

BlogBooker

Low resolution pictures

From Blog to Book.

sdd-dev.tumblr.com

Contents

1	2014	9
1.1	February	9
	work cheatsheet (2014-02-13 5:59)	9
	Wirth ideas (2014-02-16 6:16)	9
	C++ Q (2014-02-26 9:26)	10
	ВНЕЗАПНО (2014-02-27 7:20)	10
	[JOB] сокрушить супербаг (2014-02-28 9:09)	11
1.2	March	11
	[JOB] супербаг, ты под прицелом (2014-03-04 2:09)	11
	Job progress (2014-03-05 10:36)	12
	Снова работа (2014-03-11 13:39)	12
	Лекарство от фрустрации (2014-03-11 13:50)	12
	CVS branches (2014-03-11 14:28)	13
	Снова работа (2014-03-11 3:49)	14
	Первая победа! (2014-03-12 4:48)	15
	CVS (2014-03-12 6:17)	15
	Анализ систем (2014-03-13 6:23)	15
1.3	May	16
	Планы (2014-05-07 9:51)	16
	you know my name (2014-05-15 4:59)	17
	yandex algorithm 14 warmup (2014-05-16 13:22)	17
	Job & Work (2014-05-16 6:53)	18
	Ubuntu Crash (2014-05-20 15:34)	18
	Kindle DX и теория (2014-05-23 7:41)	18
	Ajax for columns (2014-05-24 16:44)	19
	Теория (2014-05-24 16:49)	19
	(2014-05-27 10:12)	19

	Смена отношения (2014-05-27 10:28)	19
	тормоза (2014-05-28 5:24)	20
1.4	August	20
	SCO (2014-08-18 14:43)	20
	Virtual Memory (2014-08-26 14:33)	20
	Yarly IM (2014-08-26 5:18)	21
	Segments, C++ (2014-08-27 6:05)	21
1.5	September	22
	Yarly (2014-09-04 16:05)	22
	Yet Another ReLaY (2014-09-19 16:08)	23
1.6	October	23
	(2014-10-18 3:16)	23
	Ресурсы (2014-10-22 13:23)	23
	Цели (2014-10-22 13:59)	23
	Принципы (2014-10-22 13:59)	24
	Разделяй и властвуй (2014-10-22 14:44)	24
	Tech stack for a web service with python and C++ (2014-10-31 15:54)	24
	Вопросы проектирования (2014-10-31 16:22)	25
1.7	November	26
	Баловство (2014-11-01 12:15)	26
	Профессионализм (2014-11-01 15:37)	26
	(2014-11-02 12:20)	26
	(2014-11-02 12:21)	27
	yoda frontend deploy part 1 - wsgi.djangoserver (2014-11-02 15:16)	27
	Домен (2014-11-02 15:35)	28
	+ django (2014-11-03 10:50)	28
	Веб Фреймворки для python (2014-11-03 14:17)	28
	А утром всё сломалось... (2014-11-03 3:33)	29
	Клиент yoda (2014-11-03 4:45)	29
	Домен (2014-11-03 4:50)	30
	Великие умы мыслят одинаково (2014-11-03 6:36)	30
	Неуникальность? (2014-11-03 7:27)	30
	Первый рабочий день в состоянии оцепенения (2014-11-05 10:54)	31
	make clean (2014-11-05 10:57)	32
	Джобс (2014-11-05 11:29)	32

Лог (2014-11-05 6:29)	32
Отношение к работе (2014-11-05 6:31)	32
Домен (2014-11-05 7:15)	32
sid (2014-11-05 8:22)	33
dev (2014-11-05 9:01)	33
Сломанное обновление по F5 (2014-11-05 9:13)	33
Языки Программирования (2014-11-06 17:04)	33
Программы людям (2014-11-07 16:46)	34
yearly (2014-11-07 5:37)	34
#9878 (2014-11-07 8:48)	35
static IP (2014-11-08 8:49)	35
quotrx+yoda: baby steps (2014-11-10 13:04)	35
F5 (2014-11-10 14:08)	35
python (2014-11-10 14:10)	36
plain text (2014-11-10 5:40)	36
Boost и Boost.Python (2014-11-11 15:39)	37
Сломанные пути при наличии двух экземпляров одной сессии (2014-11-11 8:29)	37
Management in ASoft (2014-11-12 9:04)	38
Проект (2014-11-13 3:35)	38
Интриги (2014-11-13 4:26)	39
asoft colleagues (2014-11-14 8:44)	39
quotrx.com (2014-11-19 11:05)	39
Сегодня я многое узнал (2014-11-19 12:27)	39
(2014-11-20 13:07)	40
Находки на хабре PART1 (2014-11-20 6:40)	41
Находки на хабре PART2 (2014-11-20 6:41)	42
hardware materials (2014-11-22 11:18)	43
Как связать C++ с MySQL на Ubuntu лучше всего? (2014-11-22 12:19)	44
Как устроен MySQL (2014-11-22 14:09)	45
Последний элемент (2014-11-22 14:14)	45
(2014-11-22 6:50)	46
Подвиги при сборке Conn/C++ (2014-11-23 14:57)	46
Несокрушимый) (2014-11-24 14:14)	47
Проекты (2014-11-24 16:24)	47
ListResult (2014-11-24 9:50)	48

(2014-11-25 13:47)	48
Stellart Arch Research (2014-11-26 7:30)	49
План, чтобы сделать план (2014-11-29 14:44)	49
(2014-11-30 10:10)	50
(2014-11-30 10:43)	50
OpenID (2014-11-30 10:44)	51
OpenID Connect (2014-11-30 11:38)	51
(2014-11-30 12:23)	51
The Myth of the Full-stack Developer (2014-11-30 12:56)	51
1.8 December	52
CMS для quotrx (2014-12-01 10:48)	52
PaaS vs IaaS vs VPS (2014-12-01 13:14)	52
yoda+quotrx (2014-12-01 14:02)	53
Проклятье Бэббиджа (2014-12-01 15:12)	54
django-mysql (2014-12-02 11:35)	54
Mezzanine (2014-12-02 5:37)	55
The Game of Fear (2014-12-02 8:55)	55
Имя проекта (2014-12-03 13:55)	55
(2014-12-03 3:26)	56
Django setting (2014-12-03 8:31)	56
Memento: deploy weak points (2014-12-03 8:37)	57
HOW TO DEPLOY Guide (2014-12-04 3:47)	58
Python Import (2014-12-04 8:28)	59
django admin templates (2014-12-07 14:47)	59
Linux (2014-12-07 15:25)	60
(2014-12-09 10:03)	60
Django project structure (2014-12-09 6:20)	61
mezz wysiwyg (2014-12-09 9:01)	62
Итог на обед (2014-12-10 11:51)	62
SOLID (object-oriented design) - Wikipedia, the free encyclopedia (2014-12-10 13:34)	63
Вопросы по mezz (2014-12-10 9:38)	63
Терминальные утилиты (2014-12-10 9:38)	64
(2014-12-11 14:45)	64
Upgrade (2014-12-12 14:50)	64
JPEG-isation of work (2014-12-14 10:34)	65

VirtualBox VM quickstart w/ LAMPDM (2014-12-15 5:23)	65
Программирование (2014-12-19 17:02)	66
dev ecology (2014-12-21 18:05)	67
Jedi Programmers (2014-12-21 18:08)	67
Django 1.7 migrations (2014-12-23 12:27)	67
Heartcore Dev (2014-12-23 17:24)	68
Microsoft (2014-12-25 15:38)	68
Блог (2014-12-25 17:28)	68
Бессонница (2014-12-25 19:26)	69
Ментальные инструменты для фильтрации шлака (2014-12-26 12:35)	69
Когда-нибудь у меня будет public dev blog (2014-12-26 17:32)	69
Python 3.4 Django 1.7 (2014-12-27 0:22)	70
learning curve (2014-12-27 3:11)	70
Django Tango Rango (2014-12-27 4:32)	70
import Dj models (2014-12-28 9:20)	71

1. 2014

1.1 February

work cheatsheet (2014-02-13 5:59)

* cvs/git/hg server
git.vzhost.ru

* TO UP EFFI
get-effi.sh
yum update *.rpm

* to add repo
yum-config-manager --add-repo repository_url

* look for installed effi version
rpm -qa | grep effi

* adjust eclipse key bindings

* convert file encoding
iconv -f ISO_8859-5 -t UTF-8 -o errmgr.h errmgr.h

* miel6c2 logs dir
ssh root@miel6c2; pw 12root; cd home/crm/logs

* hw management
lspci, lshw

* yum list installed

* sudo rpm -ihv libole2-*

Wirth ideas (2014-02-16 6:16)

Link: [1]Wirth ideas

1. <http://citforum.ru/programming/digest/wirth/>

C++ Q (2014-02-26 9:26)

В рамках класса при вызове метода класса полиморфизм работает?

[1]Код на ideone[2]

Ответ: ДА.

Так что я всё ещё не понимаю, как достаётся uid по методу CommObject из GlobalData, если есть более перекрытый метод в RemoteObject, который делает нечто другое. Завтра надо будет всё это тщательно изучить, иначе непонятно кто и под каким uid что вызывает, а это ключевая часть отладки.

1. <http://ideone.com/U4cB6o>

2. <http://ideone.com/RDakZj>

ВНЕЗАПНО (2014-02-27 7:20)

Я кучу времени потратил на изучение системы и теперь с её пониманием нет проблем и я могу быстро по ней навигировать и высокоуровнево её осознать.

Отслеживать цепочки исполнения для изучения системы больше смысла нет.

Описание проблемы и, следовательно, её решение на 90 % ясны.

Нужно за что-то зацепиться в проблеме, чтобы понять, что я упускаю в ней, что не даёт мне получить полное понимание и ясность.

Возможно, сегодня я уже сделал всё что мог.

Что-либо делать дальше нужно начинать с описания проблемы в багтрекере.

Пока не осознаю до конца, бессмысленно дальше что-то делать.

Вопросы по багу:

1) почему после перезапуска контейнера (=> восстановления соединения с БД), я всё равно ЧАСТО или ПОСТОЯННО КАКОЕ-ТО ВРЕМЯ не могу зайти последним пользователем (тем на котором всё сломалось)?

2) ЧАСТО или ПОСТОЯННО КАКОЕ-ТО ВРЕМЯ?

3) от чего зависит время или частота в п.2?

[JOB] сокрушить супербаг (2014-02-28 9:09)

Баг оказался состоящим из трёх частей.

1) Перед крушением контейнера в кэше `ars` создаётся битый пользователь. Это маловажно, т.к. контейнер всё равно перезагружается и весь кэш стирается. Это никак не влияет на работу системы по сути.

МБ следует не создавать его, если контейнер умер.

2) Проверка прав в рухнувшем контейнере проходит так же, что и при нарушении прав доступа. Возвращается то же исключение, что вводит пользователя в заблуждение, когда оно отображается после экрана логина. Следует посылать другое исключение и говорить пользователю что контейнер умер.

3) После перезагрузки сбойнувшего контейнера в `WWW->SessionRepo` есть `UID`, но нет роли. Каким-то образом этот сбой переходит в `CRM _Container`, где появляется пользователь без прав, которому нельзя смотреть `Main-Page` (как и другие страницы).

Следует разобраться, как битые права передаются в отрисовку и как это предотвратить. Наиболее очевидное решение - убивать пользователя в `SessionRepo` в случае краша контейнера (если получили `UID`, но не роль).

Вопросы:

1) Как в контейнере CRM появляется битый пользователь?

2) Попадает ли он туда из `WWW->SessionRepo`?

3) В `WWW->Login` при `access _violation` срабатывает `catch`, который пишет пустые права в `user-a #20001`?

1.2 March

[JOB] супербаг, ты под прицелом (2014-03-04 2:09)

Супербаг найден и никуда не денется! Осталось его убить.

Я хочу показать свою работу Мише, что бы он ни говорил про "зачем вы ко мне обращаетесь". Всё-таки, он здесь давно и я работаю под его патронатом.

После этого надо будет думать, как хирургическим образом баг пофиксить, ничего не задев и точно везде его нейтрализовав.

Я думаю, это делается так: надо научиться распознавать ситуацию, когда контейнер, к которому обращаешься, упал. И эту ситуацию надо корректно обработать в `WWW->SetRole`, бросании исключения в CRM и проверке пользователя в CRM в 1й и 2й раз.

Что касается `CC`, то там кривой пользователь не создаётся, так как контейнер умирает раньше.

Job progress (2014-03-05 10:36)

Сегодня я мощно продвинулся, но пока нет коммита, ничто не готово. Я хочу всё сделать как надо сразу - потом не будет. Поэтому я проявлю мужество и доведу дело до конца.

Для этого мне нужно

- 1) осознать, что происходит при цепочке логинов, наглядно это представить, понять почему что показывается
- 2) понять, как админ логинится во 2й раз без краша
- 3) воспроизвести и пофиксить, если нужно, ситуацию полулогина админом

Я сделаю это и моя совесть будет чиста.

Снова работа (2014-03-11 13:39)

Сегодня я сначала 2 часа валял дурака на работе - за это мне стыдно, но запишем в прокрастинацию от усталости от сложной задачи.

Потом я пробовал найти нормальный клиент CVS, не разобравшись с интегрированным в Eclipse.

Потом я разбирался с интегрированным в Eclipse.

И, наконец, на излёте дня, я разбирался с CLI клиентом.

Меня сильно выбила из колеи эта тема, отняла много сил. И я плохо понимаю, в чём тут дело пока что.

Я думаю, что я восстановлюсь и у меня получится лучше проанализировать день. Чтобы я знал, как настроиться. Чтобы знал, как относиться к таким моментам. Чтобы знал, что делать дальше.

После того как отдохну, я уверен, всё наладится.

Ячитаю отзывы в интернете, осмыслю свой опыт и пойму каким инструментом пользоваться. А после этого прочитаю по нему полное хорошее руководство и прикину, что мне завтра нужно сделать, чтобы закоммитить код правильно.

Лекарство от фрустрации (2014-03-11 13:50)

Оно просто и, в то же время, сложно. Как остановиться во время бега.

Работая, можно впасть в иллюзию. Начать проецировать свои ожидания на график вместо оценок, исходящих из опыта и фактов.

Так и было сегодня.

Как только я впал в иллюзию (в пятницу вечером), что всё готово, что разбираться с незнакомой мне системой контроля версий почему-то не надо - я сам создал себе фрустрацию.

В пятницу я устал и я очень хотел думать, что победа в моих руках, что во вторник всё будет легко. И вот, сегодня я упёрся в реальность. И что же произошло? Я не поздоровался с реальностью и не попрощался с иллюзией. Я не ударил своё воображение по рукам. Я предпочёл поверить иллюзии того, что я знаю что делать, того что всё выйдет просто - наперекор реальности. И, поверив, всё что мне оставалось - это метаться, беситься и рейжить в нетерпении. Как будто брутфорсом можно было заставить всё работать.

Брутфорс... Не моё это. Никогда он меня не выручал. Иллюзия создаёт эту цепочку:

нетерпение (предпочтение иллюзии реальности)
брутфорс (делать тогда когда надо думать или читать)
фрустрация (разочарование что квадрат не вбивается в круг)

И эта цепочка подпивает сама себя. Это порочный круг, который я увидел только сейчас. Нетерпение порождает брутфорс порождает фрустрацию порождает нетерпение.

