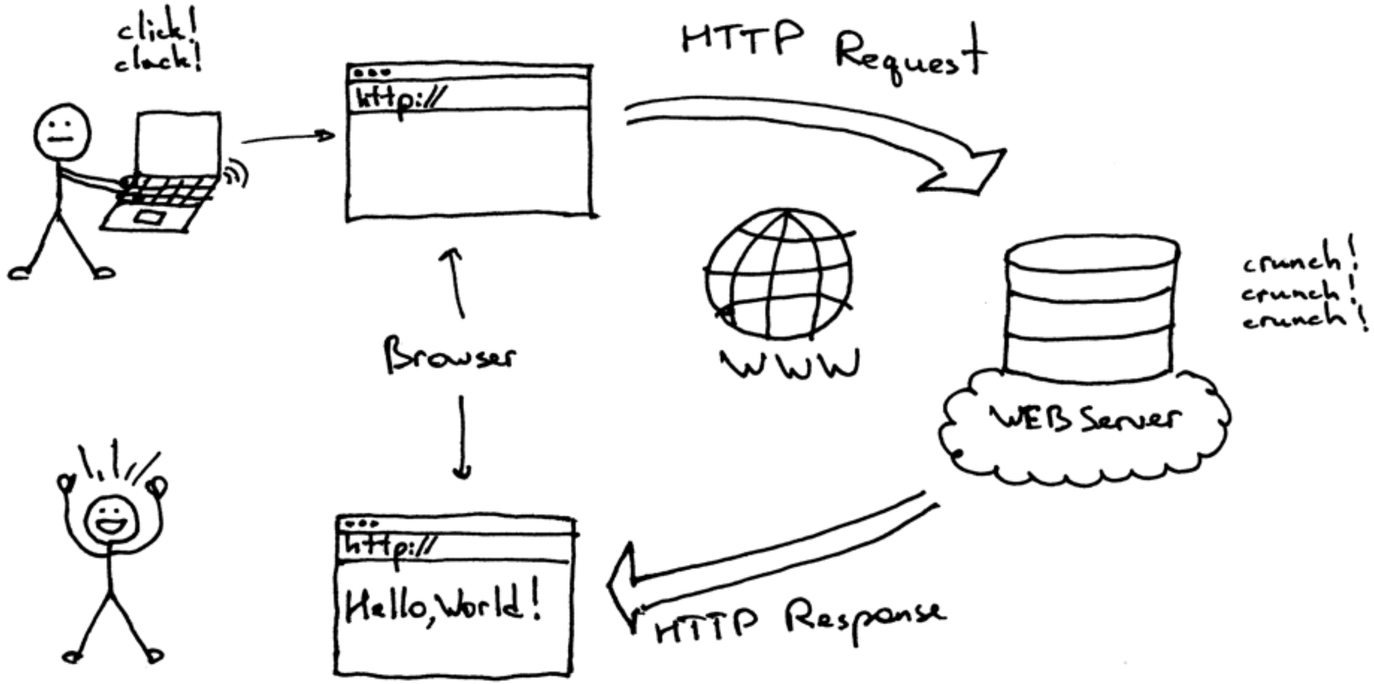
BACKGROUND JOBS

(in Rails)

