

Do it Later with Resque & Delayed Job

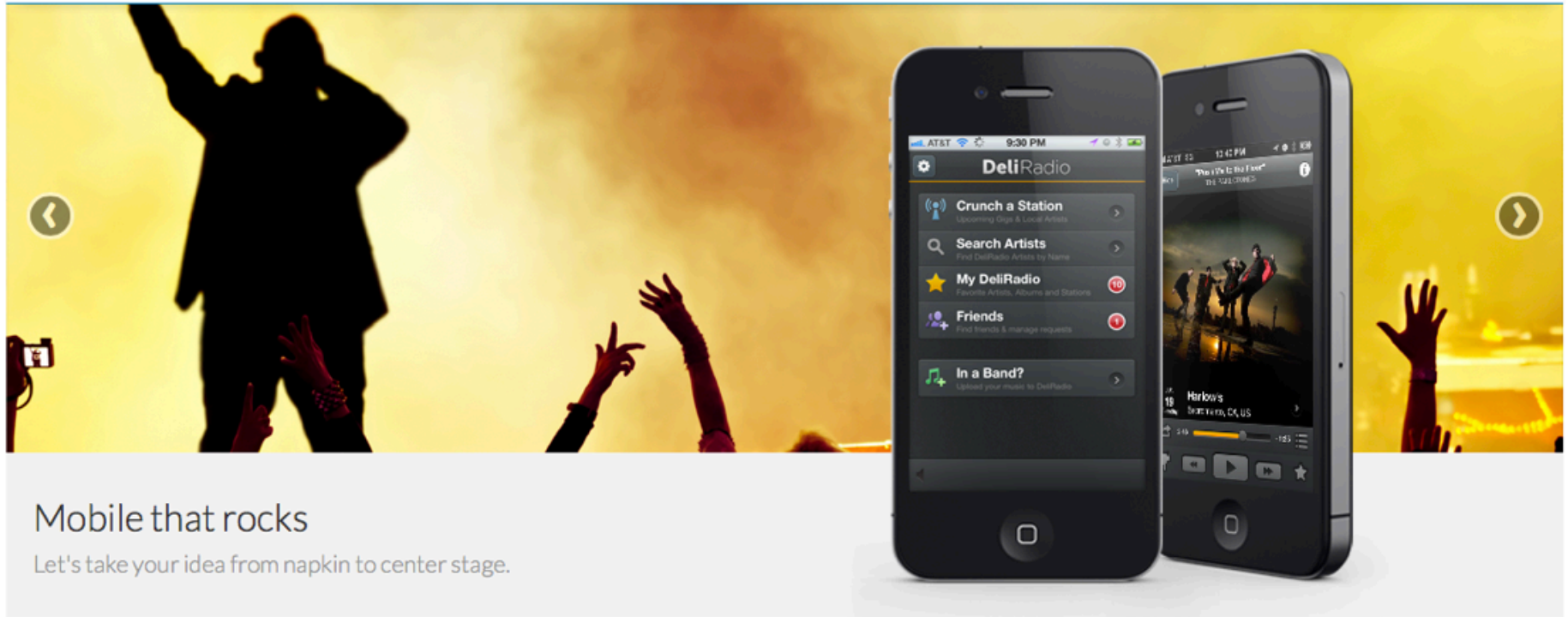
**Ben Scheirman
@subdigital**

<http://benscheirman.com>

Do it Later with Resque & Delayed Job ...and Sidekiq?

**Ben Scheirman
@subdigital**

<http://benscheirman.com>



Mobile that rocks

Let's take your idea from napkin to center stage.

We are hiring!
Rails & iOS



Customizing UITabBar

Episode #22 | 12 minutes | published on 06/28/2012

SUBSCRIBERS ONLY

In this episode, I continue deconstructing Foursquare's custom UI. This time I focus on how to customize the UITabBar with the iOS 5 customization APIs.