CVS branches (2014-03-11 14:28)

1. work with branches (switch local repo to branch)

1) [1]source

2) [2]source2

3) [3]source3

4) [4]source4

5) [5]source5

6) [6]source6

2. look what branches exist on server

[7]source

[8]source2

3. mans & tuts

[9]source

[10]source2

[11]source3

[12]book

[13]tut

[14]docs

4. [15]Meld

1. <http://www.astro.princeton.edu/~rhl/cvs-branches.html>
2. http://cvsgui.sourceforge.net/howto/cvsdoc/cvs_5.html
3. <http://users.csc.calpoly.edu/~jdalbey/205/Resources/cvsBranchMerge.html>
4. http://glacier.lbl.gov/DAQ/cvs_branch.html
5. <http://ioctl.org/unix/cvs/branches/>
6. <http://www.network-theory.co.uk/docs/cvsmanual/Branchesmotivation.html>
7. http://stackoverflow.com/questions/566093/how-do-i-identify-what-branches-exist-in-cvs#=_
8. http://versioncontrol.wikia.com/wiki/List_history_on_branch
9. http://commons.oreilly.com/wiki/index.php/Essential_CVS/Using_CVS/Tagging_and_Branching
10. https://www.cs.utah.edu/dept/old/texinfo/cvs/cvs_18.html
11. <https://www.cs.duke.edu/courses/fall02/cps108/resources/cvs/man.html>
12. [http://cvsbook.red-bean.com/cvsbook.html#Finding%20Out%20What%20You%20\(And%20Others\)%20Did%20--%20update%20And%20diff](http://cvsbook.red-bean.com/cvsbook.html#Finding%20Out%20What%20You%20(And%20Others)%20Did%20--%20update%20And%20diff)
13. <http://c2.com/cgi/wiki?CvsTutorial>
14. <http://www.nongnu.org/cvs/>
15. <http://meldmerge.org/>

Снова работа (2014-03-11 3:49)

И снова аццкая фрустрация динозаврами. Как можно блять делать diff через консоль?

К сожалению, сейчас я не могу просто взять и переделать всю среду, в которой разрабатываю.

У меня накопилась куча моментов, которые меня напрягают - контроль версий, отладка, версия среды разработки и т.д. Но сейчас на это нет времени.

Поэтому придётся выживать локальными мерами. Подключить тулзу для локального диффа.

Я думаю, после того как я убью все эти вступительные баги, я выбью неделю на то чтобы сделать всё по-своему, насколько мне это позволит моё небольшое влияние. Время убунту может и не настать. Время гита настанет нескоро. Но я смогу поставить Eclipse Indigo с плагинами для CVS, наладить отладку через IDE и gdb, наладить

IDE и написать необходимые скрипты.

Пока же - задача найти внешний клиент для CVS, этим я и займусь.

Первая победа! (2014-03-12 4:48)

Михаил меня по-своему немного похвалил, но и отметил проблемы:

1) недостаток инициативы - я спрашиваю, что делать. отчасти я прав - надо знать, как делают они. отчасти я неправ - неуверен в себе.

ЛЕКАРСТВО: читать про tcp/ip на выходных

2) недостаток абстрактного восприятия системы - я слишком полагаюсь на конкретную конфигурацию - отчасти я прав: я другого не видел. но и неправ отчасти - надо трансцендировать текущее.

ЛЕКАРСТВО: читать мануалы сегодня и мб взять домой и мб читать вики.

3) недостаток скорости - слишком много отвлечений, прокрастинации, торможения. отчасти я прав - новая система, настройка под себя, линукс и терминал новы. отчасти неправ - нарушен баланс энергии, не восстанавливаюсь вовремя. сбивается режим, появляется прокрастинация.

ЛЕКАРСТВО: моя история, общение, покупки.

ИТОГО со всем можно справиться - сегодня мануалы, на выходных tcp/ip, всё время - история и сегодня я приведу Ивана Поле к людям.

CVS (2014-03-12 6:17)

1) После коммита и билда старое задание закончилось.

2) Поставил всё-таки MELD. Это было не так просто, потому что версия из репозитория f18 была глючная, а новую пришлось ставить вручную, вручную разрешая зависимости.

3) Интегрированный в Eclipse CVS удалил, забэкапив строку подключения.

Я справился. Теперь у меня есть:

1) CVS CLI

2) MELD для diff и merge

Теперь на сегодня - читать мануалы, смотреть баг, искать где rdbms подключается к базе.

Анализ систем (2014-03-13 6:23)

Многие ситуации в анализе систем - программных, общественных, юридических, психологических - проявляют, что рядовой человек к такому анализу плохо приспособлен.

Память, в которой необходимо держать сущности и их признаки, по которым можно установить связи - эта память очень ограничена. И анализ с нуля занимает огромное время, потому что соединяя точки A и Z через B-C-D-...-X-Y, ты каждые минут 15 переполняешь свою память и забываешь, в каком ты месте.

В такой ситуации единственный выход - использовать инструменты анализа, упрощающее исследование системы. Это могут быть внутренние мнемонические техники как в "Moonwalking with Einstein", внешние мнемонические техники - рисование и перерисовывание схем с каждым добавлением фрагмента знания о системе, использование аккумулированных знаний о системах такого рода - тематическая литература, которая даёт набор шаблонов, нужных для понимания систем.

Но внутренняя и внешняя мнемоника очень трудоёмки. Литература же - во-первых, сама является системой: правильную информацию не всегда просто найти, упорядочить и усвоить; во-вторых, усваивание шаблонов, хотя и менее трудоёмкое чем брутфорс системы мнемоникой, тоже требует немало времени; в-третьих, любая система является линейной комбинацией многих шаблонов и каждый из них надо понимать; в четвёртых, кроме линейной комбинации возможны мутации от шаблонов.

Всё это делает востребованными идею инструментов анализа нового поколения - автоматически дополняющихся семантических иерархий с возможностью интерактивной работы с пользователем, визуализацией.

1.3 May

Планы (2014-05-07 9:51)

1) АСОФТ мне нравится профилем, но с такой средой это не вариант. Но все равно, удивительно, но лучше мейла.

2) У меня завал в сторону теории из-за того, что я думаю о себе как о наукоёмком программисте. Для них соотношение теория/практика где-то 50/50. Для обычных прикладников, к которым в реальности я сейчас отношусь, это соотношение 20/80. У меня хорошо прокачана теория - если не учитывать областей линукса и сетей, новых для меня, то теория у меня на уровне миддла. Однако по практике я на уровне джуниора - скорость, самостоятельность, уверенность в себе и т.д. Кроме неправильного мнения о себе, ещё одним фактором была неуверенность в себе.

Мне нужно хорошенько налечь на практику после того как кончится апатия. Так чтобы на 8 часов написания кода приходилось 2 чтения киндла.

3) Я хочу где-то в 33 быть уже тимлидом в яндексе в чём-то наукоёмком и уйти в менеджеры.

4) Когда кончится апатия, я приду в себя, раскатаюсь и найду нормальную работу. Тогда мне уже не нужны будут длинные тягомотные планы, которые вряд ли когда-то сбудутся и нужны только для того чтобы унять мою неуверенность. А до тех пор - работа и игры.

you know my name (2014-05-15 4:59)

Я заиклился на поиске названия для сайта.

Цель сайта - площадка для совместной разработки и тестирования AI в игровой форме на разных языках.

Варианты:

hive + mind, intellect, sense, mindx

ai + place, spot, camp, ground

ai + contest, endeavor, stream, flow, current, intent, craft

mind + craft

dendrii

trie

Я понимаю, что этот проект - просто моя игрушка и мне нужно много труда и воли чтобы его реализовать и мне очень повезёт, если его хоть кто-то оценит. Но хочу сделать всё по-настоящему, иначе не интересно. Поэтому доменная зона - .com.

Лучше варианты пока - aiground, dendrii, mindtrie, hivemindx - .com.

После долгих и мучительных копаний, я решил остановиться на одной из вариаций изначального hivemind.com. Надо найти крутую вариацию, которая не выглядит убого.

Наверное, я за сегодня ещё 100500 раз передумаю, но я нашёл хорошую вариацию. Выбор у меня между отсутствием вариаций и убогим доменом и наличием вариаций и крутым доменом. Причём, чем круче домен, тем сложнее найти хорошую вариацию.

Я нашёл +3 префикс красивую вариацию в зоне .com.

devhivemind.com

Возможно, я смогу купить домен в зоне .dev.

hivemind.dev - звучит круто!

yandex algorithm 14 warmup (2014-05-16 13:22)

Не знаю, в чём дело - в апатичных эффектах или в отсутствии интереса, но задачки разогревочного раунда совсем никакого интереса не вызвали.

Может быть, просто мозг настолько тормозит, что подсознательно нет никакой надежды. Тогда, конечно, пытаться не хочется.

Job & Work (2014-05-16 6:53)

На job-е нынче очередной баг. Проблема с наложением кастомизаций. Вроде бы нашёл место в коде, где накладываются кастомизации. Буду расследовать до конца дня в расслабленном режиме.

Сегодня яндекс.алгоритм разогревочный раунд. Было бы круто поучаствовать - вспомнить, что такое программирование.

Зарегистрировал домен - hivelord.com. Не идеально то, что я хотел (hivemind.com), но все равно - близко, звучно, интересно, передаёт идею. Здорово.

Хочу уже переехать на ubuntu сегодня - если будет позволять самочувствие. Жалко яндекс.браузера. И eclipse было бы здорово заменить на IDE от JetBrains. Чудеса в будущем ждут вообще.

Ubuntu Crash (2014-05-20 15:34)

Удалилась home под Ubuntu внезапно.

Firmware самый новый, SSD ок. Проблема на стыки ОСи и диска с учётом его продвинутости.

Настроил обновление ОСи и бэкапы ежедневно на NTFS с Win7. Буду верить, что в Canonical починят.

Kindle DX и теория (2014-05-23 7:41)

Я пересматриваю своё отношение к чтению теории.

Я придумал только одно правило - надо очень хорошо понимать, что и зачем хочешь. И если чтение поможет лучше чем сразу _практика - читать.

Сейчас я хочу понять три вещи:

- 1) как устроен компьютер
- 2) как устроены сети
- 3) как устроены ОСи вообще и линукс конкретно

Знать эти вещи мне действительно необходимо. На практике их действительно не просечь без теоретического фундамента, тут он 100 % нужен. Значит буду читать. Когда? По вечерам после того, как продвинулся по проекту заметно. Теория должна помогать мне его делать и прокачиваться, а не отвлекать и не заменять.

Ajax for columns (2014-05-24 16:44)

Constraint-ы на колонку должны отправлять правильный запрос на сервер. Такой, чтобы перерисовывалась таблица, типа `layer=t1`.

Теория (2014-05-24 16:49)

+ теория полезна для беглого ознакомления с большой областью типа веб разработки. Это быстрее и удобней чем 100-200 статей Википедии. Также для методологий.

На этой неделе я хочу дочитать дейтела, прошерстить хабр на предмет веба и адаптивности и прочесть цикл статей по сетям. Посмотрим, что успею.

Для языков программирования, фреймворков, платформ теория - только документация. Если знаешь основы, то в этих сферах нужна лишь практика и Мб методологии.

(2014-05-27 10:12)

- > Меня затрахала голая рациональность.
- > Меня затрахало программирование.

Смена отношения (2014-05-27 10:28)

Как только решил попробовать пойти в менеджеры, проггерские хардкорные задачи начали напрягать. Я ненавижу эту работу - сложную и бессмысленную.

Сегодня я уже ничего с ней не буду делать. Я решил стать менеджером и я не хочу больше нервничать из-за того, что не знаю, как написать код. Мне это неинтересно. Я хожу на эту работу только по одной причине сейчас - ради денег. Задачи интереса уже не представляют. Я мог бы сменить работу, но пока что ничего непонятно. Я хочу попасть на стажировку в Яндекс, если проиграю - только тогда надо будет решать, что делать дальше. Потому что если меня возьмут на курсы, то такая работа - как раз то, что мне нужно.

Так что пока продолжаю оставаться на этой "полке". Завтра приду пораньше и со всем разберусь медленно, основательно, по порядку. Вопрос за вопросом. Сегодня я уже сломал себе голову, надо чтобы она прояснилась.

Для курсов надо собрать звезду и заботать пару глав (самые основы) матстата. Я думаю, до выходных успею.

тормоза (2014-05-28 5:24)

Апатия, потом неделя рациональности, теперь - депрессия и пробуждение эмоций. Никак не могу стабильность обрести. То в одну сторону заскок. То в другую.

Мир не чёрно-белый. Программирование - это не конец света. Но программировать в асофте - да, это задница.

Я не понимаю, что мне делать. В сердце и голове - мутные эмоции и мутные мысли. И при этом надо работать. А я мечтаю о том, чтобы больше не программировать.

1.4 August

SCO (2014-08-18 14:43)

Приятно, что с Таненбаумом начал разгоняться.

Завтра добыю 4ю главу, а дальше там полезность и знакомость материала пойдут по нарастающей. Это уже будут не умозрительные понятия, а то с чем я сталкивался. Что-то может остаться за пределами, но это будут частные случаи.

Например MIDI - формат исходников музыки для электронных музыкальных инструментов.

Virtual Memory (2014-08-26 14:33)

Вещи, которые надо знать:

1) Если ОС 32х-битная, то одна программа не может быть больше 4GB. Она просто не сможет собой управлять. По идее, пейджинг, сегментация и гибридный режим здесь не помогут, потому что все проекции кусков в любом случае в сумме не могут вылезать за адресное пространство.

2) Core i7 64-х битный в плане адресации памяти, но архитектура его - это расширенная IA-32 (32х-битная).

3) В 64х-битной ОС можно вылезать за 4ГБ насколько угодно, если

- или каждый сегмент меньше 4ГБ, а в сумме - сколько угодно (лишь бы HDD хватало)
- или есть сегменты больше 4ГБ, но используется пейджинг с сегментацией (гибридная виртуализация)
- или сегментация не используется, но включён пейджинг (перформанс наверно упадёт без сегментации)

4) У меня программа, которая искала цепочку простых чисел, падала без памяти т.к. у меня была куча over 4GB одним сегментом без пейджинга, соответственно этот сегмент не влезал в физическую память, несмотря на x64 адресное пространство.

5) В мейле жаловались, что клиент игры выходит за 8ГБ. Они определённо всё знали про разные режимы виртуализации, которые можно указать компилятору, так что напрягали их две вещи: 32х-битный клиент (пользователи на старых машинах и/или ОСях), перформанс хит из-за своппинга и/или пейджинга.

6) Старые процессоры (e.g. IA-32) не поддерживали адреса вне своих 32х бит.

Соответственно, это было узким местом - на 32х битный проц не поставишь 64х-битную ось. Железо просто засбоило бы случайно при первой же инструкции, рассчитанной на 64 бита. Если бы каким-то чудом x64 ось вообще можно было установить на IA-32 проц.

Потом появилось x86-64 расширение, на которые уже можно было ставить - и затем стали появляться x64 оси.

Yarly IM (2014-08-26 5:18)

Я не понимаю, делать ли мне проект кросс-платформенным. Это был бы плюс, но не оверхед ли? Если соединения с БД, сокетами, файловой системой и ГУИ будут спрятаны за Qt фреймворком, то что мне останется?

Однако, Гуи я точно хочу делать на Qt.

Что касается БД, сети, файловой системы - то тут надо смотреть на варианты, каждое решение индивидуально. С одной стороны, я могу использовать qt или boost - это будет быстро, но возможно я многое упущу. С другой стороны, я могу закопаться до уровня самых непосредственных вызовов, потратить тонну времени и получить непереносимый опыт.

Каждый выбор софта индивидуален, но суть в том, что версия 1.0 - не кроссплатформенная, а линуксовая. Это главный императив. В остальном - по ощущениям.

