

# README

Oleg Sivokon

October 2, 2013

## Contents

<b>1</b>	<b>Original Text of the Assignment</b>	<b>1</b>
1.1	Walla! Backend Engineers Test Assignment . . . . .	1
1.1.1	General Guidelines: . . . . .	2
1.1.2	The Test Assignment: . . . . .	2
<b>2</b>	<b>What Needs to Be Done</b>	<b>3</b>
2.1	<b>DONE</b> Install and test Tornado <b>:setup:</b> . . . . .	3
2.2	<b>CANCELED</b> Make sure MySQL binding is up-to-date <b>:setup:alternatives:</b>	3
2.3	<b>DONE</b> Maybe try py2neo <b>:setup:alternatives:</b> . . . . .	3
2.4	<b>TODO</b> Figure out what Apache httpd module to use <b>:setup:info:</b>	3
2.5	<b>TODO</b> Brush up on Tweeter Bootstrap <b>:setup:info:</b> . . . . .	4
2.6	<b>TODO</b> Find out what most people use for unit testing <b>:setup:info:</b> . . . . .	4
2.7	<b>TODO</b> Ropemacs was giving me headache lately <b>:setup:</b> . .	4
2.8	<b>DONE</b> Start a GitHub repo (and fix the setup) <b>:setup:</b> . . .	4
<b>3</b>	<b>Project Description</b>	<b>4</b>
3.1	Nodes and relations I will need . . . . .	5

## 1 Original Text of the Assignment

### 1.1 Walla! Backend Engineers Test Assignment

Heya, In order to evaluate your skills as a backend Web engineer, we would like you to take this short development assignment, at your own free time.

### 1.1.1 General Guidelines:

1. Please provide
  - working code.
  - a brief design document explaining what you did.
  - a link to a server where your code can be tested.
2. Feel free to search the Web for inspiration, but please do not copy code from anywhere, keep it 100% yours.
3. Use Python as your language of implementation. Micro Frameworks and utility libraries are allowed, but not a full blown MVC.

### 1.1.2 The Test Assignment:

You are to design and implement the API and server side modules for a simplified twitter like social updates site, and a simple AJAX client for it. The application contains users; Each user can post short text messages, follow other users, and get a feed of the latest updates from the users he is following. You can also get a global feed for all the users.

1. Implement an HTTP based (ReSTful in the loose sense of the term) API that exposes the following calls (no need for authentication, choose the format you like):
  - `CreateUser(Username)`
  - `PostMessage(UserId, MessageText)`
  - `Follow(FollowingUser, FollowedUser)`
  - `Unfollow(FollowingUser, UnfollowedUser)`
  - `GetFeed(ForUserId)` *Note the aggregate feed of the users this user is following!*
  - `GetGlobalFeed()`
2. Please create an ajax based HTML page for this API:
  - **Interface** It should support getting the messages from the followed users per user Id, with real time updates, and the ability to post messages. You do not need to implement following/unfollowing. No styling required.

- Implementation Use of toolkits (jQuery, Zepto, bootstrap..) is fine, but no frameworks plz (Angular, Backbone..). No need for mobile adaption. Strive for efficiency and readability.

Good Luck! The Walla! Backend Engineering Team

## 2 What Needs to Be Done

### 2.1 DONE Install and test Tornado :setup:

DEADLINE: *2013-10-01 Tue*

- State “DONE” from “TODO” *2013-10-02 Wed 08:20*  
This was the easy part.

Was suggested during the interview

### 2.2 CANCELED Make sure MySQL binding is up-to-date :setup:alternatives:

DEADLINE: *2013-10-01 Tue*

- State “CANCELED” from “TODO” *2013-10-02 Wed 08:21*  
I’ll be using neo4j, that’s decided.

I have some vague memories of it being problematic to install, perhaps it required `mysql-devel` package? Can `setuptools` install RPM / DEB?

### 2.3 DONE Maybe try py2neo :setup:alternatives:

DEADLINE: *2013-10-02 Wed*

- State “DONE” from “TODO” *2013-10-02 Wed 08:21*  
Yup.

Just for the sake of experiment? There are couple of videos to watch. If I can get it to run easily, I’d probably use that.

### 2.4 TODO Figure out what Apache httpd module to use :setup:info:

DEADLINE: *2013-10-01 Tue*

There is some controversy about WSGI / FSGI / `mode_python` Need to make sure I’m not using some outdated / unsupported library.

## 2.5 TODO Brush up on Tweeter Bootstrap :setup:info:

DEADLINE: *2013-10-01 Tue*

This is most likely candidate for the GUI. Need to see what components are already there, and what's missing (unlikely).

## 2.6 TODO Find out what most people use for unit testing :setup:info:

DEADLINE: *2013-10-02 Wed*

We'll see what Stackoverflow has to say...

## 2.7 TODO Ropemacs was giving me headache lately :setup:

DEADLINE: *2013-10-02 Wed*

See if the stupid thing still works. Maybe we shall do w/o it. There was some other package... Probably elpy or some such, maybe it makes sense to check it out.

## 2.8 DONE Start a GitHub repo (and fix the setup) :setup:

DEADLINE: *2013-10-02 Wed*

- State "DONE" from "TODO" *2013-10-02 Wed 08:21*  
Looks like it also can now display Org files markup, so no need to export to Markdown

Not today.

## 3 Project Description

Right now I'm thinking about making it into a nomic game. The basic idea would be that by posting users may, in general, score points. More so, if someone comments on their posts, or if they themselves comment. But it would be also possible to loose points due to bad posts or comments. Sort of karma, just like there is one on the StackExchange sites.

I'm definitely going with `py2neo`, especially because it uses `tornado`. It's not as efficient as SQL (at the moment), but who cares?

### 3.1 Nodes and relations I will need

Nodes

- `User#{ "id": int, "name": string, "password": string }`
- `Message#{ "id": int, "text" : string, "score": int }`

Relations

- *user Follows user*
- *user FollowedBy user*
- *message PostedBy user*
- *user AuthorOf message*
- *user Rates(points) message*