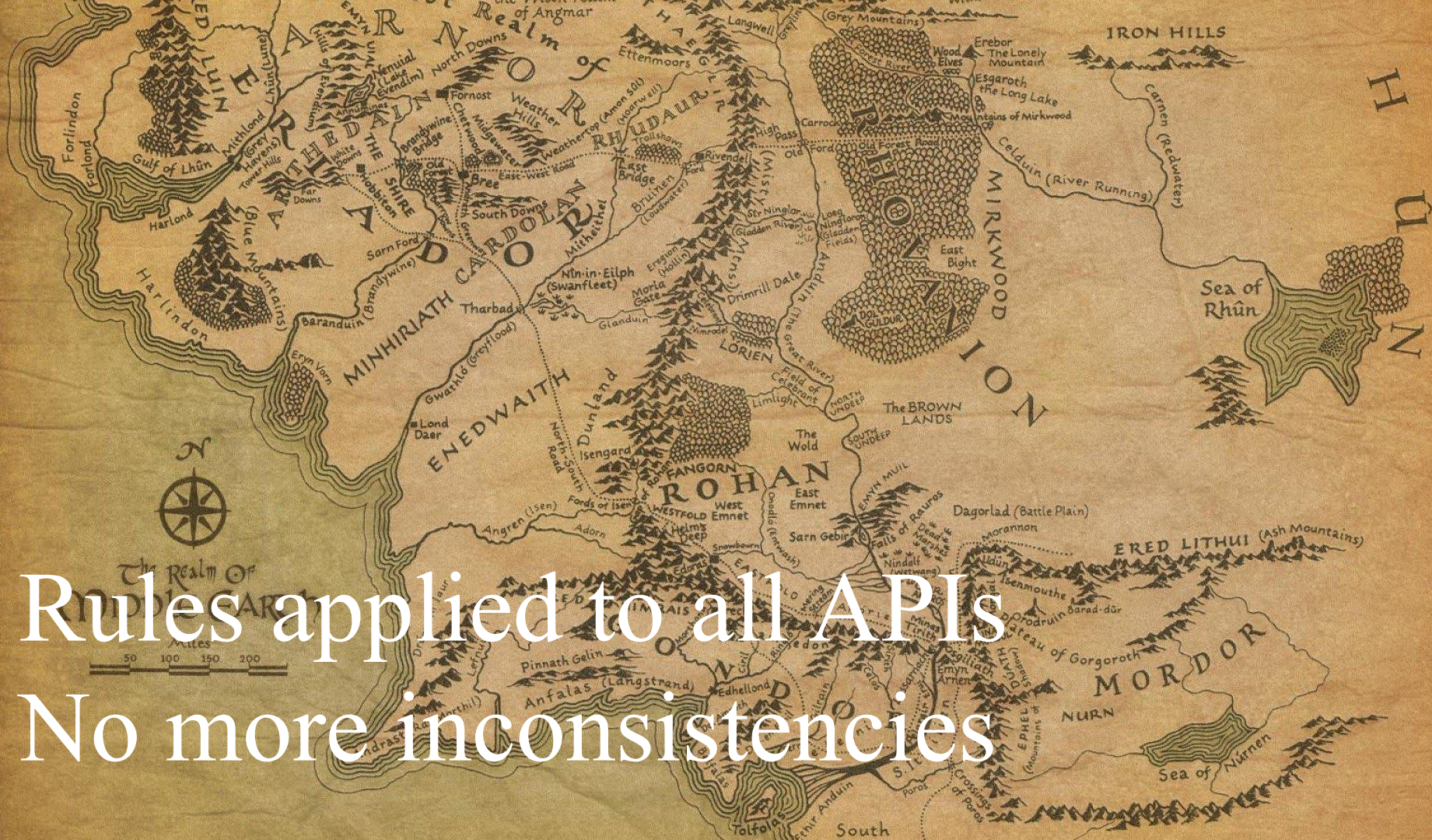
A person with curly hair is seen from the side, looking down at a desk. The desk is cluttered with various papers, some of which are handwritten and appear to be drafts or notes. A book is open on the desk, showing a page with the heading "Geology history" and some text. The scene is dimly lit, with a warm, yellowish light. The overall atmosphere is one of quiet study or research.