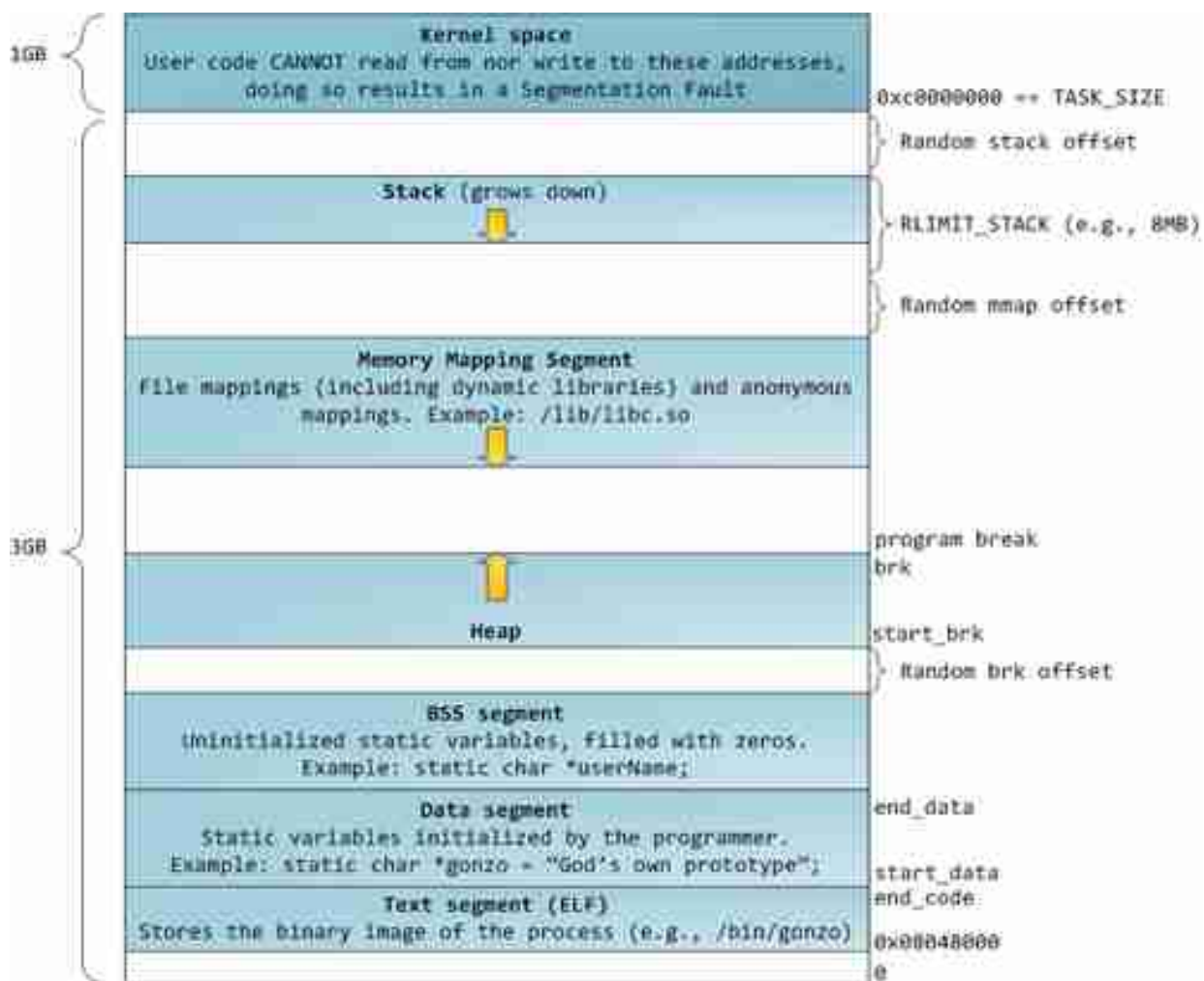
Чёрт возьми, мой гиперконтроль, моя гиперрациональность что-то оживились. Наверное, пока не вернётся искра, я буду продолжать бояться, что всё развалится без рации.

Ашла, я храню терпение и веру.

Segments, C++ (2014-08-27 6:05)

Basically, every program gets 3 spaces in the computer's memory. One is the data segment, the other one is the stack segment, and the third (but not least) is the heap. In these three areas you will basically find the program's variables.

<http://www.dreamincode.net/forums/topic/197449-the-heap-segment-in-c/>



<http://duartes.org/gustavo/blog/post/anatomy-of-a-program-in-memory/>

<http://duartes.org/gustavo/blog/post/how-the-kernel-manages-your-memory/>

1.5 September

Yarly (2014-09-04 16:05)

Сделать нетривиальный IM - вряд ли просто. Но разобраться и сделать хороший продукт - здорово. Конечно, это не замена настоящей работе. Но я хочу, чтобы я не просто мог сказать, что сделал чат из упражнения.

Я хочу сказать, что я сделал нечто своё - небольшое, но целиком моё и прекрасное.

И сделав это, я возьму с собой это чувство уверенности в индустрию.

‘ — — — ’

(oYo) —- { yarly! }

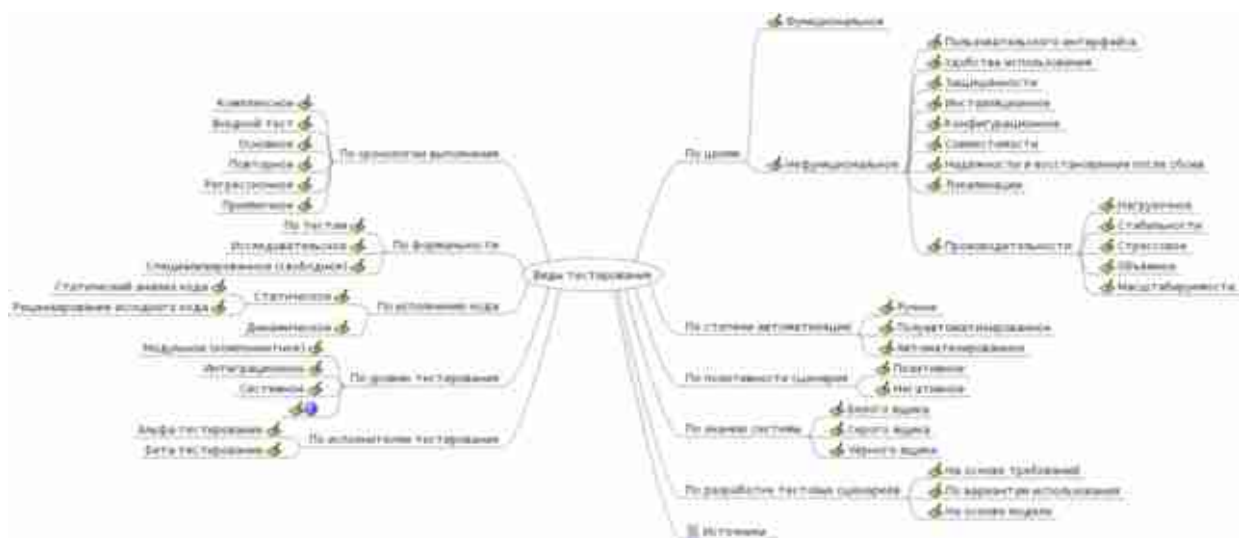
$$\Lambda_{,,,})$$

[“X”]

Скорее дочитать-восстановитк-Q Это будет круто! Хочу довести до хорошего состояния и чтобы не пропало.

1.6 October

(2014-10-18 3:16)



Виды тестирования

Ресурсы (2014-10-22 13:23)

Мои профессиональные ресурсы это:

- 1) Книги, статьи, видеолекции
- 2) Конспекты
- 3) Блог
- 4) Читы в записной книжке (OneNote)
- 5) Сиюминутное - Гугл, Википедия

Цели (2014-10-22 13:59)

Я хочу стать более практичным, прагматичным и зрелым как программист.

Хочу стать мастером и сформировать правильные привычки.

Хочу добиться правильного баланса практики и теории.

Всё это сводится к одному: хочу обрести уверенность в себе как программист. Для этого нужен опыт, много опыта - поражений и побед, чтобы научиться побеждать постоянно.

И нужен опыт своего проекта, так что я помню про yarly.

Обрести мастерство.

Practice makes perfect.

Хватит уже бояться. Хуже этого страха ничего и нет. Хотя у меня были причины откладывать практику, теперь их уже нет.

Вперёд и вверх!

Принципы (2014-10-22 13:59)

Ни дня без строчки?

Разделяй и властвуй (2014-10-22 14:44)

Завтра я начну упражнения по C++.

Я попробую практику "разделяй и властвуй": сначала проектирование на бумаге и только затем, когда собрана вся информация и решены все задачи - кодирование. Так я отделию решение высокоуровневых вопросов от решения низкоуровневых.

План, эскиз, рисунок.

ТЗ, прототип, код.

Tech stack for a web service with python and C++ (2014-10-31 15:54)

I'm designing a simple web service to get a hands-on xp of web programming. It will be a RESTful web service for plain-text quotes with CRUD capabilities for them (create-read-update-delete).

The imperative is that the project is in C++ and python, platform is Linux and I'd prefer to interface between the layers smoothly.

For now, I'm sorting through technologies to build upon. Fo now, I got the following stack:

1. HTTP (CRUD via POST-GET-PUT-DELETE)
2. Apache web server
3. mod_wsgi for python
4. python (+ Django?) to parse requests (FRONTEND)
5. Boost.Python to call C++
6. C++ for business logic (there is little of it, yet I need to know how to interface to C++ for later use)
7. ODB storage adapter
8. mysql for storage

I presume, such stack is overkill for this simple goal, I could have just written everything in C++ via Boost.ASIO and store in DB - **please tell me the alternatives**.

Nevertheless, I like this stack, for this web service is just a prototype to know the tech landscape and I think it serves the purpose.

Вопросы проектирования (2014-10-31 16:22)

1) Нужен ли Django?

Да, нужен, чтобы научиться хоть что-то им делать - подключать его и использовать. Не бояться - на будущее.

2) Написать свой веб сервер?

Да, но сначала настроить всё для работы через Apache: пусть мой сервис умеет работать и через Apache, и со своим встроенным простым сервером - MoinMoin так спроектирован.

3) Читать книги и статьи? Нет, хочу сделать своё, велосипедам - быть.

4) Как словить поменьше граблей? Тutorials, документация.

5) Что из этого всего получится? Получится веб сервис Quotr со сложной архитектурой и множеством возможностей, что позволит мне подготовиться к yarly.

6) Сколько это займёт времени? Столько сколько нужно, но не больше. Я думаю, можно уложиться в месяц, но могу ошибаться. Сложного ничего не вижу, если бы не подводные камни и деплой, если бы всё уже знать - хватило бы недели, возможно.

7) Это правда нужно? Да, Ярли сразу если делать - задача получается слишком размытой. От прототипирования все равно никуда не деться, а тут хотя бы получится осмысленный прототип.

8) Что будет с Quotr потом? Пока не знаю. Возможно, напишу статью на хабр - если чему-то научусь. Возможно, сделаю для него перманентный деплой - я не хочу чтобы проект прочто ушёл в небытие. Возможно, его можно будет использовать в Ярли - хотя и не без недоработок. В любом случае, мне бы пригодился такой сервис!

lheartquote не предоставляет необходимого мне интерфейса.

9) Будет ли API restful? Пока не знаю. Сначала надо научиться брать GET запросом одну единственную цитату типа

<http://192.168.1.4:8080/quotes/1>

Дальше посмотрим.

1.7 November

Баловство (2014-11-01 12:15)

Выбор домена:

- quotr.io,
sniptr.io - модная доменная зона, 100 \$ в год
- sniptr.org - немодная доменная зона, 20 \$ в год
- sniptr.co - компромис, 40 \$ в год

Профессионализм (2014-11-01 15:37)

Я не уважаю программистов, которые усложняют технические детали для пользователей (будь они программисты или юзеры).

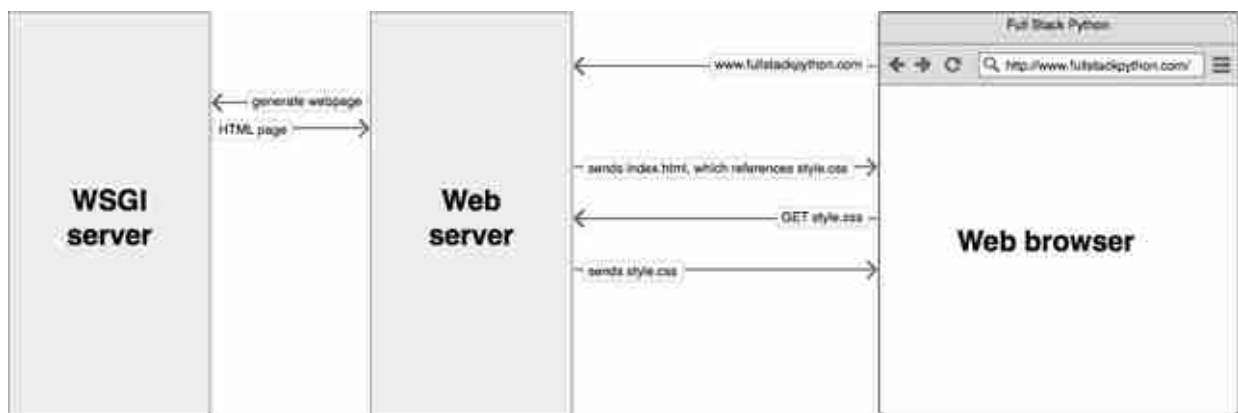
Я презираю тех, кто делает это ради эго.

(2014-11-02 12:20)

> Главное для меня в деле - на работе или вне - это вызов. Остальное вторично.

- Я

(2014-11-02 12:21)



from fullstackpython.com :: mod_wsgi

yoda frontend deploy part 1 - wsgi.djangoserver (2014-11-02 15:16)

Мне удалось настроить комбинацию Apache+mod_wsgi. Я пока не понимаю, установлен ли django - если да - то как, и если нет - то как.

Я плохо понимаю, что конкретно я установил - только ли простое wsgi приложение? Видимо, да.

Traceback:

1. [1]tut - Parts 1-3
2. [2]site config name fix - не "wsgi", но "wsgi.conf"
3. [3]apache2.4 IP scheme fix - /etc/apache2/ports.conf и * в /etc/apache2/sites-available
4. [4]grant folder permissions fix
5. [5]upgrading apache access control to 2.4 - /etc/apache2/ports.conf и * в /etc/apache2/sites-available

1. tut
2. <http://serverfault.com/questions/528502/a2ensite-error-site-not-exist-when-creating-virtualhost-apache2-on-ubuntu>
3. <http://stackoverflow.com/questions/14425436/cannot-access-to-web-server-via-local-ip-address-on-ubuntu-12-10>
4. <http://stackoverflow.com/questions/18948996/403-forbidden-you-dont-have-permission-to-access-folder-name-on-this-server>
5. <http://stackoverflow.com/questions/18392741/apache2-ah01630-client-denied-by-server-configuration>

Домен (2014-11-02 15:35)

Приятная всё-таки тема его выбирать, главное не переборщить.

sniptr.io - мне конечно нравится гораздо больше чем sniptr.co

snipt.io - гораздо проще для глаза

sniptr.io - тоже неплохой вариант, хотя длинный

+ django (2014-11-03 10:50)

Похоже, всё-таки надо использовать `sudo service apache2 restart`, а не `reload`.

Хотя, возможно, это совпадение и я просто зря выпендривался с именами сайта без "расширения".

