

# Advanced Software Engineering 2013

EatAndMove.herokuapp.com

Green Team

# Summary

- ▶ Our application
- ▶ Technologies used
- ▶ Project Management
- ▶ Lessons learnt
- ▶ Conclusion

# Our application

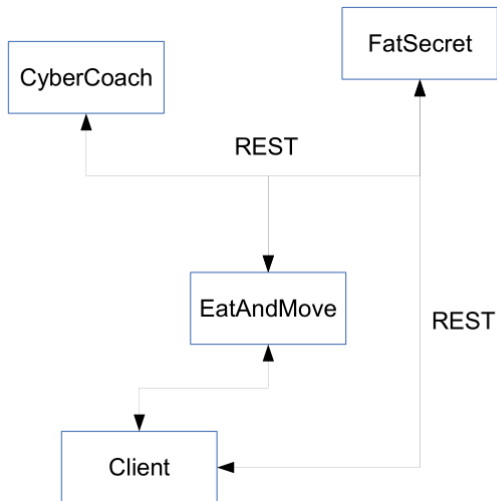
## Eat & Move is the combination of two web services

- ▶ With CyberCoach, you can manage your training sessions
- ▶ With FatSecret, you can monitor your calories consumption
- ▶ What you can do:
  - ▶ plan your exercise sessions;
  - ▶ calculate your daily calories needed;
  - ▶ coach your diet;
  - ▶ watch and manage your weight history.

Let's try!

<http://eatandmove.herokuapp.com>

## Behind the scenes



# Technologies used

## Languages/framework

- ▶ RoR 4.0.0
- ▶ Ruby 2.0.0
- ▶ HTML/CSS
- ▶ JavaScript

## Softwares

- ▶ JetBrains RubyMine (IDE)
- ▶ Notepad++
- ▶ Git
- ▶ Wireshark
- ▶ nodejs (Windows)

# Technologies used cont'd

## Platform as a service

- ▶ Heroku

## Revision control

- ▶ Github

## WebServices

- ▶ CyberCoach
- ▶ FatSecret

# Technologies used cont'd

## Gems

- ▶ fatsecret-omniauth (FatSecret REST)
- ▶ Activeresource (RESTful resources as model)
- ▶ (HTTParty)
- ▶ SQLite3/PostgreSQL (database dev/prod)
- ▶ Nokogiri (XML parser)
- ▶ turbolinks (removed)



## How to include a new webservice?

- ▶ FatSecret is a platform:
  - ▶ for accessing and distributing nutrition information;
  - ▶ exercise and weight management tools
- ▶ an API is available (REST + JavaScript)
- ▶ communication
  - ▶ profile: `oauth_token` and `oauth_secret`
  - ▶ REST: `session_key`

# Project Management

## How we planned it!

- ▶ Get familiar with the topic (1 week)
- ▶ Collect the needed Data (1 week)
- ▶ Application Logic (2 weeks)
- ▶ Friendly Interface (2 weeks)
- ▶ Testing, debugging (1 week)
- ▶ Scrum method (1 sprint = 1 week)
- ▶ team work (Github) + some pair programming if needed (4 eyes)
- ▶ rails testing suite

## How we done it!

- ▶ Get familiar with the topic (0.5 week)
- ▶ Collect the needed Data (0.5 week)  
→ **FatSecret**
- ▶ Application Logic (**4 weeks**)
- ▶ Friendly Interface (**1.5 week**)
- ▶ Testing, debugging (**0.5 week**)
- ▶ Scrum method (1 sprint = **3 days**)
- ▶ team work (Github) + **a lot of** pair programming (4 eyes)
- ▶ ~~rails testing suite~~

# Lessons learnt

## Difficulties encountered

- ▶ ROR is powerful but has to be learned
- ▶ development environment VS Heroku (fonts, database, assets, ...)
- ▶ activereource and CyberCoach (HTTParty, wireshark, time, userid VS username)
- ▶ FatSecret session key integration

## We appreciate. . .

- ▶ team work
- ▶ programming sessions during lecture time
- ▶ activereource example

# Conclusion

- ▶ after weeks of work, we can propose a working web application
- ▶ the application respects our expectations
- ▶ very time consuming project
- ▶ we know the RoR basics
- ▶ integrating web services is not as easy as it seems