All needed rules  
to ensure consistency



Rules discussed once  
No more time loosed  
Focus on what's really important





Rules applied to all APIs  
No more inconsistencies





Both users and designers  
are happy





So all we  
have to do is  
describe rules  
in a styleguide?

## *Chapter III*

# THE BATTLE OF THE API STYLEGUIDE



# Write

Georgy Zhukov

The coach has been busy all morning in  
the first part of the day. The day  
has been a very busy one for me and  
at night I have been very busy.

My dear George, I have been very busy  
all day. I have been very busy all day.  
I have been very busy all day. I have  
been very busy all day. I have been  
very busy all day. I have been very  
busy all day. I have been very busy  
all day. I have been very busy all day.



A man with long white hair is seated at a desk, intently reading a large, aged document. The scene is dimly lit, with three lit candles in a holder on the left providing the primary light source. A large, round magnifying glass sits on the desk in the foreground. The background is filled with stacks of books and papers, suggesting a library or study. The overall atmosphere is one of quiet concentration and historical scholarship.

Rules based on  
your existing APIs  
or others



A man with long white hair is seated at a desk, intently reading a large, aged document. The scene is dimly lit, with three lit candles providing the primary light source. The desk is cluttered with various items, including a glass and some papers. The background shows a bookshelf filled with books.

Based on existing  
API styleguides  
[apistylebook.com](http://apistylebook.com)  
(shameless plug)

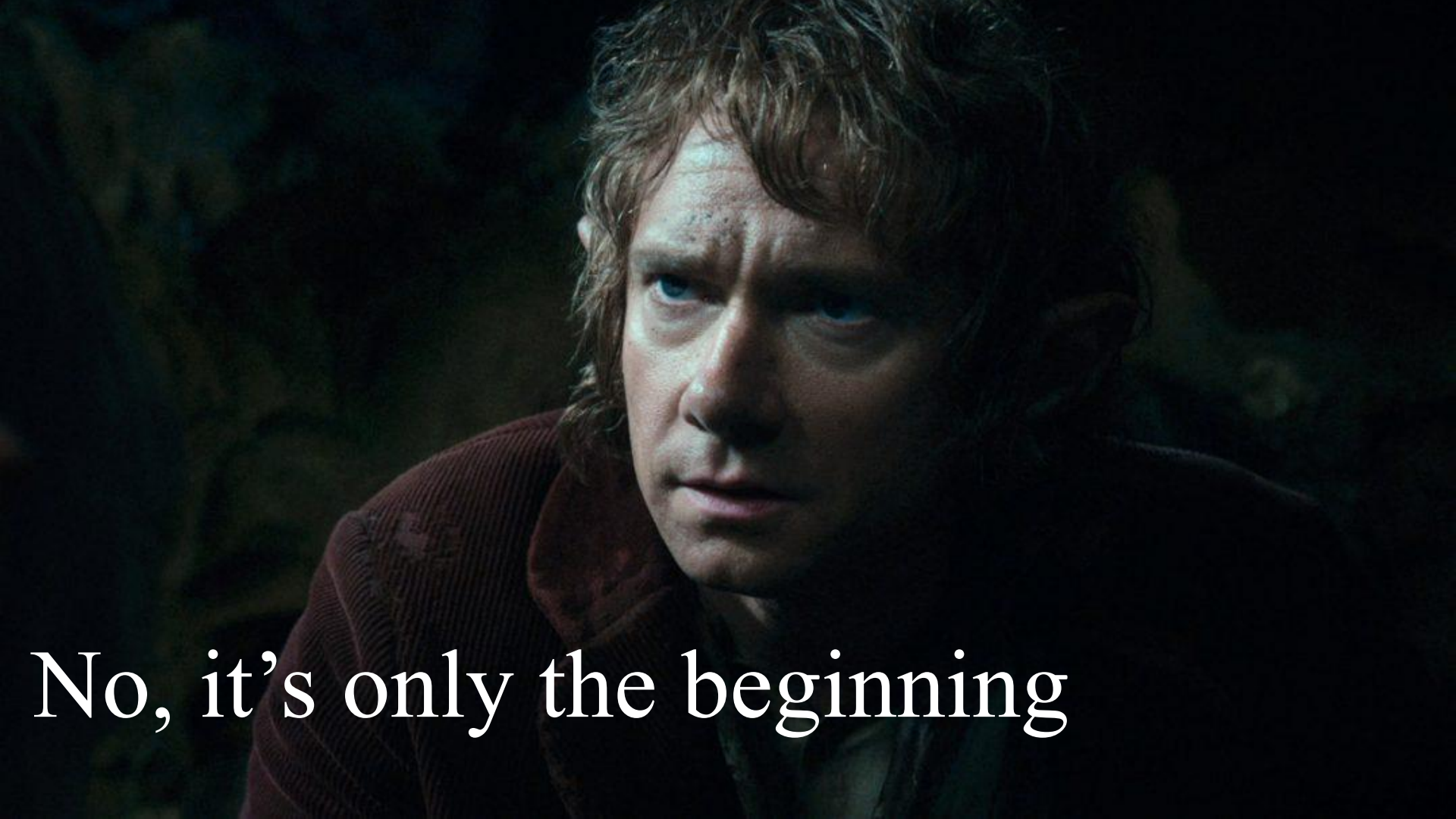
A cinematic scene featuring three orcs from 'The Hobbit' franchise. They are gathered around a large, crackling campfire at night. The orc in the center is stirring a large metal pot with a long-handled spoon. The other two orcs are looking on, one holding a small wooden bowl. The scene is lit with the warm, orange glow of the fire, contrasting with the dark, shadowy background of a forest.