Веб Фреймворки для python (2014-11-03 14:17)

Их много. С django всё как-то слишком сложно, негибко. Он насоздавал тонну всякой фигни за меня, нифига не объяснил, документация обширная, но и убогая.

django, возможно, я использую для построения страниц сайта.

pylons тоже похож на полноценный mvc фреймворк.

Но для веб сервиса мне нужно что-то более простое и гибкое.

web.py похож на то, что мне нужно сейчас. Его использует Яндекс. Для меня это уже отличная рекомендация.

cherrypy мне меньше нравится - что-то несерьёзное и этим отталкивающее в нём есть.

flask - что-то хипстерское.

bottle - похож на кустарщину.

Вопрос выбора фреймворка непросто.

Всё затрудняется тем, что я не до конца понимаю, что происходит. Прежде всего, нужно осознать это - задать простые вопросы и устаканить в голове картину происходящего.

А утром всё сломалось... (2014-11-03 3:33)

Утром я не смог зайти на `wsgi.djangoserver`.

Проблем могло быть несколько:

- 1) сайт назывался `wsgi`, что не круто, учитывая что так называется модуль
- 2) я криво рестартовал апач - надо `sudo service apache2 restart`
- 3) кеш браузера надо чистить и данные `hosted apps`
- 4) рестарт и добавление/удаление сайтов делается через `sudo`
- 5) переназвал сайт в `yoda.io`
- 6) переназначил домен с `192.168.1.3` на `127.0.0.1` в `/etc/hosts`

В целом, я пока не ощущаю полного контроля за конфигами.

Однако, сайт снова в строю.

Клиент yoda (2014-11-03 4:45)

Пока что он будет чисто на `bash` скриптах. Может быть, так ему и следует остаться.

Сейчас

```
quot get yoda.io
```

Показывает мне вывод `app.wsgi`.

Моя задача на сегодня - настроить Django и поразбираться с бекэндом. Хорошо бы - начать его настраивать.

Клиент будет меняться, пока что хватит и такого псевдонима.

Домен (2014-11-03 4:50)

Дизайн, деплой и программирование сайта сделаю сам. Для галочки и для экономии.

Домен:

quotr.io

- или -

snipt.io

Совсем круто было бы сделать два примера - один под сниппеты, другой под цитаты. Но пока что отталкивает то, что это дорого. И гемора в поддержке будет вдвое больше.

Великие умы мыслят одинаково (2014-11-03 6:36)

Нашёл сайт sniprr.io, который делает то же, что я хочу сделать - но по-другому.

Делать сайт с доменом, который отличается всего на одну букву - не круто.

Так что - quotr.io

Идея простая. А даже сложные идеи уже всеми исследованы.

Неисследовано только то, что очень ново, очень необычно или очень сложно. Крайности. А я не хочу сейчас их исследовать - я сделаю простое, но своё.

И - пусть будут цитаты. В конце концов, цитаты мне нравятся, а сниппеты - так, побочный продукт.

Неуникальность? (2014-11-03 7:27)

Разве две вещи, которые выполняют одну и ту же функцию - идентичны всегда?

Нет. Я мог бы купить другие ботинки, втрое дешевле, но я выбрал тимбсы. Я мог бы выбрать любой дистрибутив линукса, но я (после долгих раздумий) выбрал ubuntu. Я мог бы выбрать любое кофе, но мне нравится Lavazza.

Это вопрос не только функции, но и удобства и вкуса.

Я сделаю не pastebin. Я сделаю хороший сайт с цитатами.

Насчет того, какие ресурсы я готов вложить, насчет модели его платности, размера ресурсов на каждого пользователя - с этим я определяюсь позже.

Но - я хочу сделать нечто, что будет жить. Продукт, а не пример. И я готов оплатить хороший домен и хороший хостинг.

quotr.io будет не демо-стендом и не площадкой для скачивания. Это будет маленькая платформа - для хранения цитат. К чёрту хранение сниппетов, есть ideone и многое другое для этого.

quotr.io - будет the right thing для хранения цитат.

Первый рабочий день в состоянии оцепенения (2014-11-05 10:54)

Прошёл не так уж плохо.

Я вспомнил контекст.

Я подобрался близко к решению задачи.

Я понял, что надо руководствоваться своими стандартами того, что правильно и неправильно. Я - свой самый строгий и самый честный руководитель.

Завтра продолжу с того, на чём сегодня остановился - сделаю, чтобы URL stack был виден в логге. Дальше надо понять, как обработка \$LINK отличается от обработки \$LINK2. Это должно определить способ правки бага.

Я понимаю, что для программиста важнее всего полностью проанализировать ситуацию. Я мог принимать решения в слепую, но это противоречит моим стандартам профессионализма.

Выгонят меня или нет - мне неважно. Главное - быть верным своим стандартам. Это опора внутри меня. Чем больше таких опор в себе я найду, тем легче мне будет двигаться по жизни.

Сейчас я внутренне очень целостен, очень силён, но совершенно не адаптирован. Я подобен акуле на суше.

Пора отрастить ноги и научиться дышать!

Адаптация = эволюция.

make clean (2014-11-05 10:57)

Как я и предполагал, проблема не включения моего вывода в лог была в том, что заел билд.

make clean всё исправил.

Завтра буду разбираться с SessionRepo::pushNode.

Джобс (2014-11-05 11:29)

Теперь я отношусь к работе в асофт всерьёз.

Я не намерен больше косячить второпях.

Да, я буду работать медленно: оцепенение, сложный проект.

Но я буду работать хорошо. С минимумом ошибок. А значит - с максимально полным анализом задач.

Лог (2014-11-05 6:29)

При многопоточной обработке, Stellart в логе сервера пишет всё в перемешку.

Это очень путает, однако теперь я это знаю и разберусь.

Я понял, при каком запросе генерируется страница с лишними элементами в пути (после нажатия на F5).

Дальше разберусь.

Отношение к работе (2014-11-05 6:31)

Я работаю не за страх, а на совесть. Точка.

Домен (2014-11-05 7:15)

quotr.io - что-то хипстерское, нет?

quotrx.com - может, лучше? дешевле точно.

sid (2014-11-05 8:22)

У нас sid хранится как cookie.

Есть некая инфра хранения данных о сессии - PSS, SessionSerializer - эти детали пока неважны. Важно, что некоторым образом NetAddress привязывается к sid и хранится затем как куки.

dev (2014-11-05 9:01)

В программировании, в работе компьютера - нет ничего случайного.

Это требует дисциплины, но это и даёт определённую: на любой вопрос есть ответ и его можно получить обычно довольно просто.

Сломанное обновление по F5 (2014-11-05 9:13)

Мне нужно разобраться, чем отличается ссылка

/wnd/CRM/Event _DiaryList

от ссылки

/wnd/CRM/Event _DiaryList/StandardInput/, _this=14, _nocache

в смысле действия на session _repo->path

* _nocache не реляет: он влияет на параметры сессии, которые, как я вижу всегда пусты, мёртвая ветка кода или задействована иначе

* _this - не смотрел

* StanardInput - реляет

Языки Программирования (2014-11-06 17:04)

Мне нужно его знать и сделать что-то на нём было бы очень и очень победой.

Но я не думаю, что "бэкэнд умирает".

Я сомневаюсь, чтобы яндекс-гугл-спутник собирались переводить серверную часть на JS. Это как-то... что-то не то.

Я хочу расширить своё портфолио языков - к скудному C/C++ (и C # где-то в прошлом) добавить:

- * python - отличный язык для простых серверных задач

- * JavaScript - нужен для фронтэнда

- * Java - нужен для простых Enterprise приложений, не требующих чудес производительности

C C #-ом я не хочу связываться. Он закрытый и имплементит microsoft way java.

Мне интересны такие штуки как Haskell, Go, R, наследники lisp - но это уже сильно потом или же в свободное время.

Прежде чем считать себя полиглотом, мне нужно прочитав эту книжку про языки, где лямбда на обложке.

Программы людям (2014-11-07 16:46)

Программы должны отвечать простым потребностям, которые должны быть сформулированы коротко, ясно и человеческим языком.

yarly (2014-11-07 5:37)

Я думаю, что Yarly всё же стоит сделать, но только тогда, когда я уже буду на хорошей работе, никуда не торопиться, с хорошим представлением о себе.

А когда тогда ШАД? Математика? Всё это?

Я думаю так: не стоит забывать про эту вершину, но планировать детально что-либо - ещё рано. У меня пока другая вершина в голове.

А там - я могу пойти на CS или анализ данных или отдельно - на биоинформатику. ВШЭ - неинтересна, слишком сложно. И потом - я/ всегда хотел в ШАД.

Что касается того, когда туда идти... Ясно, что тогда, когда я состоюсь как программист. Это значит хотя бы пару лет поработать хорошо и дорасти до старшего разработчика. В эту пару лет не стоит отвлекаться на математику. Я хочу стать крутым бекэндером. И у меня как раз будет возможность параллельно с работой постепенно что-то делать. Возможно, в команде.

Сейчас у меня есть другая вершина - вырасти из асофта. И у меня есть достойный проект - Yoda + quotrx.com. В них ровно столько сложности, сколько нужно, чтобы на практике осознать всё про сервисы и про бекэнд веб разработки.

#9878 (2014-11-07 8:48)

* я получаю запрос

как только я распарсил аргументы и знаю, что это запрос к entrypt, мне нужно понять, первый ли он для этого я смотрю на процедуру логина и ищу, какой кусок данных можно использовать как флаг первого посещения

дальше я создаю ветвление в зависимости от флага первого посещения

- если флаг=да, то ничего не делаю

- если флаг=нет, то нужно редиректить, тогда

мне нужно убедиться в том, что с последним запросом в путь не попало мусора

дальше - взять из пути последнюю ноду

взять у неё url

найти готовый метод редиректа на url

срединоктить

static IP (2014-11-08 8:49)

Пока что мне нет смысла ставить адрес типа 192.168.1.* в /etc/hosts для резолва quotrx.com, потому что адреса в моей домашней сабнете динамически меняются - роутер выдаёт их, оставляя только свой адрес стабильным.

Мне нужно будет разобраться, как присваивать статические адреса - я нашёл статьи. Это первое, что нужно сделать по проекту. Уметь проверить сервис со смартфона, когда я дома на вайфае - это очень полезно.

Сервис, не сайт - потому что он чисто динамически даёт контент и не имеет практически никакого GUI.

quotrx+yoda: baby steps (2014-11-10 13:04)

1) сделал чтобы вывод сервиса было видно в смартфоне - достаточно было поменять time при том же тексте. зря ругал IE - opera вон даже с html не хочет показывать.

2) сделал ещё один отдельный питоновский файл как адаптер для импорта плюсового кода.

F5 (2014-11-10 14:08)

Завтра на свежую голову разберусь с тем, как подгрузить отдельно фрейм. Я уверен, что дело в том, что при нажатии на кнопку обновления запрос имеет отправной точкой фрейм и браузер, зная это, перегружает только фрейм. Как это сделать программно, на стороне сервера - вот вопрос. Или есть какой-то параметр http запроса типа referer, который за это отвечает - или надо грузить сначала главную страницу с нуля. Практика и

статьи.

Сейчас у меня флаг не завязан на сессию, поэтому всё и ломается при перезаходе. Сначала надо перенести флаг в сессию. Потом, для того, чтобы не происходило ошибки при повторном входе, надо проверить умирает ли сессия при переходе в браузер на stfc18 без выхода. Мне кажется, нет. А это значит, что в методе логина надо обнулять флаг.

python (2014-11-10 14:10)

Питон фрустрирует - всё-таки гугловских упражнений совсем недостаточно, чтобы научиться питонить, не зная при этом тыщу языков. Язык учится гораздо дольше.

Нужно прочитать книжку по питону. И ещё - до того, как идти в Спутник, надо прочитать книжку по языкам, ту где лямбда на обложке.

plain text (2014-11-10 5:40)

IE на WP8 перестал воспринимать plain text вывод моего сервиса. Это особенность IE. Он не умеет воспринимать plain text как страницу. Django применял вёрстку вокруг текста.

Web.py не лезет с вёрстками, и это хорошо.

1) попробовать Opera

2) Судя по двум статьям из web.py cookbook [1]hello и [2]serve-xml, мне достаточно изменить код так:

```
render = web.template.render('templates/', cache=False)

class hello:
def GET(self):
web.header( 'Content-type', 'text/html' )
return '<some html>Hello, world!<some html>'
```

Это вариант без использования встроенных шаблонов web.py с языком Templetor. После того, как сделаю по-своему, по-простому, могу попробовать и с ним.

Есть один момент, из-за которого может не сработать: web.py для шаблонов делает так:

```
render =
web.template.render('templates/', cache=False)
```

```
return render.response()
```

Render - это encoder/converter/wrapper для шаблонов. Мне он пока не нужен.

Если (внезапно!) так не получится, значит посмотрю в статью про [3]templetor, там есть способ совсем лайтово его использовать. Но вообще-то, нормальный веб фреймворк должен позволять отвечать по HTTP чистым html-ем без шаблонов. Пользователь не должен быть обязан следовать логике фреймворка, когда фреймворк стоит на пути у пользователя - именно фреймворк должен уступить дорогу.

3) Если web.py меня всё-таки разочарует (всё-таки версия всего лишь 0.37), перейду на CherryPy.

1. <http://webpy.org/cookbook/helloworld>
2. <http://webpy.org/cookbook/xmlfiles>
3. <http://webpy.org/docs/0.3/templetor>

Boost и Boost.Python (2014-11-11 15:39)

Boost - это просто набор библиотек. Часть из них требует компиляции, часть требует только для проверки целостности, часть - чисто заголовочные.

Скрипты, которые я использовал для установки Boost, просто компилируют то что нужно, а потом перемещают в особую папку.

Поверх этого, желаю блеснуть своей программерской крутизной, ребята из бустового сообщества сделали тонну велосипедов:

- * bjam (boost jam, форк какого-то расман-а)

- * boost.cmake (переделали cmake)

- * boost.build (оригинальный велосипед)

Я их использовать не намерен. Я понимаю Makefile-ы и я понимаю CMake и я понимаю PATH и LD_LIBRARY_PATH и .bashrc и .bash_aliases и g++ с флагами (хотя и не люблю его).

Я смог заставить python вызывать мой C++ hello world. Завтра продолжу. Начну с того, что мой сервис станет C++ powered. Пока же - сервис у меня отдельно, а C++ с Boost.Python - отдельно.

Как много проблем из-за тех, кто лагает и/или усложняет.

Сломанные пути при наличии двух экземпляров одной сессии (2014-11-11 8:29)

Мой план такой:

1) удалить или закомментировать все свои изменения - wazzup, meld

2) проверить, что проблема есть - по темт кейсам, тут есть несколько переменных - пользователь(Admin и), браузер (буду использовать хром и файрфокс), окно(2 экз браузера)/таб(2 открытых таба в браузере) - для всех комбинаций мне надо понять как дружат между собой 2 отдельных пути - path и path_ob

3) составить таблицу на основе 2

4) дальше, имея конкретную информацию, можно подойти к Михаилу и спросить, какие из этих кейсов By Design, а какие являются ошибкой

5) исправить ошибки

6) протестировать

7) закоммитить

8) вернуть правку под 9878

9) протестировать

10) коммит2

11) мёрж в 9 (?)

Management in ASoft (2014-11-12 9:04)

Ярослав жалуется на отсутствие менеджерских способностей у Михаила и я с ним согласен. Могу ли я как-то воспользоваться этой ситуацией экологично? Не нагло, не разрушительно и без лишнего риска для своих планов?