Agenda

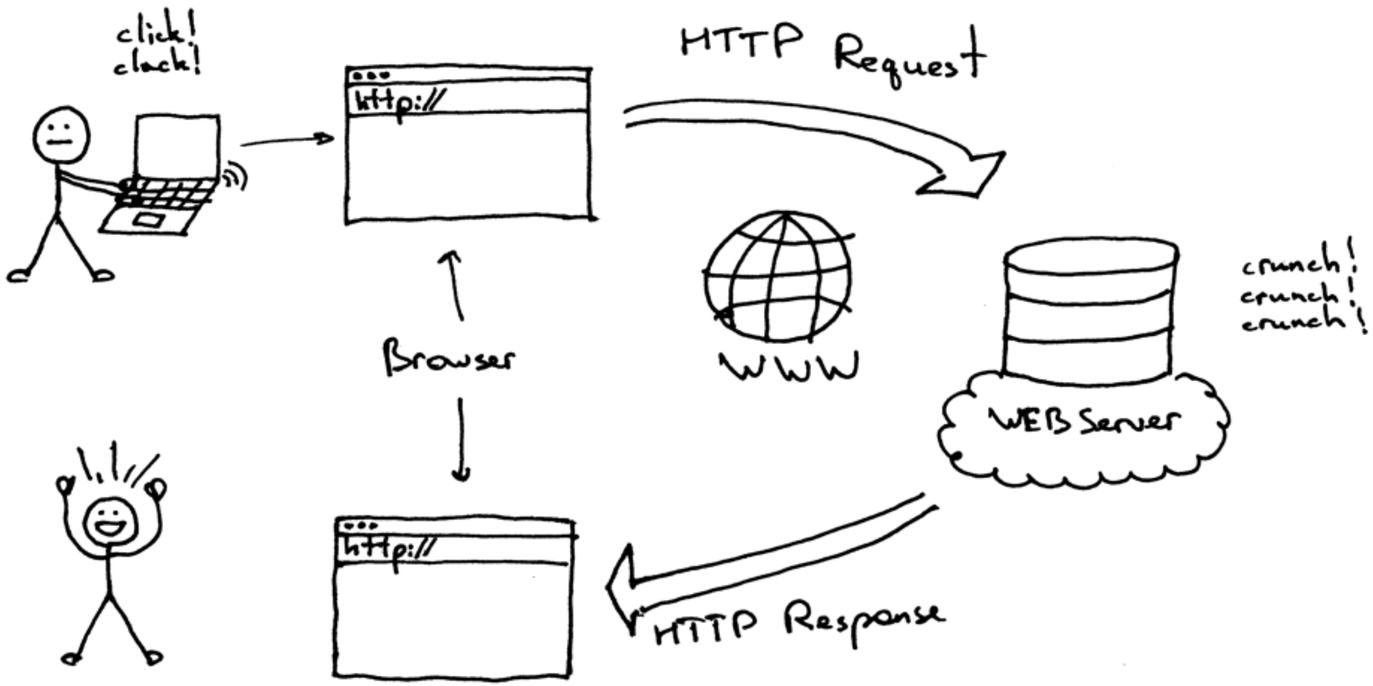
- What’s a background job?
- Getting started with ActionMailer & ActiveJob
- Stepping up your game with Sidekiq