And don't forget your  
fellow API designers



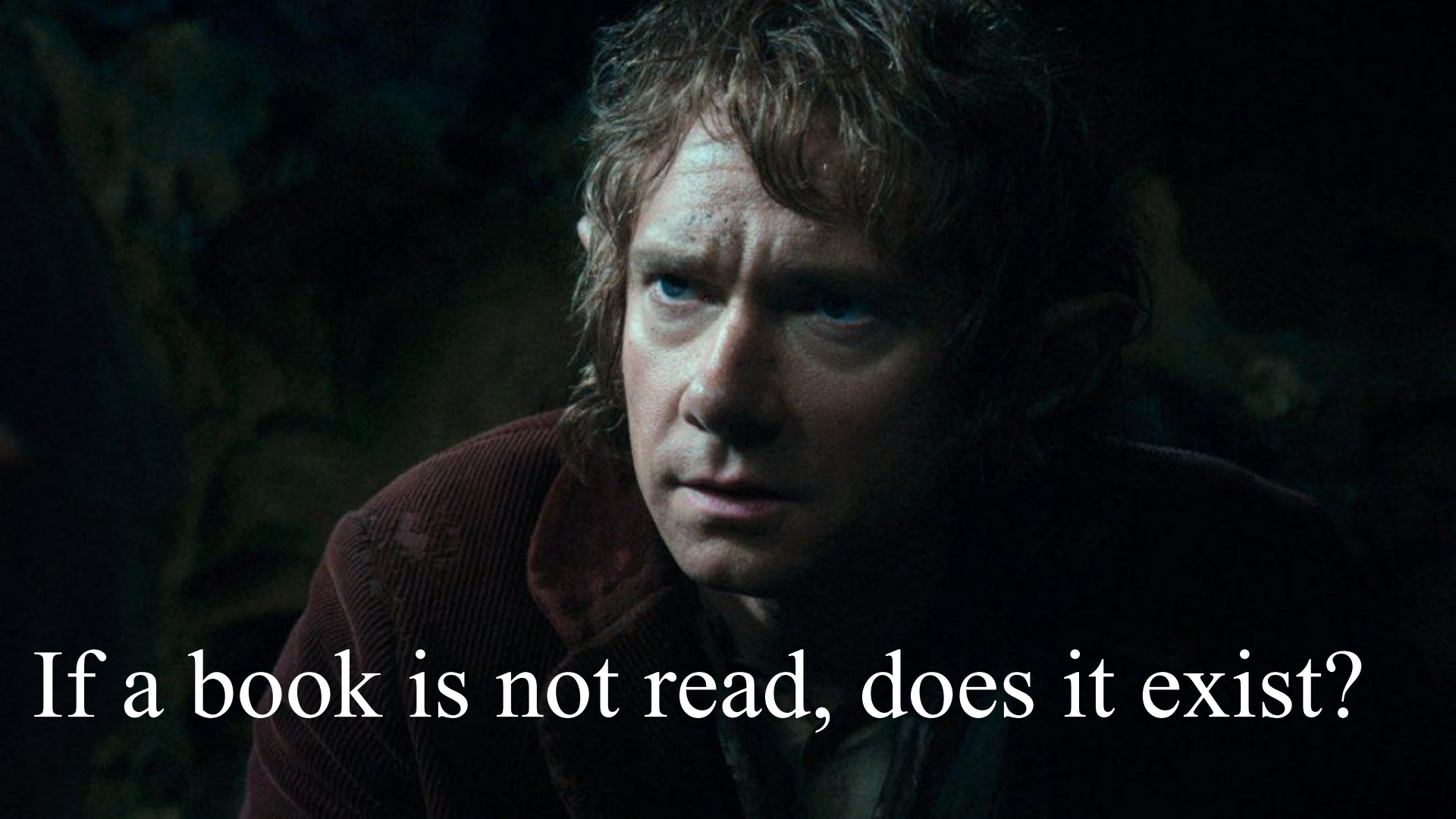


And it's done?



No, it's only the beginning





If a book is not read, does it exist?

A man with brown hair, wearing a dark red coat and brown trousers, is running barefoot through a lush green landscape. He is carrying a backpack and a basket. In the background, there are rolling green hills with several hobbit holes (small houses with round doors) built into them. A wooden signpost with three arrows pointing in different directions is visible on the left. The scene is set during sunset or sunrise, with a warm, golden light. The text "Share and promote" is overlaid in white serif font on the left side of the image.

Share and promote





What else?



API styleguide will need to evolve





Update  
Improve



Is that all?





One last thing



Rules may not be followed



A close-up shot of Gandalf the White, an elderly man with long, flowing white hair and a beard. He is wearing a brown robe and holding a wooden staff. The background is dark and out of focus. The text "Enforce and control" is overlaid in white serif font on the right side of the image.

Enforce  
and control



Is this  
control only  
done by people?





Mostly but some try to automate  
part of these controls...