Посмотрим.

Пока я просто попытаюсь пропихнуть те два вопроса, о которых говорил Слава в свои курируемые вопросы.

Проект (2014-11-13 3:35)

У проекта сейчас две задачи на эту неделю + выхи:

1) внятно закончить деплой

2) настроить вайфай и получить при тесте приличную скорость (тесты в разное время, в т.ч. вечером)

Интриги (2014-11-13 4:26)

Возможно, я могу что-то сделать, чтобы получать более адекватные задания и работать с более адекватными людьми. Устроить революцию. Я не хочу никого подсидживать, я хочу, чтобы workflow эволюционировал.

Но это не значит, что сейчас я должен об этом думать.

asoft colleagues (2014-11-14 8:44)

Михаил пытается построить свою собственную защитную политику.

Артём ведёт себя как конформист.

Я перешёл от того, чтобы быть резонёром (тоже, вообще-то, одна из простых ролей), к роли... Какой? Выжидающего? Я просто хочу сохранять равновесие и извлечь максимум за оставшиеся 3-4-5 месяцев в асофте. Я хочу как можно меньше дискомфорта, как можно больше опыта, приятной работы и хорошую зарплату. Наверное, эта роль называется - взрослый волк-одиночка. Здоровый эгоизм?

Я постараюсь выстроить заново отношения с Михаилом - для этого нужны режим, спорт, производительность, флоу.

Я постараюсь закрепить за собой положение старшего в отношении Артёма, однако только до тех пор пока это обосновано. Если он внезапно окажется лучшим программистом чем я (как минимум, для здешних задач) - so be it.

Я думаю, что у меня есть мои пол года. (уже меньше.) Так что не стоит тратить время на рефлексии. Задачи сами себя не решат.

quotrx.com (2014-11-19 11:05)

На примере wishlistr: есть приложения, которые просто и приятно использовать, они полезны и удобны. Они делают жизнь лучше.

Написал этим шведам. Надо не забыть проверить ящик на рамблере.

Сегодня я многое узнал (2014-11-19 12:27)

1) Для python системы Ansible есть Ruby аналог Puppet, а над ним надстройка - Hierra и ещё много других.

Зачем мне такая инфа? Хм... Может быть когда-нибудь пригодиться

2) Мои проекты имеют тенденцию срastаться у меня в голове и взаимодействовать. Так, графы и featmap начали перекликаться с atomdeerp. Можно даже и quotrx к этому прикрутить. Я считаю, это здорово.

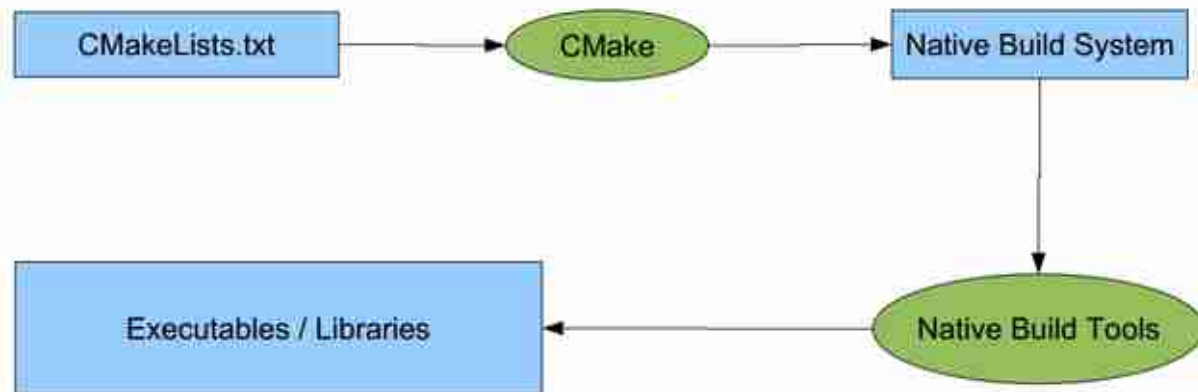
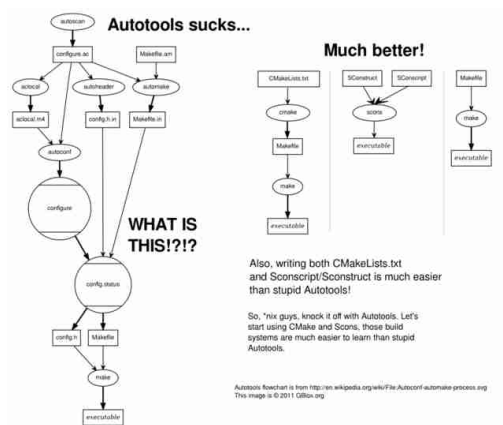
3) Я люблю организовывать. Иногда это просто способ прийти в себя, медитация через не-очень-то-полезный труд. А иногда это действительно помогает.

4) Программирование - это классно. Надо заниматься им чаще и с ясной головой. И не отвлекаться на всякую фигню.

5) Возможно, со своей уникальной перспективы я смогу сделать классный стартап сразу. А может быть, первые 9 не сработают. Но если это интересно и не надо жить впроголодь - что в этом плохого? По крайней мере 1 пользователь у этих штук будет.

6) Мне правда интересно, как можно совмещать day job со своими проектами чтобы хорошо продвигаться и там, и там. Задача для Джедая.

(2014-11-20 13:07)



CMake, make

Находки на хабре PART1 (2014-11-20 6:40)

1) tceh - серьёзный коворкинг

2) trac - похоже, хорошая трекилка на питоне

3) freemium - мне нравится больше, но он, похоже, сложнее; я бы начал с него - как пользователю мне эта модель больше нравится, я очень редко платил за софт; реально - только когда апгрейдился с бесплатной версии на платную

4) проект должен быть понятен людям, любим людьми, полезен им, ориентирован на них; т.к. пока что я делаю проект для себя, меня этому учить не надо; но эту позицию легко потерять (наверно), что очень дорого стоит

5) не в тему поста, но - есть уже не один вариант как найти ремонтников через веб - возможно, даже можно юзать Яндекс.мастера (кстати, это и есть моя идея о кооперации)

6) идеи малоценны (и так знал) и краткоживущи (понял теперь, когда увидел очередную свою идею уже реализованной)

7) мой первый проект - разведка моих возможностей и технологий и укрепление моей любви к делу; вот во втором уже можно думать, как соотнести мои интересы с рынком - причём так, чтобы не опоздать

8) фриланс

- прокачки навыков там особо нет

- много нервов

- мало платят

- много простора для прокрастинации

- мало PRO общения

- нет PRO роста

+ самоорганизация

+ бизнес-навыки

+ гибкость

+ люди

+ стартапы

9) От фриланса до предпринимателя - один шаг. Но сейчас я не готов развивать бизнес-навыки - ни на 1м этапе, ни, наверное, на 2ом.

10) Мб я и почитаю книжки про стартапы - те, что все читают. Надо же знать "тренды"

11) Оффтоп - на амазоне классно юзать вишлисты

Находки на хабре PART2 (2014-11-20 6:41)

1) Работать в чужом стартапе - грубо говоря, отстой

2) Проблема отсутствия контента и пользователей - "курицы и яйца" - реашема, добавил в перлтриз в папку yoda

3) Серьёзный стартап предполагает конкуренцию

4) Ходить на мероприятия для стартаперов - глупо, там "хипстеры"

5) В финансовом плане всё очень жёстко со стартапами и тут мне надо будет постепенно адаптироваться к стали - на этапе 2.

6) Стартапов много и 95 % не выгорают. Это марафон и надо быть готовым к 9 поражениям ради 10го проекта, на котором будет победа. Конечно, всё может оказаться милосердней, но нервов и решимости должно быть именно столько. Поэтому и нужен 2й этап.

7) Было бы классно законтчить проект на этапе 2 с Яндексом. С ними было бы здорово работать.

8) Для того, чтобы уйти с работы и делать стартап нужны деньги - но не на зарплату кому-либо и офис, а на то, чтобы сидеть в коворкинге и не голодать те пол года или год (условно), что нужны для того, чтобы всё стало срастаться.

9) Прежде чем уходить с работы, нужны деньги, прототип, связи, намётки команды и много много много воли.

10) Стартап - это слово должно вселять дрожь в слабонервных.

11) Стартап - это очень серьёзно и требует много мужества.

- 12) В среде стартаперов тонна дерьма. Нужна мощная интуиция.
- 13) Стартап-линч - классная штука для понимания стали в среде стартапов IT.
- 14) Лидер компании - не тот, кто разводит блабла, а тот кто принимает решения. Опыт из армии.
- 15) В какой-то момент было бы полезно походить посмотреть на серьёзных стартаперов.
- 16) "Руководство по созданию стартапов" - классная серия статей, проводящая через все грабли по многу раз самым болезненным способом.

hardware materials (2014-11-22 11:18)

Статья: [1]PART1 и [2]PART2

Процессор

Кремний (silicon) - основа чипа

Примеси фосфора, арсения, галлия, бора - превращают кремний в полуметалл, улучшают проводящие свойства и позволяют ему служить массивом транзисторов

Медь (copper) - соединяет транзисторы

Диоксид кремния (silicon dioxide) - изолятор между слоями медных соединений

Жёсткий диск

Алюминий или стекло + примеси - основа диска в HDD

Куча редких металлов - слои поверх основы и головка HDD

Материнская плата

FR-4 = стекловолокно + эпоксидка - основа, изолятор

медь - проводник

1. <http://www.techradar.com/news/computing/pc/the-weird-and-wonderful-materials-that-make-up-your-pc-1089510>
2. <http://www.techradar.com/news/computing/pc/the-weird-and-wonderful-materials-that-make-up-your-pc-1089510/2#articleContent>

Как связать C++ с MySQL на Ubuntu лучше всего? (2014-11-22 12:19)

Просто API и обёртки

1) MySQL::Connector/ODBC - кроссплатформенная обёртка вокруг "call level interface" (стандарта на простейшую работу с разными бд по TCP/IP), для меня API этой библиотеки только всё усложнит: кроссплатформенность мне не нужна сейчас, а если бы и была нужна, я бы использовал что-нибудь получше.

2) sqlapi++ (sqlapi.com) - коммерческая штука, фтопку. либы не должны быть коммерческими.

3) mysql C API - не хочу спускаться до Си

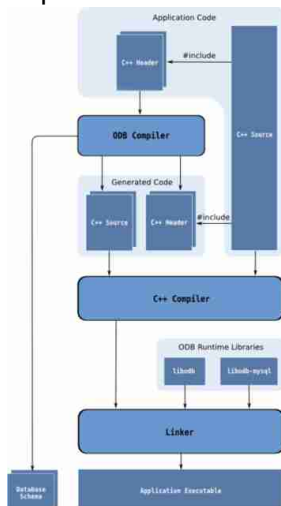
4) mysql++ - C++ обёртка над mysql C API

5) MySQL::Connector/C++ - уже лучше, обычный драйвер для работы с mysql из C++. По сути, замена mysql++ - видимо Oracle растоптала чувака, который выбежал вперёд их в пекло.

6) Остаётся SOCI как более красивая C++ обёртка mysql C API. Используется в CERN, развивалась уже лет 5, простая, есть на гитхабе, версия 3+. Однако, запросы хранятся как строки и чтобы сделать как надо, надо самому писать инфру для их хранения иным образом. Т.е. нет классов Query, Connection и т.д.

ORM-ы

1) ODB - (a) требует изучения их собственного препроцессингового псевдоязыка, (b) использования их собственного компилятора - всё это сильно усложняет процесс сборки. Это создаёт пайплайн. У меня плохие ассоциации со словом пайплайн. Я считаю, что не должно быть никаких велосипедов и костылей в моём случае. Зачем ещё один компилер? Зачем свой язык приделывать к определениям препроцессора? Какая-то маргинальная технология. Больше всего меня добивает эта схема, которую я не могу осилить:



Ад какой-то. Как gnu autotools.

2) QxORM - Qt way. фтопку.

3) Wt::Dbo - Wt way. фтопку.

4) LiteSQL - не развивается уже, не собралось. Печаль.

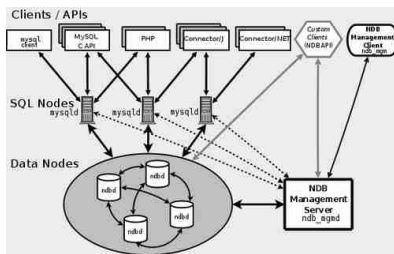
Мне странно, что вообще этот вопрос как-то слабо освещён, что я не нашёл тонны статей. Похоже, совершенно не трендовая вещь. Все либо из под винды юзают ODBC обёртки, либо удовлетворяются mysql C API или Connector/C++. Но большинство просто юзает Java или php.

Ни одного нормального ORM - 4 инвалида. И главное, я понимаю, что мотивации делать такую штуку мало. Поддерживать её ещё потом. А денег за это никто не платит. Фигово.

Так что остаётся использовать простые адаптеры и сделать свой слой абстракции. Из того, что я смотрел - Connector/C++ приличная вещь, но синтаксис SOCI куда проще. Однако, SOCI не поддерживается так активно и за ними нет яндекса (тот же web.ru яндекс может при необходимости купить).

Я буду использовать Connector/C++, поддерживаемый Oracle.

Как устроен MySQL (2014-11-22 14:09)



По сути, есть просто МНОГО разных API, которые все в итоге приходят к mysqld.

Последний элемент (2014-11-22 14:14)

Я остановился на Connector/C++ и знаю, что выбрал правильно.

Однако я плохо понимаю устройство MySQL и баз данных в целом - как они взаимодействуют с ОСью. Поэтому завтра я хочу почитать ещё книжки по MySQL и дополнить свои конспекты.

Вопросы:

1) Сколько процессов/потоков в mysql?

2) Какие?

3) Как все эти API вяжутся между собой?

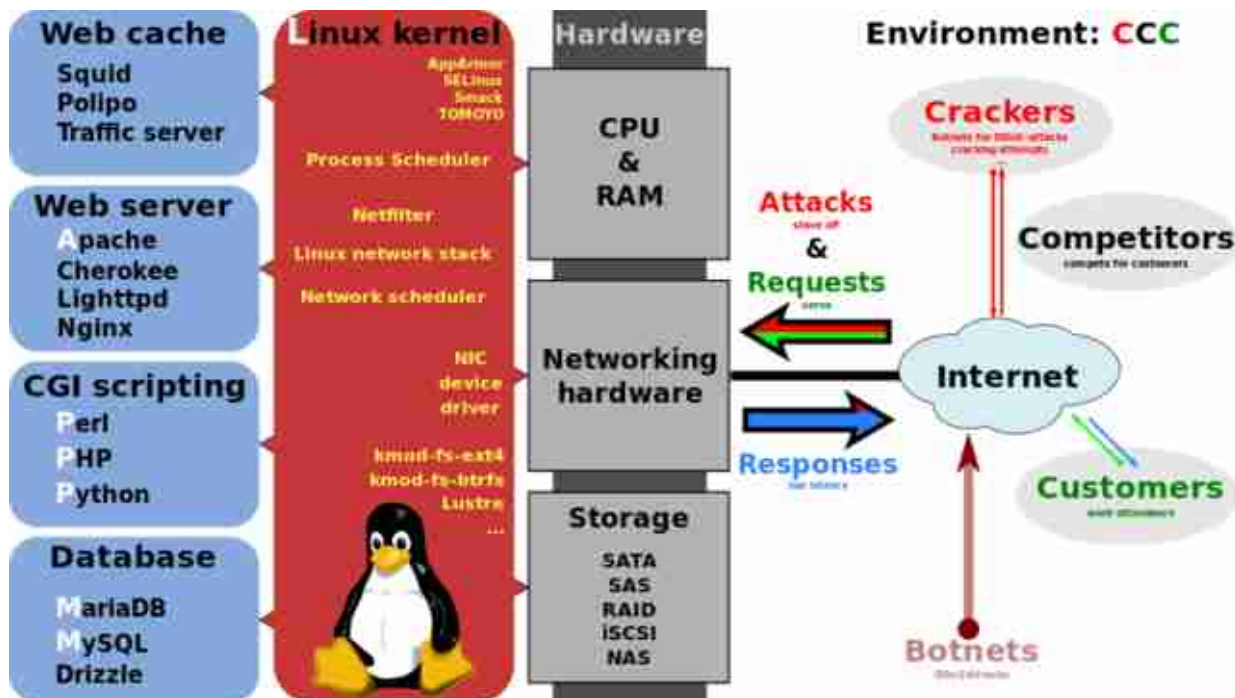
4) Как получать доступ к mysql локально и удалённо?