Synchronous vs Asynchronous



https://ruslanspivak.com/lsbaws-part1/

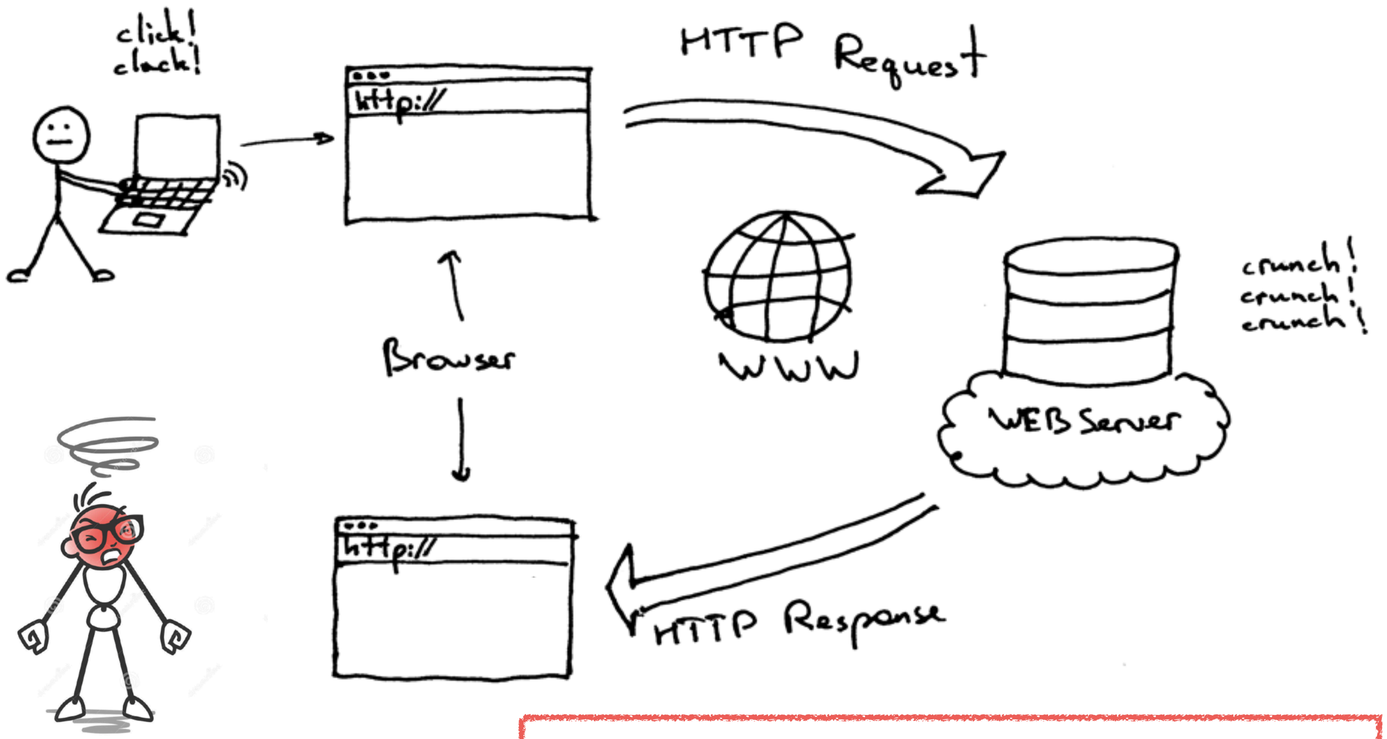
Synchronous vs Asynchronous



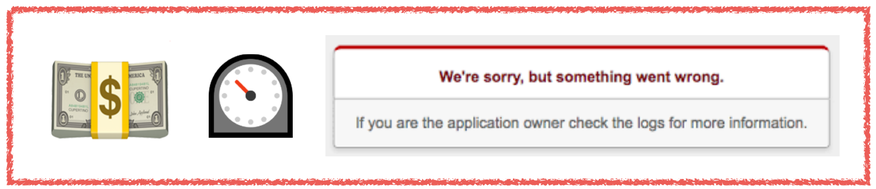
- Sending emails
- Generating PDFs/CSVs
- Processing payments
- Calling external APIs

https://ruslanspivak.com/lsbaws-part1/

Synchronous vs Asynchronous

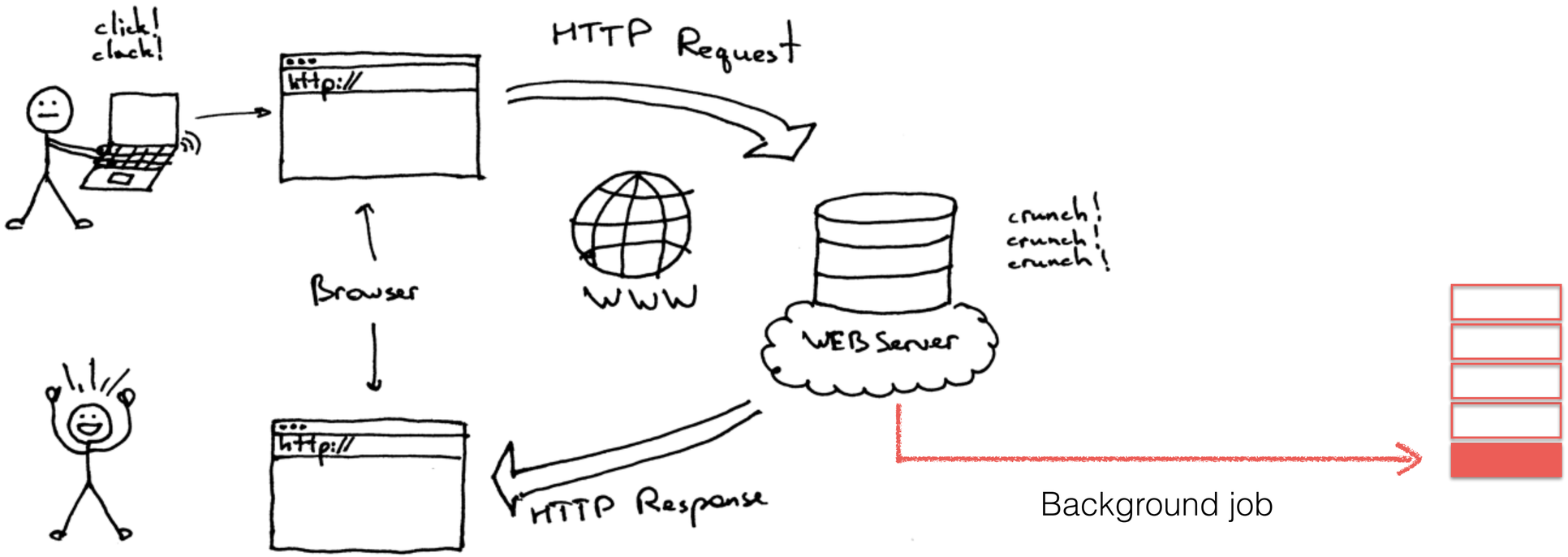


- Sending emails
- Generating PDFs/CSVs
- Processing payments
- Calling external APIs



https://ruslanspivak.com/lsbaws-part1/

Synchronous vs Asynchronous



https://ruslanspivak.com/lsbaws-part1/

Getting Started

with

ActionMailer and ActiveJob

Leanne

Cindy
(Leanne's Mom)

