

Can you finally REST?!

meet.php # 13
by: Max Małeckki @emgieszet

Cool developers:

- Mobile developers
- Node.js developers
- Ruby developers



Can I be as cool as they are?

**You write APIs for them.
Cool developers can't live without you!**

7 sins of your API

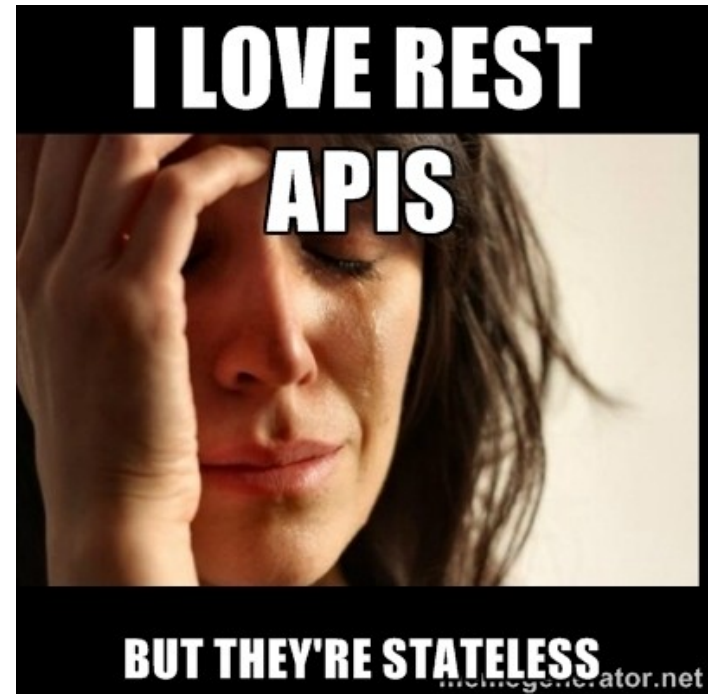
- What makes you uncool?
- Why „cool“ developers hates you?

0. Lack of design



1. Session Management

- RESTful – means stateless
- To authorize & authenticate use
 - OAuth v2
 - Wsse



2. Content type constraints

GET /api/dogs.json – 200

GET /api/dogs.xml – 404

- Return a response in a format defined in the headers of a request:

GET /api/dogs/123

Accept:

application/vnd.example+xml,

application/xml,

application/json

Accept-Language: de, fr, en

3. Api url's

`/api/dogs?id=123`

- Instead use uri templates
`/api/dogs/123`
- Get parameters are good for filtering
`/api/dogs?color=black`

3. Avoidance of HTTP Methods

`/api/dogs/show`

`GET /api/dogs`

`/api/dogs/new`

`POST /api/dogs`

4. Complexity

- Make it simple and universal

5. Lack of documentation

- If you got documentation in PDF, be sure that something is wrong.

6. Api is not cacheable

- Do you know:
 - varnish?
 - haproxy?
- So design your API to be cacheable
- PHP is not super fast consider the CACHE!

Quick tips

- Test your API – TTD is not that hard.
- Please, oh please use serialization
- Make sandbox for vendor developers

Richardson REST maturity model

- Level 0 - Swamp of POX
- Level 1 - Resources
- Level 2 - HTTP verbs
- Level 3 - Hypermedia controls

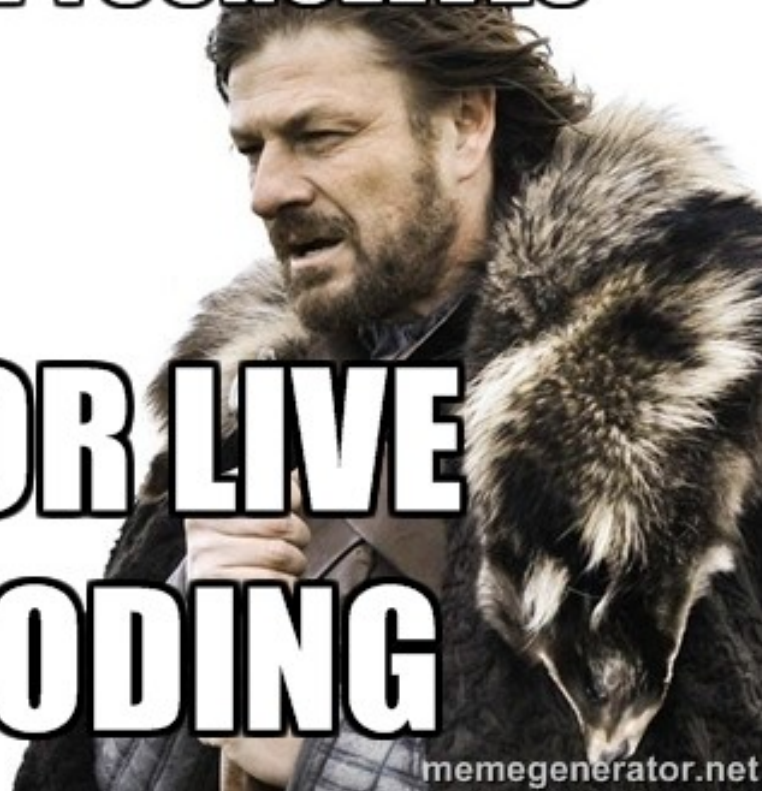
Richardson Model in Symfony2

- Level 0 & Level 1 & Level 2:
 - FOSRestBundle
 - NelmioAPIDocBundle
- Level 3:
 - BazingaHateoasBundle
- Authentication:
 - FOSOAuthServerBundle

Live Code

BRACE YOURSELVES

**FOR LIVE
CODING**



You want that code?



Questions?

Want more?

- <http://shishkin.github.io/presentations/rest/>
- <http://blog.steveklabnik.com/posts/2011-07-03-nobody-understands-rest-or-http>
- <http://martinfowler.com/articles/richardsonMaturityModel.html>
- <http://timelessrepo.com/haters-gonna-hateoas>
- <http://www.slideshare.net/kifahibrahim/symfony2-as-an-api>
- <https://speakerdeck.com/gordalina/rest-apis-made-easy-with-symfony2>

My Question:

Explain the HATEOAS acronym

Hypertext As The Engine Of Application State

Thank you!

You're cool developer?
Now you can finally REST!