5) Как MySQL масштабируется?

Я не хочу контролировать, я хочу владеть темой.

У меня скачаны 3 книги на ридер. Если их не хватит, найду что-то помощнее.

(2014-11-22 6:50)



Подвиги при сборке Conn/C++ (2014-11-23 14:57)

Выяснил, что

```
g++ -o $(TARGET) -lmysqlcppconn $(TARGET).cpp
```

не работает, а

```
g++ -o $(TARGET) $(TARGET).cpp -lmysqlcppconn
```

работает.

Какой-то дурацкий синтаксис у g++.

Несокрушимый) (2014-11-24 14:14)

Я властелин моей судьбы:
Я капитан моей души.

Проблема со средой на работе - с настроением людей и корявостью задач.

Мой ответ - статья для братьев по ордену: [1]link.

Также, я сформулировал свои ответы для Антона. В результате:

(1) осознал чуть больше - а именно как дозировать и как обосновывать своё изучение проекта и составление схем. Дозировать - по текущей задаче (срочных не бывает). Обосновывать - тем, какие задачи мне удаётся решать. Если они могут их решить без меня и моих схем - им следует так и сделать.

(2) поддержал человека, который хочет выздороветь от гнили внутри. Это поступок, достойный Джедая.

Теперь у меня нет блока по отношению к работе. Я

оформлю схемы гораздо лучше, переведу в цифровой формат и положу на вики. Задачи, по которым я знаю архитектуру - буду делать с музыкой. М не легче будет приходить на работу вовремя и я буду лучше работать. Я приобрету друзей среди тех, в ком есть любовь. Когда я буду уходить из асофта, это будет уход победителя. Человека, который одержал верх над своими слабостями и готов к яркой жизни и рабочим подвигам.

1. <http://www.templeofthejediorder.org/forum/open-discussions/108853-man-vs-environment#171388>

Проекты (2014-11-24 16:24)

TOTJO может помочь мне с проектами во многом. Я пока не понял как. Как минимум, я могу предложить им использовать свой сервис - quotrx.com. Им будет приятно юзать джедайский сервис, а мне - иметь пользователей и знать, что среди них есть мои братья и сёстры.

Но это только начало. Я не думаю, что посвятить себя апгрейду totjo напрямую - хороший план. Но социальный элемент totjo может дать многое и в плане идей и в плане фидбэка.

ListResult (2014-11-24 9:50)

Я хороший программист. Структура фреймворка - это ад. Без всяких насмешек это адский адский ад для программиста. Можно сделать хуже, но тогда бы это не работало совсем.

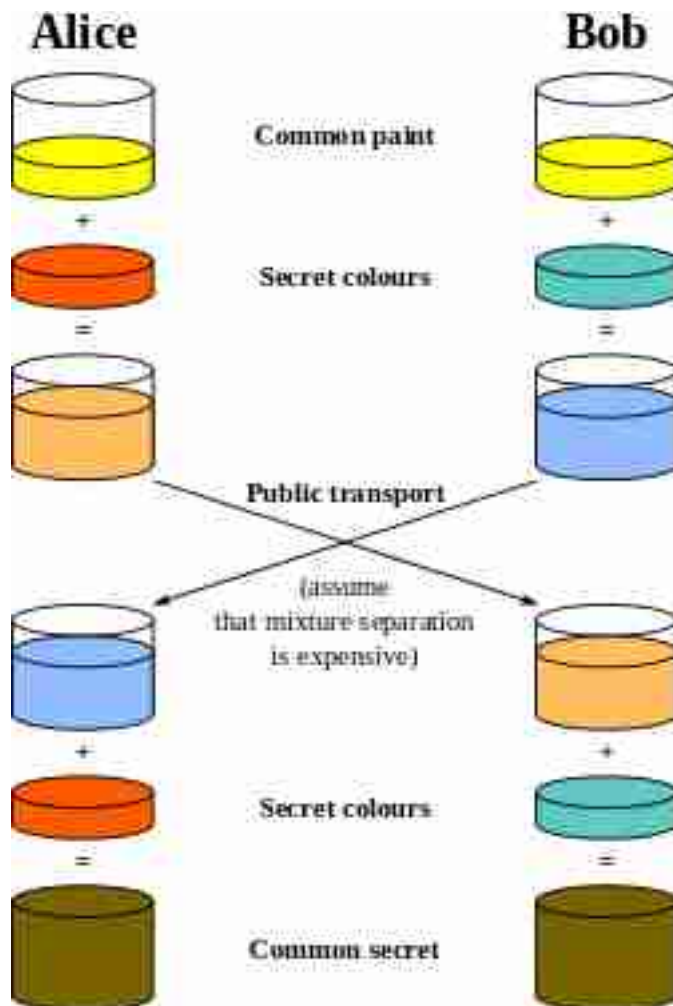
Сложность кода невероятная. Как можно так писать и при этом быть психически здоровым?

Меня просто удручает копание в этом... коде.

Я не могу его понять и поэтому не могу сдвинуться с места в расследовании текущего бага.

Я очень упорный человек. Но всё моё упорство тонет в этом болоте. И меня это убивает.

(2014-11-25 13:47)



[1]DH Key Exchange

1. http://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange

Stellart Arch Research (2014-11-26 7:30)

Всё непросто, но я продвинулся. Я осознал, как происходит пассивная коммуникация по протоколу Comm.

1) Нужно нарисовать схему про это.

2) Нужно осознать как происходит работы с LR, у меня есть точка входа: EM::Complete. Думаю, это будет проще.

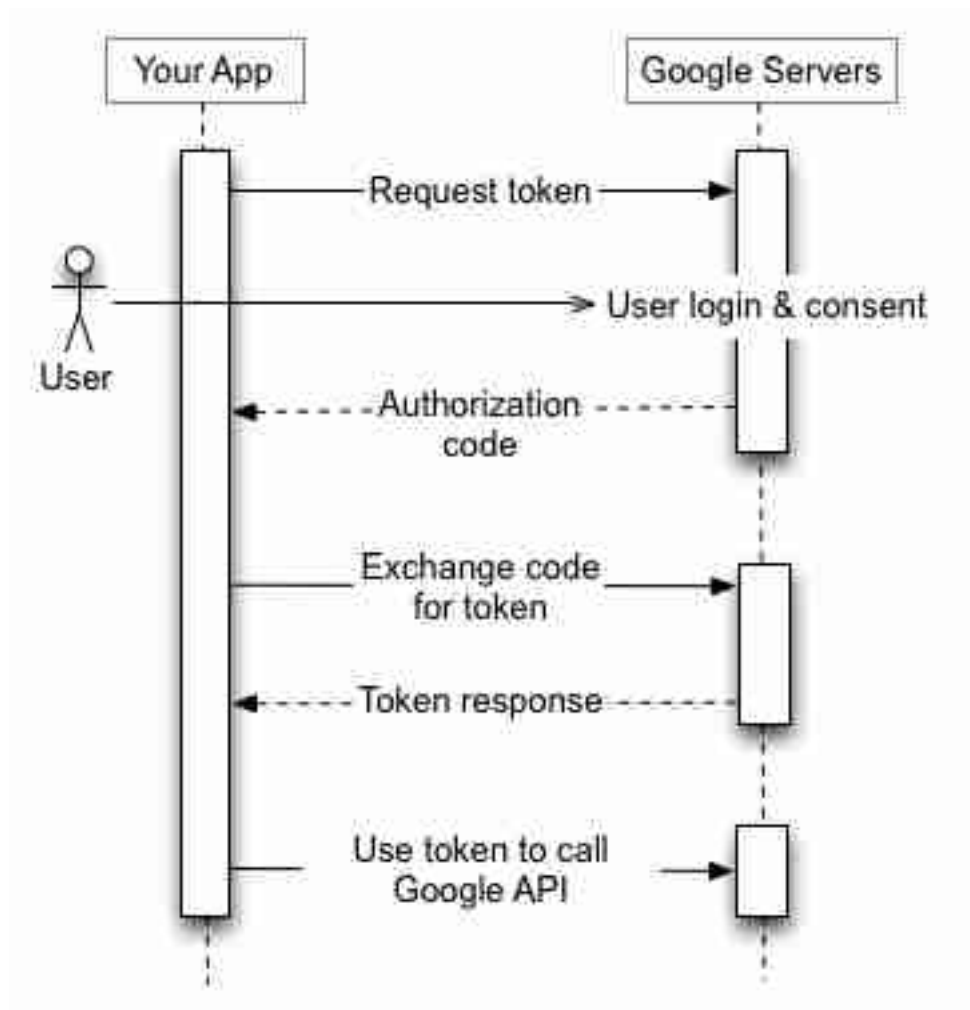
3) И потом - вторая схема

План, чтобы сделать план (2014-11-29 14:44)

Время. Его может не хватить. Завтрашний план в силе - мне нужно знание. По задачам постараюсь как можно больше отложить в дополнительные. Но так, чтобы идея не пострадала. Прогрессивный jpeg и всё такое.

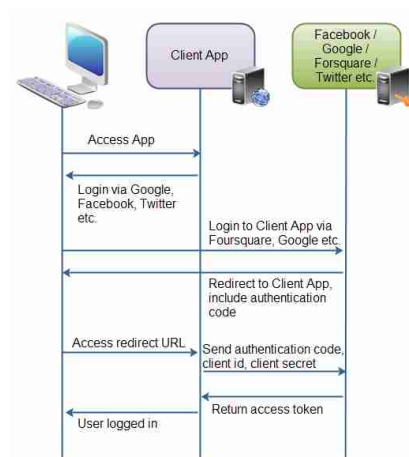
Я люблю программирование. То, что мне так страшно - признак того, что это мой икигай. Иначе мне было бы пофиг.

(2014-11-30 10:10)



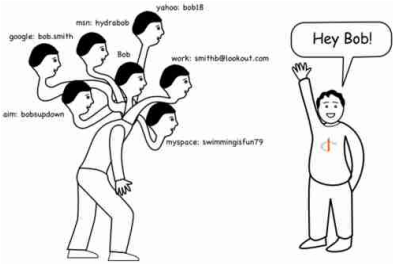
OAuth

(2014-11-30 10:43)

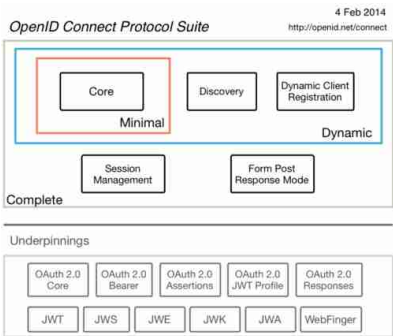


OAuth

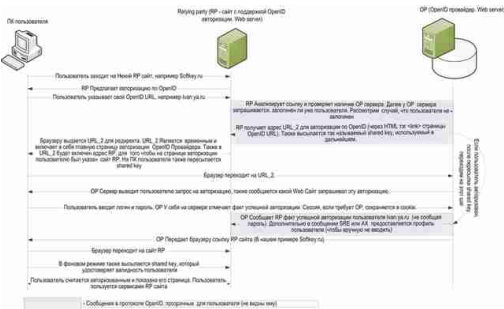
OpenID (2014-11-30 10:44)



OpenID Connect (2014-11-30 11:38)



(2014-11-30 12:23)



OpenID

The Myth of the Full-stack Developer (2014-11-30 12:56)

Link: [1]The Myth of the Full-stack Developer

1. <http://andyshora.com/full-stack-developers.html>

1.8 December

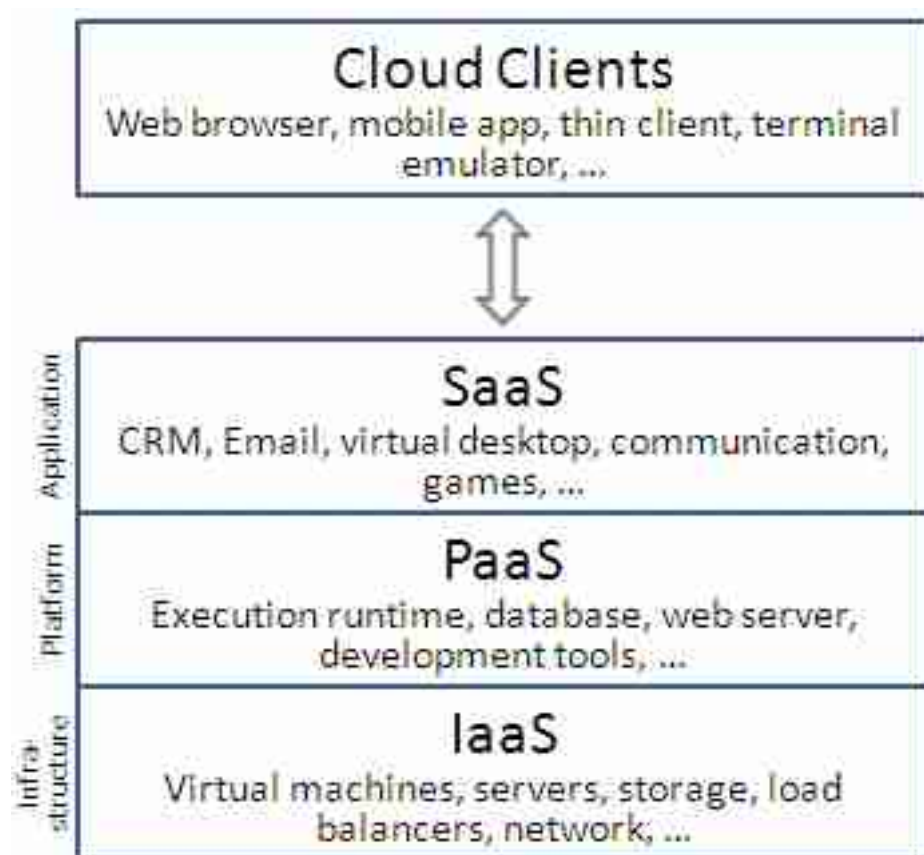
CMS для quotrx (2014-12-01 10:48)

1) Я не знаю, позволит ли мне CMS сделать то, что я хочу или нужно писать свой сайт

2) Я нашёл 3 CMS, которые мне интуитивно нравятся - Mezzanine 3.0, django-cms 3.0, wagtail 0.81. Версия имеет значение. Я буду выбирать между django-cms и Mezzanine, хотя Mezzanine, при прочих равных, нравится мне больше.

3) Сомневаюсь, что quotrx.com сможет работать без проблем на CMS, даже самой лучшей. Возможно, в какой-то момент я отброшу CMS и попробую написать сайт на django+python. Начну с Mezzanine - возможно, полностью или частично на какое-то время хватит.