From: **Cindy Shapton** <[redacted]@sympatico.ca>
Date: Sun, Jul 7, 2013 at 7:56 AM
Subject: sun

Good morning Leanne Jonathan Pauline Nick Marie Jonathon Philip and Mathew

Looks like you have beautiful weather in Vancouver John
Hope you are enjoying your weekend Leanne, Hot and humid in Toronto.
Pauline I hope it doesn't rain for your party.
Have a good day! Love yas! Take good care!!

Goals:

1. Create an email 
2. With the day's temperature 
3. Sent 1/day 

ActionMailer

```
$ rails generate mailer weather_report greet
```

ActionMailer

```
$ rails generate mailer weather_report greet

  create  app/mailers/weather_report_mailer.rb
  create  app/mailers/application_mailer.rb
  invoke  erb
  create    app/views/weather_report_mailer
  create    app/views/layouts/mailer.text.erb
  create    app/views/layouts/mailer.html.erb
  create    app/views/weather_report_mailer/greet.text.erb
  create    app/views/weather_report_mailer/greet.html.erb
```

ActionMailer

mailers/weather_report_mailer.rb

```
class WeatherReportMailer < ApplicationMailer

  def greet
    @greeting = "Hi"

    mail to: "to@example.org"
  end
end
```

views/weather_report_mailer/greet.html.erb

```
<h1>Test#greet</h1>

<p>
  <%= @greeting %>, find me in
app/views/test_mailer/greet.html.erb
</p>
```

ActionMailer

mailers/weather_report_mailer.rb

```ruby
class WeatherReportMailer < ApplicationMailer

  def greet
    @greeting = "Hi"

    mail to: "to@example.org"
  end
end
```

views/weather_report_mailer/greet.html.erb

```erb
<h1>Test#greet</h1>

<p>
  <%= @greeting %>, find me in
app/views/test_mailer/greet.html.erb
</p>
```

```
=> <FamilyMember id: 1, email: "test@gmail.com", "2017-05-07 16:49:17", updated_at:
"2017-05-07 16:49:17", name: "Jane", city: "Toronto">
```

```
=> <FamilyMember id: 1, email: "test@gmail.com", "2017-05-07 16:49:17", updated_at:
"2017-05-07 16:49:17", name: "Jane", city: "Toronto">
```

```ruby
class WeatherFetcher

  def get_weather(city)
    latlng = Geocoder.coordinates(city)
    api_key = ENV.fetch("DARK_SKY_API_KEY")

    url = HTTParty.get(
      "https://api.forecast.io/forecast/#{api_key}/#{latlng.first},#{latlng.last}",
      headers: { 'ContentType' => 'application/json' }
    )

    response = JSON.parse(url.body)

    in_celcius(response['currently']['temperature'])
  end

  private

  def in_celcius(temperature_in_f)
    ((temperature_in_f - 32) * 5.0 / 9.0).round
  end

end
```

mailers/weather_report_mailer.rb

```
class WeatherReportMailer < ApplicationMailer
  default from: 'mom@gmail.com'

  def greet(family_member)
    @family_member = family_member
    @weather_info =
WeatherFetcher.new.get_weather(family_member.city)

    mail to: family_member.email, subject: "#
{Date.today.strftime("%A")} Morning Weather
Report"
  end
end
```

views/weather_report_mailer/greet.html.erb

```
<h1>Good morning <%= @family_member.name %></h1>

<p>Today will be <%=@weather_info %>°C</p>
```

http://localhost:3000/rails/mailers/weather_report_mailer/greet.html

mailers/weather_report_mailer.rb

```
class WeatherReportMailer < ApplicationMailer
  default from: 'mom@gmail.com'

  def greet(family_member)
    @family_member = family_member
    @weather_info =
WeatherFetcher.new.get_weather(family_member.city)

    mail to: family_member.email, subject: "#
{Date.today.strftime("%A")} Morning Weather
Report"
  end
end
```

views/weather_report_mailer/greet.html.erb

```
<h1>Good morning <%= @family_member.name %></h1>

<p>Today will be <%=@weather_info %>°C</p>
```

http://localhost:3000/rails/mailers/weather_report_mailer/greet.html

ActiveJob

```
$ rails generate job send_weather_report
  create app/jobs/send_weather_report_job.rb
```