## *Chapter IV*

# THE API DESIGN HAVENS





**OPEN API**  
INITIATIVE

```
paths:
  /persons:
    get:
      summary: Gets some persons
      description: Returns a list containing all persons. The list supports paging.
      parameters:
        - $ref: "#/parameters/pageSize"
        - $ref: "#/parameters/pageNumber"
      responses:
        200:
          description: A list of Person
          schema:
            $ref: "#/definitions/Persons"
        500:
          $ref: "#/responses/Standard500ErrorResponse"
    post:
      summary: Creates a person
      description: Adds a new person to the persons list.
      parameters:
        - name: person
          in: body
          description: The person to create.
          schema:
            $ref: "#/definitions/Person"
```

# Machine readable API description





# Zalando's Zally

## A minimalistic API linter

<https://github.com/zalando-incubator/zally>



Check if your [SWAGGER Schema](#) conforms to [Zalando's REST API Guidelines](#)

**BY URL** **EDITOR** **RULES**

ENTER FULL PATH TO YOUR SWAGGER FILE

`https://raw.githubusercontent.com/OAI/OpenAPI-Specification/master/examples/v2.0/json/petstore.json`

**VALIDATE**

**VIOLATIONS**

# Rule #174 is violated

■ **MUST** – Use common field names

`#/definitions/Pet`: field 'id' has type 'integer' (expected type 'string')