PaaS vs IaaS vs VPS (2014-12-01 13:14)



PaaS/IaaS/SaaS - облако, не понастраиваешь
VPS - сервер

Мне больше нравится [виртуальный] сервер. DigitalOcean похож на подходящий.

yoda+quotrx (2014-12-01 14:02)

Проект расплзается. Моё внимание распыляется.

Нужно заново сформулировать план, очевидно. Более реальный. Более чёткий.

1) Фреймворки, документация

- django
- web.py

2) Doorway

- mapping
- registr
- auth

3) Convoy

- threads
- sessions
- cookies
- logging

4) Storage

- data
- secure
- search

5) План

- Модель

- API: CRUD, search, auth

- Схема DB

- Все задачи

- Чёткое урезание ff в extras

- Задачи в TODO по слоям

6) Книги

- Abelson/Sussman

- python
- django

Проклятье Бэббиджа (2014-12-01 15:12)

[1]Бэббидж

Чуть было не попался. Ради одобрения хабраюзеров стал раздувать проект, стремясь каждую деталь довести до совершенства.

Лучше корявый работающий проект, чем прекрасный неработающий.

Буду делать всё по минимуму. А читать про питон, джанго и языки - когда-нибудь потом. Например - после того, как проект будет задеплоен в бой и будет работать в альфе на боевом деплое.

Я не делаю "серьёзный проект". Я "играюсь", по сути. Довести проект до мощного состояния? Это требует очень много ресурсов. Хотя один проект так сделать стоит. Но я не успею это сделать до дедлайна, а после дедлайна я буду занят только работой от 2х месяцев до года.

Мне нужно принять, что проект окажется сделан в минимальном виде. Это гораздо лучше чем не сделать ничего - а фишки у меня все равно будут. Но да, это не промышленный гигантский проект. В своём видении я ошибся. Из-за страха.

Пусть у меня будет просто место, где можно хранить цитаты. И всё.

И плевать на хабр.

1. https://ru.wikipedia.org/wiki/%D0%91%D1%8D%D0%B1%D0%B1%D0%B8%D0%B4%D0%B6,_%D0%A7%D0%B0%D1%80%D0%BB%D1%8C%D0%B7

django-mysql (2014-12-02 11:35)

Link: [1]django-mysql

Писать о своих подвигах в stackoverflow - приятное занятие.

1. <http://stackoverflow.com/questions/12772687/django-error-loading-mysqldb-module-no-module-named-mysqldb/27253657#27253657>

Mezzanine (2014-12-02 5:37)

Я установил и настраиваю Mezzanine, основанный на django.

Три вещи, которые делают его (и созданный под него проект) отдельными от моего "конвейера" это:

* база данных - я нашёл как настроить persistence на свою сервер - через manage.py запускается встроенный в django wsgi скрипт

Стандартный мануал по деплою Mezz говорит:

```
python manage.py createdb --noinput
python manage.py runserver
```

Но я не намерен ему следовать, так же как и мануалам, которые предлагают мне ради одной задачи ставить virtualenv. Пока я могу обойтись без неё, я буду обходиться. Незачем усложнять и так непростую для меня конфигурацию деплоя.

Мои шаги:

- 1) я хочу прочитать про всё что управляется из manage.py
- 2) я хочу осознать возможности конфига mezz поверх django
- 3) встроенный django сервер я не буду запускать
- 4) wsgi скрипт 2й мне не нужен
- 5) мне нужно перенаправить тестовый hello вывод на url quotrx.com/test
- 6) мне нужно связать url quotrx.com с файлами в папке quotrx-mezz через импорт, чтобы после прихода на wsgi endpoint запрос по главному url уходил на quotrx-mezz
- 7) мне нужно прописать в конфиге mezz свою базу, чтобы вся дополнительная, необходимая ему инфа была мне подвластна

The Game of Fear (2014-12-02 8:55)

(RED) QUEST : TRIAL1 TRIAL2 ... TRIALX

(BLUE) TRIAL : DARE1 DARE2 ... DAREX

(GREEN) DARE : FEAT1 FEAT2 ... FEATX

* QUEST 1m, TRIAL 1w, DARE 1d, FEAT 1h

* QUEST Project, TRIAL Sprint, DARE Workday, FEAT Ticket

FEAT = RD (read) | TH (think) | DO (do smth) | SR (search)

Имя проекта (2014-12-03 13:55)

Yoda - имя сервиса

Quotrx - имя сайта

Какое же имя у всего проекта?

White Silence? Мрачно и абстрактно, хоть и можно расшифровать как web service / web site. Можно составить анаграмму. Но я хотел бы, чтобы проект назывался значимо - это же мой квест. Ignis - искра. Не "из искры разгорится пламя", но "искра огня Прометея", основа жизни Человека. Это то, что я ищу в себе в этом походе.

(2014-12-03 3:26)



Django setting (2014-12-03 8:31)

1) default settings

django/conf/global_settings.py.

2) overriding default settings

export DJANGO_SETTINGS_MODULE=mysite.settings

3) mezzanine settings

project/settings.py
project/local-settings.py

4) WSGI settings

In your live server environment, you'll need to tell your WSGI application what settings file to use. Do that with `os.environ`:

```
import os
os.environ['DJANGO_SETTINGS_MODULE'] = 'mysite.settings'
```

Memento: deploy weak points (2014-12-03 8:37)

1) Apache:

* configs:

- site (WSGI1, paths)
- main (WSGI2, paths)
- default site

* service

- `sudo apache2 service restart`

* url mapping

- IPs
- `/etc/hosts`
- urls
- root dir access problem (000-default.conf)

* log

2) FS

* symlinks

paths
rights

3) .bashrc:

- * PATH
- LD_LIBRARY_PATH
- PYTHON_IMPORT_PATH
- DJANGO_DEFAULT_SETTINGS

4) MySQL:

- * mysql.conf (somewhere)

5) django:

- * global [default]

- * my override

- mysql
- urls

- * local [mezz addon]

- mysql
- keys

HOW TO DEPLOY Guide (2014-12-04 3:47)

Я буду дополнять свой Guide по мере деплоя проекта.

Сейчас в него надо будет добавить:

- 1) Использовать github для файлов
- 2) Установку django
- 3) Установку Mezzanine со ссылкой на сайт
- 4) Правку конфигов django под mysql базу
- 5) Процесс merge-a
- 6) Разборки с PYTHONPATH и Apache

<http://felecan.com/2011/definitive-guide-installing-django-ubuntu/>

- * .pth
- .bashrc
- apache envvars
- apache process and django process users & rights

- 7) django static content
- 8) mod_wsgi -> daemon mode

Python Import (2014-12-04 8:28)

1) Hooks

- * python hook ([1]link)
 - /usr/local/python2.7/site-packages/*.pth
(needs some site infra like folders)
 - /usr/local/python2.7/dist-packages/*.pth
 - (a place for all main python configuration)
 - sitecustomize module (site should be on path already)
 - usercustomize module (site should be on path already)
- * apache hook
 - envvars file
(not everything allowed, e.g. PYTHONPATH w/ 1+ paths is not)
 - WSGIPythonPath
- * django hook

2) System

- * .bashrc
(problem: apache starts as another user)

3) Runtime

- * import statement
- * from ... import ... statement
- * from ../..name import ... statement

1. <https://docs.python.org/2/library/site.html>

django admin templates (2014-12-07 14:47)

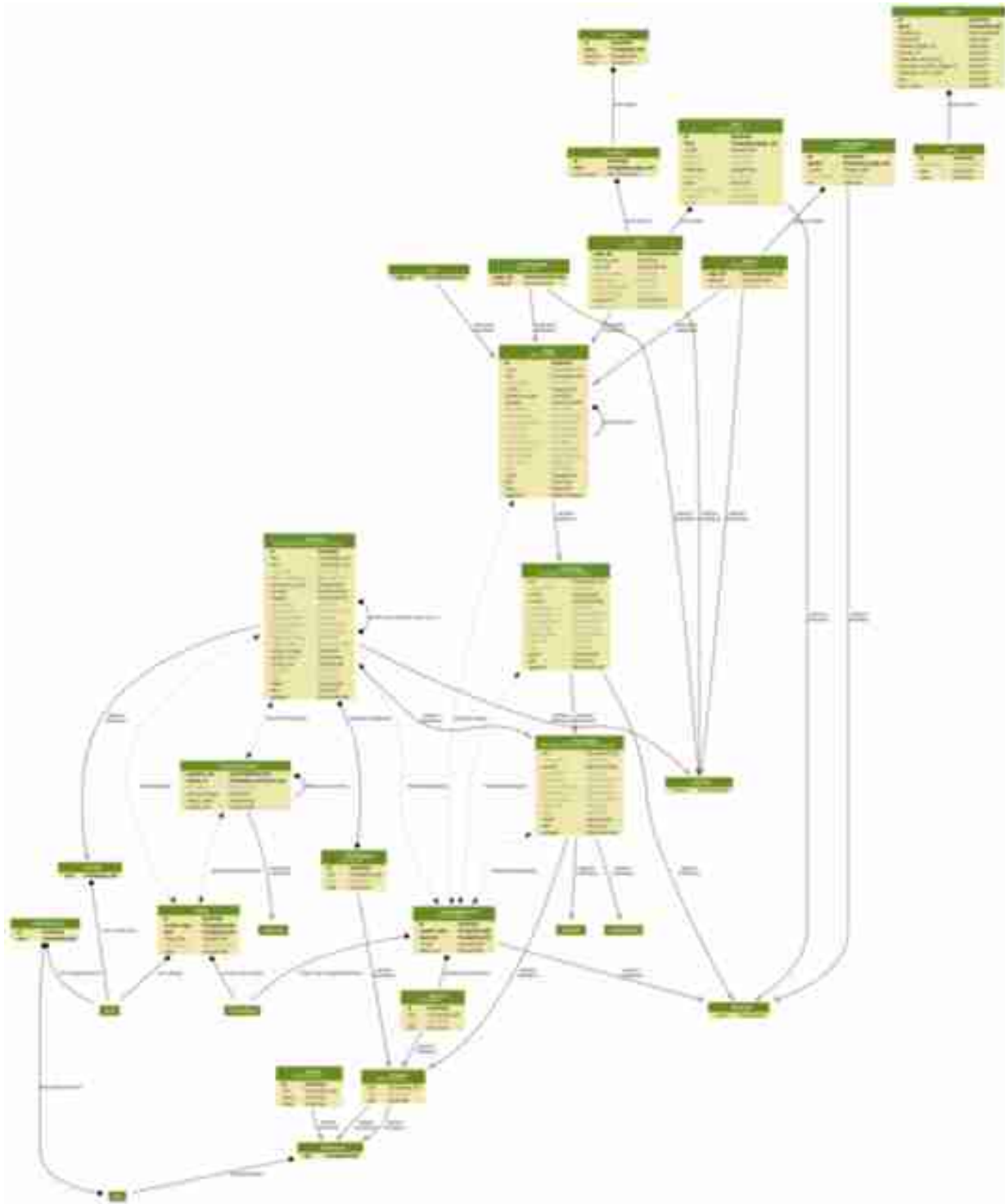
Я могу перейти на ручное управление и кастомизировать админский интерфейс, встроенный в django. Я знаю, что для этого надо повозиться в settings.py. Возможно также написать пару аккуратных строк кода. Не больше. Но стыковка деталей была бы сложна.

Пока что я не вижу в этом нужды. Меня устраивает "автомат". Меня устраивает дефолтное поведение django.

Linux (2014-12-07 15:25)

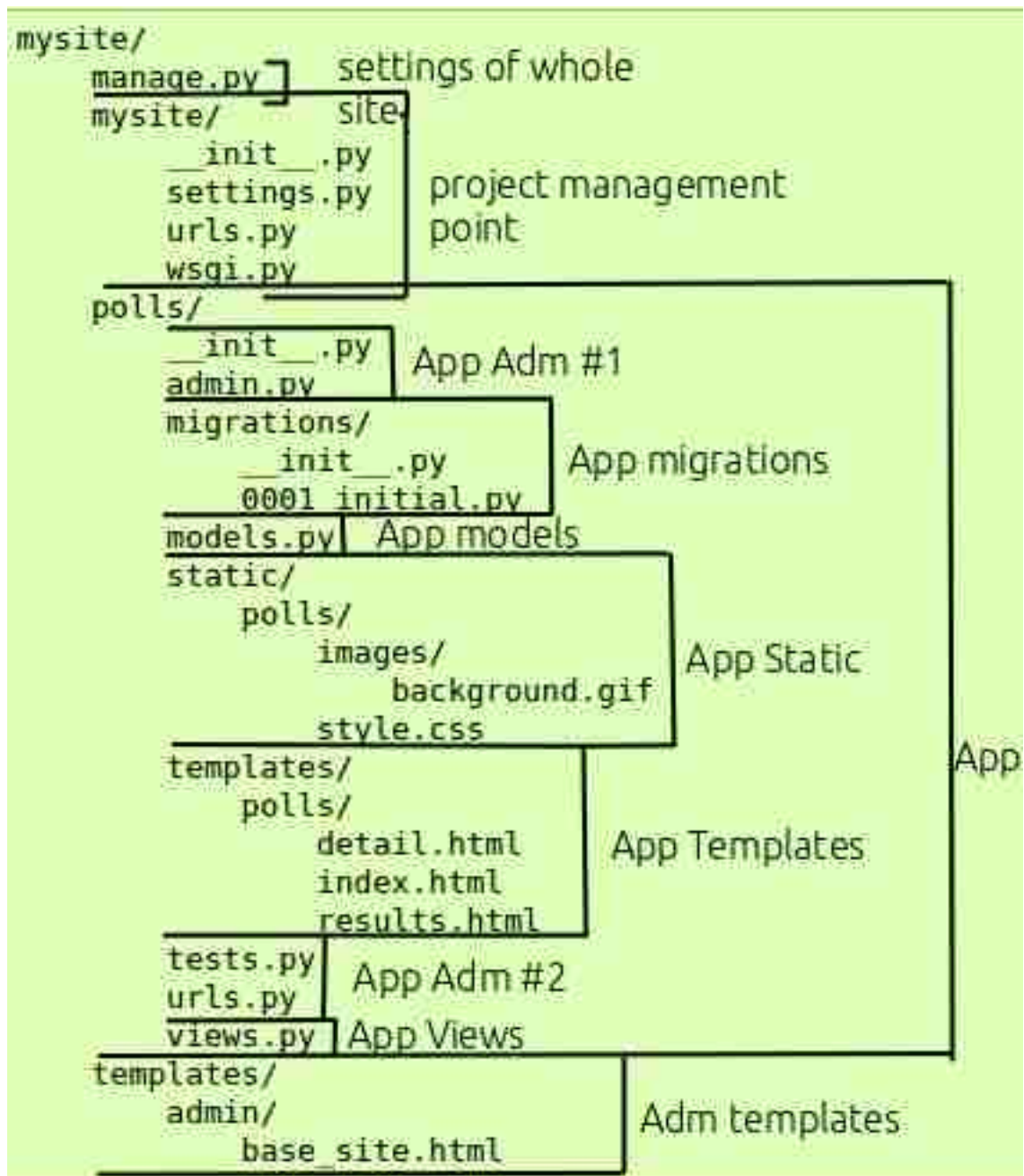
Надо уже создать шпаргалку. Пути, команды. Someday.

(2014-12-09 10:03)



Mezzanine Model Graph

Django project structure (2014-12-09 6:20)



Такова структура django проекта с приложением, статикой, админской обвязкой и шаблонами.