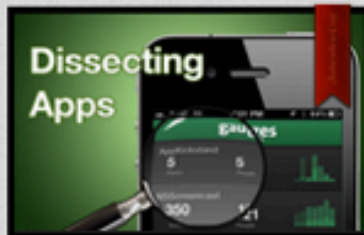


Customizing UINavigationController

Episode #21 | 21 minutes | published on 06/26/2012

SUBSCRIBERS ONLY

In this episode, I decompose the Foursquare UI and recreate the custom navigation bar, using the iOS 5 customization APIs. You'll see how to set a custom background image, a title view that you can tap on, a custom bar button item, and a custom back button style.



Dissecting Apps

Episode #20 | 13 minutes | published on 06/14/2012

SUBSCRIBERS ONLY

It can be helpful to draw inspiration from how existing applications are built. In this screencast, we'll look at how to extract & view images from iOS applications you've downloaded from the App Store. In addition, we'll use a proxy to intercept and inspect network traffic so you can see how application APIs behave.



Push Notifications

Episode #19 | 18 minutes | published on 06/07/2012

SUBSCRIBERS ONLY

Ruby + Web = 

Expressive

Flexible

Malleable

Fun

Ruby is not known for:

Raw Speed

Multi-Threading

But, Ruby apps can scale

Plenty of examples

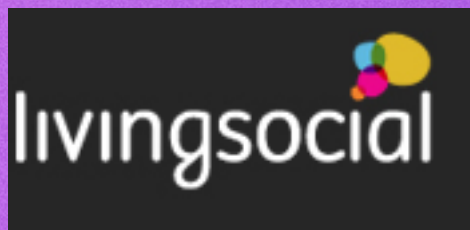
But, Ruby apps can scale

Plenty of examples



GROUPON®

github
SOCIAL CODING



How do Rails apps scale?

Caching

Multiple web processes

Mongrels

Unicorn workers

Heroku dynos

Etc

Distributing the Load

Heroku's Dyno Model



Heroku's Dyno Model



**Keep
Requests
Lightweight**

**Perform
complex work
in the
background**

What work should we perform in the background?

What work should we perform in the background?

Sending emails

Communicating with 3rd party APIs

Calculating stats

Rebuilding indexes

Resizing/processing photos

Post processing of models

Encoding audio/video

Extract info out of media files

Warm caches

Many solutions for this...

Delayed Job

Resque

SideKiq*

Delayed Job

https://github.com/collectiveidea/delayed_job

Just a gem

Uses your database*

Rake task to run workers


```
# Gemfile
```

```
gem 'rails'
```

```
gem 'sqlite'
```

```
gem 'delayed_job_active_record'
```



```
$ rails generate delayed_job
```

```
$ rake db:migrate
```



```
class Foo
  def bar
    puts "baz"
  end
end

(irb) > Foo.new.delay.bar
```

Caller decides async


```
class Foo
  def bar
    puts "baz"
  end
  handle_asynchronously :bar
end

(irb) > Foo.new.bar
```

Caller doesn't care...


```
class MyJob
  def perform
    puts "I'm doing background work!"
    sleep 10
  end
end
```

```
(irb)> Delayed::Job.enqueue MyJob.new
```


Demo: Fetching Link Titles (using Delayed Job)

Handling Failures

Failure count + sliding delay
Continue until job is 30 days
old

Pros of DJ

Easy

Works with AR / Mongoid

Just a gem

Failure handling

Downsides of DJ

Downsides of DJ

Somewhat stale development*

Uses your app's database

Table locking

**Worker queries expensive
with a large queue**

No built-in dashboard

Some things to look at

Delayed Job Admin

Delayed Job (Mongoid adater)

Resque

Based on Redis*

Multiple queues

Queue Priority

Built-in Dashboard

Why Queue Priority?

calculate_stats queue

 ← **deliver_purchase queue**

Jobs in Resque

Plain classes

Must set class variable `queue`

Must implement class method
`perform`

Defining Jobs in Resque

```
class SendNewsletterJob
  @queue = :mailer

  def self.perform(newsletter_id)
    newsletter = Newsletter.find(newsletter_id)
    User.subscribed.find_each do |user|
      UserMailer.newsletter(
        user,
        newsletter).deliver
    end
  end
end
```