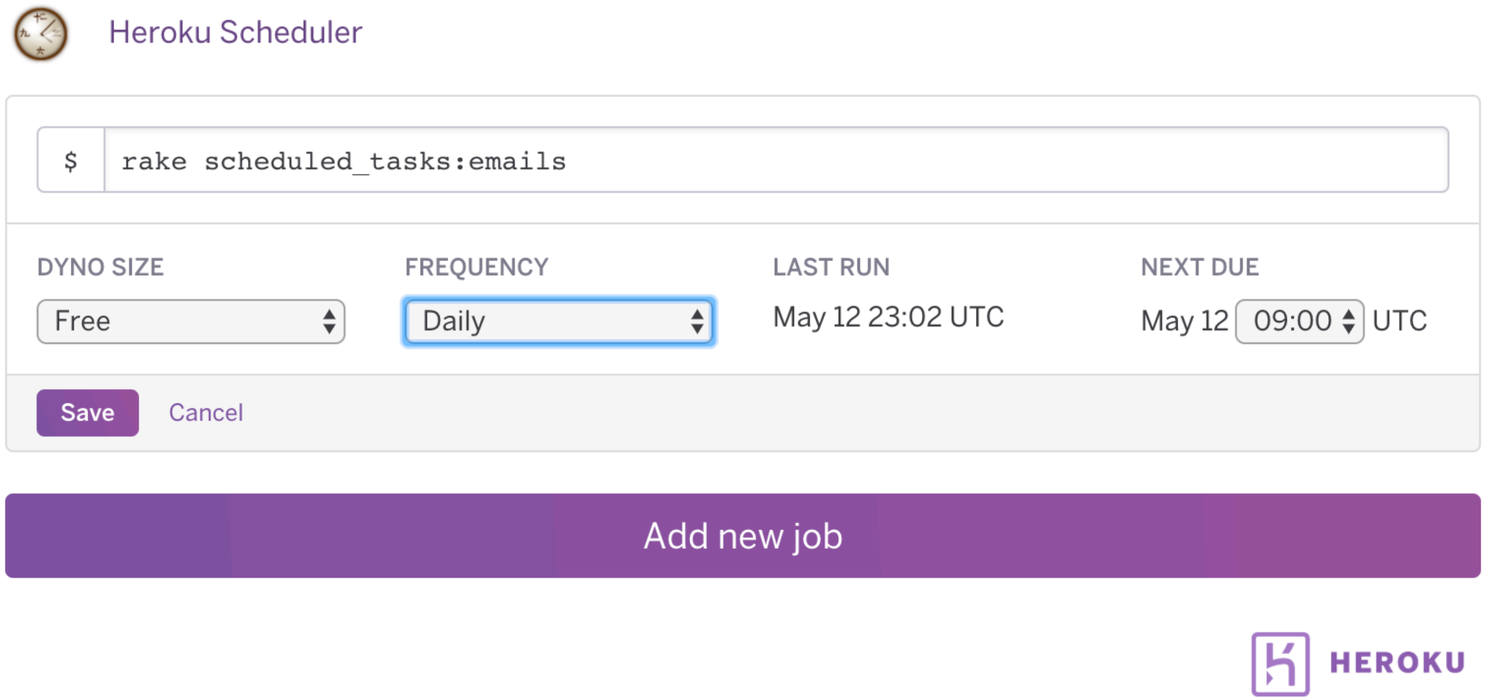
ActiveJob

```
$ rails generate job send_weather_report
  create app/jobs/send_weather_report_job.rb
```

```
class WeatherReportJob < ApplicationJob
  queue_as :default

  def perform(*args)
    # Do something later
  end
end
```

```ruby
class SendWeatherReportJob < ApplicationJob

  def perform(family_member_id)
    family_member = FamilyMember.find(family_member_id)
    WeatherReportMailer.greet(family_member).deliver_now
  end

end
```

Queuing the Job



HEROKU

Rake

Allows you to define and execute “tasks” from the command line

lib/tasks/scheduled_tasks.rake

```
namespace :scheduled_tasks do

  desc "Runs scheduled tasks"
  task :emails do
    FamilyMember.all.each do |family_member|
      SendWeatherReportJob.perform_now(family_member.id)
      puts "Executing SendWeatherReportJob for #{family_member.name}"
    end
  end

end
```