Rule: [/#174](#)

Paths:

- `#/definitions/Pet`

■ **MUST** – Define and Assign Access Rights (Scopes)



## API Guidelines

### Table of Contents

1. Introduction
  2. Principles
  3. General Guidelines
  4. Security
  5. Compatibility
  6. JSON Guidelines
  7. API Naming
  8. HTTP
  9. Performance
  10. Pagination
  11. Hypermedia
  12. Data Formats
  13. Common Data Types
  14. Common Headers
  15. Proprietary Headers
  16. Deprecation
  17. API Operation
  18. Events
- Appendix A: References
- Appendix B: Tooling
- Appendix C: Changelog

## Must: Use common field names and semantics [174]

There exist a variety of field types that are required in multiple places. To achieve consistency across all API implementations, you must use common field names and semantics whenever applicable.

### Generic Fields

There are some data fields that come up again and again in API data:

- **id** : the identity of the object. If used, IDs must opaque strings and not numbers. IDs are unique within some documented context, are stable and don't change for a given object once assigned, and are never recycled cross entities.
- **xyz\_id** : an attribute within one object holding the identifier of another object must use a name that corresponds to the type of the referenced object or the relationship to the referenced object followed by **\_id** (e.g. **customer\_id** not **customer\_number** ; **parent\_node\_id** for the reference to a parent node from a child node, even if both have the type **Node** )
- **created** : when the object was created. If used, this must be a **date-time** construct.
- **modified** : when the object was updated. If used, this must be a **date-time** construct.
- **type** : the kind of thing this object is. If used, the type of this field should be a string. Types allow runtime information on the entity provided that otherwise requires examining the Open API file.

Example JSON schema:

# Rule #174 is defined in guidelines

APIHandyman:zally arnaud\$ zally

NAME: code ZALLY  
Zally-CLI - Zally Command Line Interface

USAGE:  
zally [global options] command [command options] [arguments...]

VERSION:  
1.1

COMMANDS:  
rules List supported rules  
lint Lint given `FILE` with API definition  
help, h Shows a list of commands or help for one command

GLOBAL OPTIONS:  
--linter-service URL, -l URL Linter service URL (default: "http://localhost:8080") [\$ZALLY\_URL]  
--token OAUTH2\_TOKEN, -t OAUTH2\_TOKEN OAuth2 Token OAUTH2\_TOKEN [\$TOKEN]  
--help, -h show help  
--version, -v print the version

There's also an API and a CLI



roxspring Imposed stable ordering on /supported-rules

Latest commit 6ff6e68 21 days ago

..

📄 <a href="#">AbstractRule.kt</a>	server: implement vendor extension to ignore rules by code (#406)	3 months ago
📄 <a href="#">ApiValidator.kt</a>	server: implement api parameter to ignore rules by code (#406)	3 months ago
📄 <a href="#">AvoidLinkHeadersRule.kt</a>	server: Provide new guideline links and guideline codes #497	2 months ago
📄 <a href="#">AvoidSynonymsRule.kt</a>	adding id and link to AvoidSynonymsRule (#537)	a month ago
📄 <a href="#">AvoidTrailingSlashesRule.kt</a>	server: Provide new guideline links and guideline codes #497	2 months ago
📄 <a href="#">CommonFieldTypesRule.kt</a>	server: Provide new guideline links and guideline codes #497	2 months ago
📄 <a href="#">CompositeRulesValidator.kt</a>	server: implement api parameter to ignore rules by code (#406)	3 months ago

# You can define your own rules





So, no more  
humans?



Such tool really helps  
But we still need humans

*Epilogue*

# THE FELLOWSHIP OF API DESIGNERS



“**G0oD**” *API* Design  
is nothing without  
consistency



No consistency  
without  
clearly defined rules



Rules  
are nothing without  
governance  
(establish, share, update, control)



It's time to build your  
fellowship of API designers



# THE LORD OF API DESIGNS

BY

ARNAUD LAURET @APIHANDYMAN

THE END