Стоит ли мне переделывать всё под неё - пойму, когда разберусь, что предлагает Mezzanine в этом отношении - rd "mezz", rd "mezz-tut".

mezz wysiwyg (2014-12-09 9:01)

Надо войти по quotrx.com/admin и выбрать при логине опцию admin. Тогда в панели администратора, в разделе Pages можно редактировать страницы wysiwyg редактором.

Удобно. Попробую так сделать те 3 формочки, что мне нужны. Хотя если reg/auth уже есть, стоит ли её сейчас менять? Кастомизировать, возможно.

С одной стороны хочется всё сделать по-своему, с другой - я реально не уверен, что это того стоит. Так, например, я привернул к проекту плюсы и web.py, а потом (как и подозревал) оказалось, что django с python и mezz достаточно.

Сначала освоюсь с возможностями mezz и django, если не хватит - только тогда уже буду делать что-то своё.

Однако, регистрация, поиск и CRUD должны быть на индексной странице сайта, так что нынешнюю индексную стоит забэкапить и написать свою.

Итог на обед (2014-12-10 11:51)

HTML5BP - Это чисто фронтэндная тулза. Не связанная с PDM. Не хочу тратить время на неё сейчас. Я плохо знаю JS и не силен в создании сайтов с нуля, во фронтэндной кухне.

Попробую обойтись PDM + модулями:

1. pkgz

Сначала посмотрю, какие есть модули - все интересные для создания форм я уже отобрал.

2.demo

Поковыряю demo-стенд mezz.

3. pycharm

Попробую использовать связку Pycharm LiveEdit + Chrome.

4. plan

В итоге, нужно создать максимально детальный план, как я завтра стану делать формы. На основе отвеченных утром вопросов, пунктов 1-3 этого поста и записей.

SOLID (object-oriented design) - Wikipedia, the free encyclopedia (2014-12-10 13:34)

Link: [1]SOLID (object-oriented design) - Wikipedia, the free encyclopedia

1. [https://en.wikipedia.org/wiki/SOLID_\(object-oriented_design\)](https://en.wikipedia.org/wiki/SOLID_(object-oriented_design))

Вопросы по mezz (2014-12-10 9:38)

0. Programming mezz pages

Bad idea, but just in case - [1]mezz-content-arch

1. mezz Page Tree

Each Page instance is stored in a hierarchical tree to form the site's navigation, and an interface for managing the structure of the navigation tree is provided in the admin via `mezzanine.pages.admin.PageAdmin` .

2. mezz admin customization [customizable]

[2]mezz-admin-cust

3. mezz search [customizable]

[3]mezz-search-eng

4. mezz reg/auth [customizable
]

[4]mezz-user-accounts

5. mezz WYSIWYG capabilities [?]

[5]mezz-wysiwyg [6]tinymce [7]redactor

Вопрос в том, есть ли способ редактировать страницу целиком, а не контент. Как в blogger - кастомизировать шаблон, дрэгндропать блоки.

6. Django's ModelAdmin class extension

[8]django-docs

7. Django admin actions

[9]django-docs

1. <http://mezzanine.jupo.org/docs/content-architecture.html>
2. <http://mezzanine.jupo.org/docs/admin-customization.html><http://mezzanine.jupo.org/docs/admin-customization.html>
3. <http://mezzanine.jupo.org/docs/search-engine.html>
4. <http://mezzanine.jupo.org/docs/user-accounts.html>
5. <http://mezzanine.jupo.org/docs/admin-customization.html#wysiwyg-editor>
6. <http://www.tinymce.com/>
7. <http://imperavi.com/redactor/>
8. <https://docs.djangoproject.com/en/dev/ref/contrib/admin/>
9. <https://docs.djangoproject.com/en/dev/ref/contrib/admin/actions/>

Терминальные утилиты (2014-12-10 9:38)

tree - просто бомба!

htop крут.

mc крут.

(2014-12-11 14:45)

Решил забить на abelson-sussman пока. Что-то это не улучшает моё понимание. Возможно, другая книга подошла бы больше, пока же - переключусь на python и django.

Может быть, я неправ. Но эту - очень сложно читать и она древняя. Сейчас нет времени на это.

Upgrade (2014-12-12 14:50)

Upgrade - необходим.

Такие развёрнутые tutoriales как Full Stack Python и Effective Django хороши, но всё же мне не хватает стройности и целостности книги.

Так что, после того как я решил отложить чтение Абельсона-Сассмана, теперь я решаю что сначала мне нужно прочесть книгу по Django.

Завтра до встречи я занимаюсь делом, так что апгрейдится буду перед сном. Вот такой апгрейд - действительно инвестирование времени, а не трата.

JPEG-isation of work (2014-12-14 10:34)

Я понял, что книга даёт больше инфы, чем мне надо, так что решил читать её более бегло, сделать zoom out в смысле подробности. Оставшееся добегу когда будет нужно.

Это же надо делать и с проектом.

Есть и философский аспект в этом: брать/делать только необходимое.

VirtualBox VM quickstart w/ LAMPDM (2014-12-15 5:23)

```
$ sudo apt-get install python
$ sudo apt-get install python-pip
$ sudo apt-get install build-essential python-dev
```

```
$ sudo apt-get install mysql-server
$ sudo apt-get install libmysqlclient-dev
$ sudo apt-get install python-mysqldb
```

```
#####
```

mysql creds:

```
mysql root | mysql -u root -p / STD
```

```
mysql mezz | mysql -u mezz -p / masterkey
```

```
#####
```

```
$ pip install mezzanine
# Create a project
$ mezzanine-project myproject
$ cd myproject
```

```
#####
```

```
@ local _settings.py
DATABASES = {
```

```
"default": {
# Ends with "postgresql_psycopg2", "mysql", "sqlite3" or "oracle".
"ENGINE": "django.db.backends.mysql",
# DB name or path to database file if using sqlite3.
"NAME": "quotrsys",
# Not used with sqlite3.
"USER": "mezz",
# Not used with sqlite3.
"PASSWORD": "masterkey",
# Set to empty string for localhost. Not used with sqlite3.
"HOST": "",
```

```
# Set to empty string for default. Not used with sqlite3.
```

```
"PORT": "",
```

```
}
```

```
}
```

```
#####
```

```
$ python manage.py collectstatic
```

```
$ python manage.py collecttemplates
```

```
$ python manage.py syncdb
```

```
##### dj/mezz site
```

```
login:
```

```
Login = admin (django superuser)
```

```
Password = masterkey / default
```

```
#####
```

```
$ python manage.py runserver
```

```
#####
```

```
point browser:
```

```
Url = 127.0.0.1:8000
```

```
#####
```

Программирование (2014-12-19 17:02)

П. работает также как всё остальное. Знаешь себя, знаешь предмет, набираешь опытЮ совершенствуешь навык, набираешься знаний, улучшаешь свою способность фильтровать информацию.

Я поторопился с "Pro Django", поэтому идёт тяжело и плохо. Терпение окупается - вот что я понял. RTFM!

Так что "Pro Django" я на время оставляю и буду читать "Definitive Django". Уверен, пойдёт гораздо легче. Книга большая, читать буду умным способом. Основы+то что нужно подробно, остальное по диагонали, но всё по порядку и с конспектом (конспектировать только ключевые моменты, без деталей - схемы и определения) затем - "Pro Django" до конца по диагонали.

Не знаю, что это займёт по времени, но мне уже не терпится начать программировать.

Буду стараться закончить с bg на этих выходных и спонедельника наконец приступить к коду.

Если в "Def Django" будет законченный пример приложения, возможно стоит его использовать как tut - так как я делал с google-python-class.

Я не люблю смешивать теорию с практикой хаотично. Но надо научиться видеть в хаосе гармонию. Двигаться итерациями: прочитал-понял-применил. Кажется, что это дольше, но -возможно - быстрее и эффективнее.

Я не люблю смешивать теорию с практикой, потому что я жадный - я хочу большой кусок теории, потом большой кусок практики. Может быть, это и не лучший подход.

Если же в "Def Django" не будет tut-a, то сразу после того как закончу читать книжки по Django, пройду tut на сайте. Хотя, вполне возможно, это один и тот же tut.

Возможно, мне стоит по другому взглянуть на соотношение теории и практики. Что должно быть первым? Как они должны соотноситься? В каких объёмах? По каким итерациям?

План на завтра есть, виртуалки есть, только я всё ещё не сплю.

dev ecology (2014-12-21 18:05)

I still often feel [1]"impostor syndrome". There's a lot of intellectual bullying in IT, it's nasty. Jedi Way is not to build yourself a pedestal out of your knowledge. I think people who do this are afraid to be incompetent themselves, so that's like a vicious circle that makes dev community ecology bad. But I've found out that ubuntu, python and django communities are very much positive.

1. https://en.wikipedia.org/wiki/Impostor_syndrome

Jedi Programmers (2014-12-21 18:08)

Если бы такая группа была, было бы круто. Не знаю, что бы мы делали, но точно бы нашли что.

Можно:

- * Учить прогать других товарищей
- Улучшать totjo
- Сделать свой проект
- Участвовать в благотворительных хакатонах
- Участвовать в конкурсах командой
- Обмениваться опытом

Но не стоит обольщаться - создать даже небольшую группу - непросто. Мне это точно нужно, так что я не зря стараюсь. Но я не буду навязываться слишком. Если получится - круто, нет - значит и так найдём чем заняться.

Django 1.7 migrations (2014-12-23 12:27)

1st time (models just coded)

* python manage.py migrate

after 1st time

* ->models.py (change your models' code)

* python manage.py makemigrations (create migrations for those changes)

* python manage.py migrate (apply those changes to the database)

Heartcore Dev (2014-12-23 17:24)

Я нашёл 2 варианта куда перенести блог и заложил их в pearltrees.

Пока что у меня нет ни материала, ни морального права создавать публичный блог.

После проекта. Пока - не время.

Терпение.

Microsoft (2014-12-25 15:38)

Стыдно сказать (немного), но пройдя хардкор, я научился ценить то что есть у MS - особенно Visual Studio.

Хотя, не ей единой.

Может быть, раз C# стал опенсорсом, они начнут продавать студию под linux? Для них это был бы win, я считаю.

Блог (2014-12-25 17:28)

Я назову его digital crusade

Найду новый клёвый пикселарт рыцарей для хедера

Заведу на гитхаб педжес или на logdown

Вот, отвлечению дань выплатил. Теперь буду заниматься делом.

Бессонница (2014-12-25 19:26)

Пробовал порно. Не особо помогло. Полез в техноические прокрастинаии. Выбирал площадку для технического блога.

LogDown оказался довольно убогой штукой - wordpress с codehighlighting в миниатюре. Никакой гибкости.

Сам вордпресс как и блоггер мне не подошли - там слишком попсово и людно.

Github Pages, которые все рекомендуют, превращают техно-блоггинг в отдельный челлендж поначалу.

Но теперь уже всё равно - фреймворком больше - фреймворком меньше. Так что ruby и jekyll как технология и github pages как хостинг. Фишка не в хардкорности, а в том, что я могу кастомизировать - раз, не плачу - два, учу новое - три. Хотя я бы предпочёл блог на меззанин. Возможно, стоит после проекта подумать об этом.

Ментальные инструменты для фильтрации шлака (2014-12-26 12:35)

[1]Три сита [2]Сократа

[3]Бритва [4]Оккама

1. http://www.primepuzzle.com/leeslightest/triple_filter_x.html
2. <https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D0%BA%D1%80%D0%B0%D1%82>
3. https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B8%D1%82%D0%B2%D0%B0_%D0%9E%D0%BA%D0%BA%D0%B0%D0%BC%D0%B0
4. https://ru.wikipedia.org/wiki/%D0%9E%D0%BA%D0%BA%D0%B0%D0%BC,_%D0%A3%D0%B8%D0%BB%D1%8C%D1%8F%D0%BC

Когда-нибудь у меня будет public dev blog (2014-12-26 17:32)

Я не буду делать его на logdown

Я не буду делать его на Github Pages: это Ruby и Jekyll, не в тему.

Я сделаю его на PDM. И сделаю его для себя.

После того, как проект будет закончен, я подготовлен, работа найдена.

Когда-нибудь. И блог будет называться... как? Я пока не решил.

Digital Trye | Digital Crusade | Heartcore Dev | Knight Code

И домен будет первого уровня, но не дорогой. А хостинг - можно попробовать один из [1]этих.

1. <https://wiki.python.org/moin/FreeHosts>

Python 3.4 | Django 1.7 (2014-12-27 0:22)

Нашёл способ юзать более современный питон на box2 (попутно угробил его, так что рпишлось клонировать с box1):

[1]askubuntu Q

Что касается меззанина, то Dj1.7 он ещё не поддерживает, а может ли он работать на чистом питоне 3.4 - вопрос зависимостей как его клея, так и его модулей. Это сложный вопрос и я думаю, что буду использовать питон2 для Mz tut, а для работы - комбинацию Py2.7-Dj1.6/1.7-Mz-3.1

Версионность ПО - отдельная сложная тема и вообще, и в этом случае тоже.

1. <http://askubuntu.com/questions/314773/rename-computer-in-ubuntu-13-04>

learning curve (2014-12-27 3:11)

Интересная концепция.

Django Tango Rango (2014-12-27 4:32)

То, какие вопросы задаются и какие ответы даются в самом начале проектирования приложения послужит мне для стадии планировании quotjo.

Я могу прямо идти по этому списку вопросов и делать свои схемы.

Rango в основе очень похоже на то базис quotjo.

Вопрос в том, поможет ли мне или помешает то, что я хочу привязать Mezzanine.

1) После танго (уже завтра) я возьмусь собрать в кучу свои знания по Django - посмотрю схемы и конспекты. Как-то всё соберу воедино в голове.

2) Потом прочитаю и поконспектирую доки по Mz.

3) Потом пройду 2 tutorials по Mz.

4) Потом надо будет собрать в кучу мысли по Mz поверх Dj и ответить себе на вопрос: делать ли мне сайт на чистом Django или использовать Mezzanine? Что будет быстрее для достижения моей цели - умного сайта цитат с API, user-driven content-ом, нашным дизигном?

5) Дальше я вернусь к 2й части танго как к шаблону и по ней составлю свой план. Т.к. я плохо представляю скорость и временные траты, план должен быть составлен для разработки методом "прогрессивного JPEG-a".

6) У меня будет две недели, чтобы сделать всё приложение-сайт.

* (Т) Шаблоны, css, js, html

(V) View, url схемы для сайта и сервиса, весь glue код

(М) Модели, хранение в БД через ОРМ, безопасность

+ тесты, доки, админка, авторизация, социальный ff, git

7) Боевой деплой можно в расчёт 2х недель не брать - да, придётся повозиться (мб даже пару дней или неделю), чтобы всё аккуратно перенести, но:

* у меня есть опыт

на DigitalOcean-е, который я хочу использовать как площадку, всё очень user-friendly

у меня будет отличная отмазка и перед родными и перед работодателем, если выйдет +неделя

Таким образом, главное - на утро понедельника оказаться в точке (6). Я дико невыспанный, завтра встречи, я отвлекаюсь. Но я сделаю всё, что могу.

import Dj models (2014-12-28 9:20)

When importing Django models, make sure you have imported your project's settings by importing django and setting the environment variable DJANGO_SETTINGS_MODULE to be the project setting file. Then you can call django.setup() to import the django settings. If you don't, an exception will be raised. This is why we import Category and Page after the settings have been loaded.

gads

BlogBook v0.5,
L^AT_EX 2_ε & GNU/Linux.
<http://www.blogbooker.com>

Edited: December 28, 2014

