

# Full Stack Practices in Rails Development

@yorzi

wangyaodi@gmail.com

**Andy Wang**

**Rails Developer Since 2008**  
**@Intridea**

# Agenda

5 mins + 5 mins + 35 mins

# **Rails Best Practice (5 mins)**

# Normally... we face

- Large & complicated applications.
- Teamwork and different coding styles.

# Then, code becomes

- Rigidity(僵硬)
- Fragility(脆弱)
- Imobility(固定)
- Viscosity(粘滯)
- Needless Complexity
- Needless Repetition
- Opacity(晦涩难懂)

# So, what's good code?

- Readability
- Flexibility
- Effective
- Maintainability
- Consistency
- Testability

# Most of Best Practices

<http://www.slideshare.net/ihower/rails-best-practices>

<http://www.slideshare.net/lcaliaLabs/rails-best-practices-14132100>

<https://speakerdeck.com/xdite/maintainable-rails-view>



# **Ralis Development Tips**

## **(5 mins)**

# Tip #1 - Use Git

- specially when coding on rails
- quick interation
- easy teamwork

# Tip #2 - Quick deployment

- Heroku for testing.
- Mina + Nginx + Unicorn for production.

<https://github.com/alfuken/mina-rails-unicorn-nginx-god>

# Tip #3 - Edit .gitignore

- secret\_token.rb
- settings

## Tip #4 - Edit .rvmrc

- rbenv VS. rvm
- .ruby-gemset & .ruby-version

# Tip #5 - Use Postgres.app

- whatever, it's just simple.

# Tip #6 - Add config in initializers

- generators.rb
- mailer.rb
- log\_formatter.rb

# Tip #7 - Fake seed for testing

- use faker
- create quality testing data in seed



# Tip #8 - Global Settings

- something like settingslogic.

# Tip #9 - Find Gems' Problem

- always check out issues of a gem if you firstly use it.
- hack it by yourself.

# **Tip #10 - Work first, then be correct.**

- do validations in last step.
- only integrate existing data and prepared logics.

# Tip #11 - Remove the useless

- blank actions.
- blank js/css/helpers files

**Real Practices From Real Code**  
**(35 mins)**

**Thank You!**