Enqueueing Jobs in Resque

```
Resque.enqueue(SendNewsletterJob, 256)
```


Redis



<http://redis.io>

Key value store

Fast

Atomic

$O(1)$ push & pop

Master-slave configuration

Redis Configuration



Store entire set in memory

Configure sync to disk

Every write (SLOW)

Don't Flush (FAST)

Every second (COMPROMISE)

Installing Redis



On a Mac..

```
brew install redis
```

On the server (Ubuntu)

```
wget http://redis.googlecode.com/files/redis-2.4.13.tar.gz  
tar xzvf redis-2.4.13  
cd redis-2.4.13  
make  
sudo make install
```

Hosted with Redis To Go

<http://redistogo.com/>

Installing Resque in a Rails app

Gemfile

```
gem 'resque'
```

lib/tasks/resque.rake

```
require 'resque/tasks'
```


Running the web console

Separate sinatra app

resque-web

Mounted in your rails app

routes.rb

Mounting the web console

```
require 'resque/server'

YourApp::Application.routes.draw do
  mount Resque::Server.new, :at => "resque"
  ...
end
```


Running the worker(s)

```
$ QUEUE=* rake resque:work
```


Selecting Queues / Priorities

```
$ QUEUE=urgent,* rake resque:work
```


Make sure Rails environment is loaded

```
$ QUEUE=* rake environment rescue:work
```


Make sure Rails environment is loaded

```
# lib/tasks/resque.rake

task "resque:setup" => :environment do
  # any other setup
end
```


Demo: Fetching User Avatars (using Resque)

Pros of Resque

Pros of Resque

Easy*

Fast

No impact on app db perf

Admin dashboard

Flexible

Drawbacks of Resque

More complex

More things to setup, monitor

Wonky reporting on Heroku

Things to look at

Resque Mailer

https://github.com/zapnap/resque_mailer

Resque Scheduler

<https://github.com/bvandenbos/resque-scheduler>

Resque Retry Support

<https://github.com/lantins/resque-retry>

Queues

The list below contains all the registered queues with the number of jobs currently in the queue. Select a queue from above to view all jobs currently pending on the queue.

Name	Jobs
avatar	0
encoder	0
mailchimp	5984
mailer	0
process_asset	0
remote_encoder	0
failed	5

19 of 19 Workers Working

The list below contains all workers which are currently running a job.

	Where	Queue	Processing
	chimera.local:37561	MAILCHIMP	MailchimpSubscribeJob just now
	chimera.local:37570	MAILCHIMP	MailchimpSubscribeJob just now
	chimera.local:37568	MAILCHIMP	MailchimpSubscribeJob just now
	chimera.local:37562	MAILCHIMP	MailchimpSubscribeJob just now
	chimera.local:37559	MAILCHIMP	MailchimpSubscribeJob just now
	chimera.local:37572	MAILCHIMP	MailchimpSubscribeJob just now
	chimera.local:37573	MAILCHIMP	MailchimpSubscribeJob just now
	chimera.local:37564	MAILCHIMP	MailchimpSubscribeJob just now
	chimera.local:37571	MAILCHIMP	MailchimpSubscribeJob just now
	chimera.local:37558	MAILCHIMP	MailchimpSubscribeJob just now



subdigital Ben Scheirman

I hope mail chimp is ok with this :) [pic.twitter.com/ikhyFaLT](#)

[Details](#)

Sidekiq

**“What if we could replace 20
Resque / DJ workers with 1
Sidekiq worker?”**



“That’s a bold statement”

Sidekiq

**Master
Process**





By the power of Celluloid*

Takeaways

Takeaways

Lots of work can be done in the background

Keep requests lightweight, defer as much as possible

Choose your own adventure

Write job classes in ruby

Monitor your jobs!

Questions?

Ben Scheirman

@subdigital

<http://benscheirman.com>

<http://nsscreencast.com>