

A grayscale silhouette of a person in mid-air, jumping over a gap between two dark, irregular shapes that represent ground or cliffs. The person's arms are outstretched forward and slightly upward, and their legs are bent in a jumping motion. The background is a light gray.

The way of a backend programmer

**a.k.a. Years of blood,
sweat and tears**

AR, Seznam.cz

Pilsen 2018



Who am I?

Alex Rembish

- Siberian
- Backend developer with 10+ years of experience
- Python enthusiast (6+ years)
- Senior Lead Programmer at **SKLIK.cz**
SEZNAM.CZ
- **Currently mostly working on YAML files**

Backend programmer?

`{ }`
BACKEND
DEVELOPER

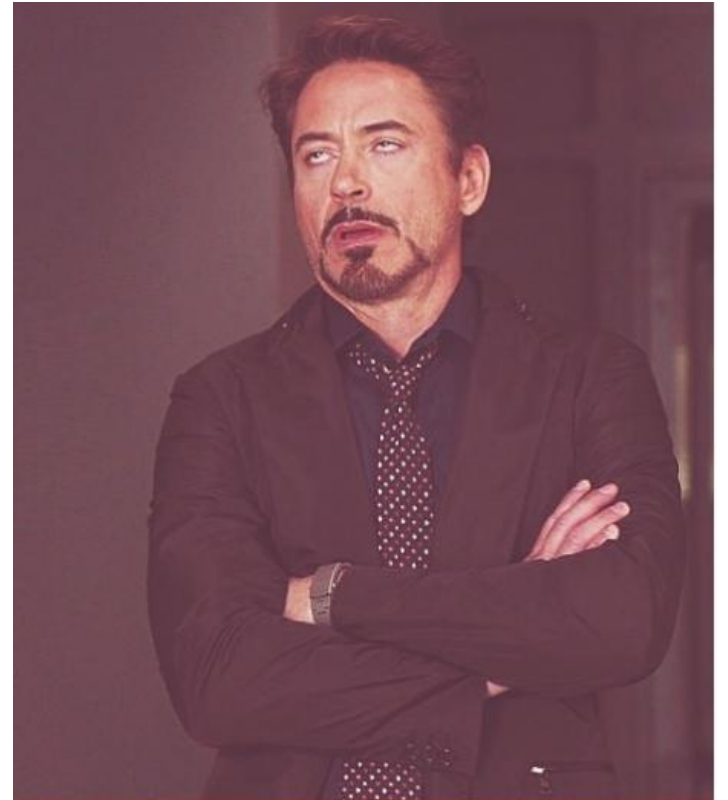
`< / >`
FRONTEND
DEVELOPER



Backend programmer?

TL;DR who doesn't want
to do frontend work

Difficulty Level: Jr.



Junior goals

1. **Coding itself**
2. **Gaining experience**
3. Playing with technologies
4. Making spare money
5. Prestige
6. ...
7. PROFIT?



Expectation of a Junior

1. **Desire to learn**
2. **Listening skills**
3. **Asking questions**
4. **Clean code**
5. ...
6. Getting shit done



GET
SHIT
DONE

Junior problems

Most useful feature you can imagine



Junior vs team problems

1. **Not asking questions**
2. Mess in knowledge
3. "Reinventing the Wheel" problem
4. Technology misuse/abuse
5. Dirty code



Team vs juniors problems

1. **Get shit done!**
2. **Uncontrolled development**
3. Too busy to explain
4. Unclear expectations
5. Unclear tasks
6. Unclear workflows
7. Cold start too cold
8. Undocumented environment
9. General delegation problems
10. ...



Level up!

1. Hard skills
2. Product knowledge



MULTI-CLASSING

Because wizards run out of spells

Difficulty Level: **MID**



Four horsemen of the Apocalypse

1. Routine
2. Overconfidence
3. Loss of interest
4. YOLO development





Three Years problem

**Developer's
burn-out after
X years***

* generally three years

Let's flip a coin



**Team develops
programmers,
but programmers
make a team**

Technical debt



Level Up! Level Up!

Routine \Rightarrow **Self-development**

Overconfidence \Rightarrow **Automation**

Loss of Interest \Rightarrow **Proficiency enlargement**

YOLO development \Rightarrow **Refactoring and reviews**



Difficulty Level:
Señor

Product problems

1. **Get shit done!**
2. **Uncontrolled development**
3. Too busy to explain
4. Unclear expectations
5. Unclear tasks
6. Unclear workflows
7. Cold start too cold
8. Undocumented environment
9. General delegation problems
10. ...



...and skills

DIY: Get shit done

1. **Talk about problems**
2. New features vs Refactoring ratio
3. Project bootstrapping
4. Basic dev documentation
5. Build and deployment automation
6. Preproduction environment
7. Basic tests
8. Legacy code reducing
9. ...
10. PROFIT



Level Up! Level Up! Level Up!

Routine \Rightarrow **Soft skills growth**

Overconfidence \Rightarrow **Bullet-proof environment**

Loss of Interest \Rightarrow **Product development**

YOLO development \Rightarrow **Easy to support and update**

CTO

Fullstack

Lead

Batman

**...and get ready
for the next level**

DevOps

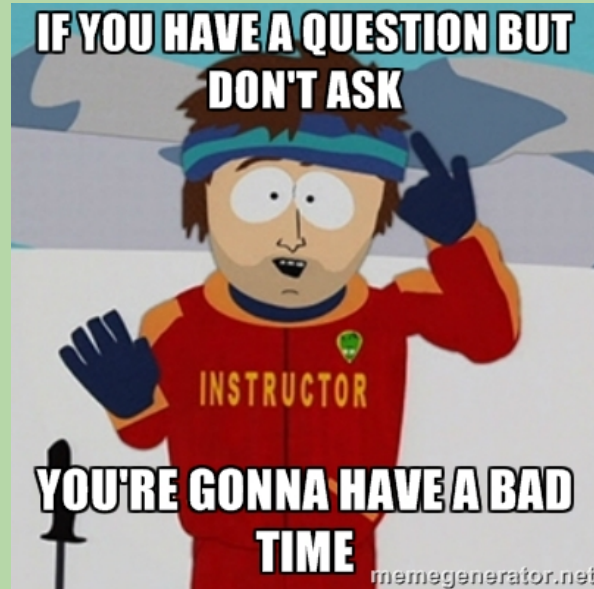
Product

Architect

If you're looking for
a company,
which promotes
self-development,
I know one :)



Q&A



Sincerely yours *YAML*
programmer

Twitter: **@rembish**

LinkedIn: **linkedin.com/in/rembish**

Presentation: **slides.com/rembish**