HEROKU

```
$ rake scheduled_tasks:emails
```

Pause & Recap

1. Created a mailer
2. Fetched a `family_member` and some `weather_info`
3. Created a job that tells the mailer to send the email
4. Created a rake task that executes the job
5. We used **heroku scheduler** to schedule the rake task in production

Sidekiq

- Why use it? I lied a little…

What we have → **What we want**

```
SendWeatherReportJob.perform_now
```

```
SendWeatherReportJob.perform_later
```

Sidekiq

- Why use it? I lied a little…

What we have → **What we want**

```
SendWeatherReportJob.perform_now
```

```
SendWeatherReportJob.perform_later
```

- Rails by default comes with an “immediate runner” queuing system
- Only keeps the jobs in RAM
- Process ends —> 👋 job
- Rake processes end as soon as they’re over!
- Job queues, never executes

Sidekiq

- Background processing framework for Ruby

Creating the Job	Redis	Sidekiq server
`SendWeatherReportJob.perform_async`	Stores job data	Pulls jobs from the queue in Redis and processes them

Sidekiq

Gemfile

```
gem 'sidekiq'
```

config/application.rb

```
class Application < Rails::Application
  # ...
  config.active_job.queue_adapter = :sidekiq
end
```

config/routes.rb

```
require 'sidekiq/web'

Rails.application.routes.draw do
   mount Sidekiq::Web => '/sidekiq'
end
```

Sidekiq

jobs/send_weather_report_job.rb

```ruby
class WeatherReportJob
include Sidekiq::Worker

  def perform(params = {})
    family_member = FamilyMember.find(params['family_member_id'])
    WeatherReportMailer.greet(family_member).deliver_now
  end

end
```

Sidekiq

lib/tasks/scheduled_tasks.rake

```
desc "Runs scheduled tasks"
  task :emails do
    FamilyMember.all.each do |family_member|
      SendWeatherReportJob.perform_async(family_member_id: family_member.id)
      puts "Executing SendWeatherReportJob for #{family_member.name}"
    end
  end
```

Sidekiq

lib/tasks/scheduled_tasks.rake

```
desc "Runs scheduled tasks"
  task :emails do
    FamilyMember.all.each do |family_member|
      SendWeatherReportJob.perform_async(family_member_id: family_member.id)
      puts "Executing SendWeatherReportJob for #{family_member.name}"
    end
  end
```

```
gem 'clockwork'
```

clock.rb

```
every(1.day, 'send_weather_report_job', at: '08:00') {
  `rake scheduled_tasks:emails`
}
```

~~Heroku Scheduler~~

Sidekiq

```
gem 'foreman'
```

Procfile

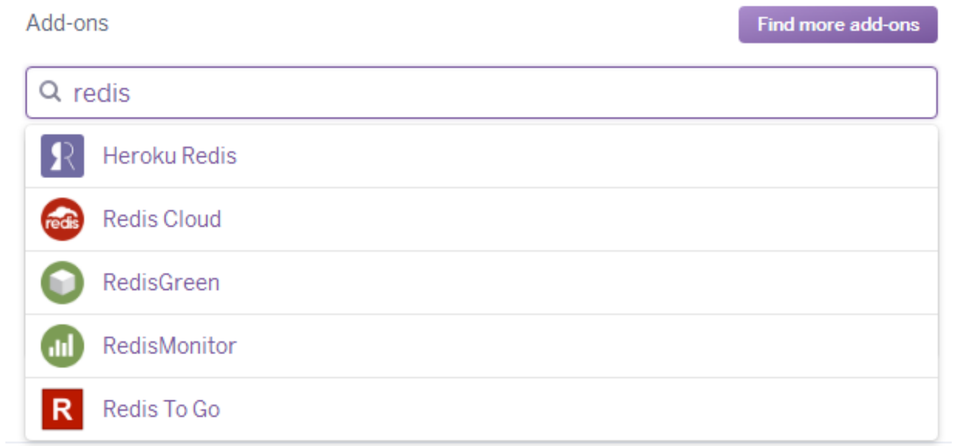
```
web: bundle exec puma -C config/puma.rb   # command to start your server
worker: bundle exec sidekiq -c 3          # start sidekiq
clock:  bundle exec clockwork clock.rb    # start clockwork
```

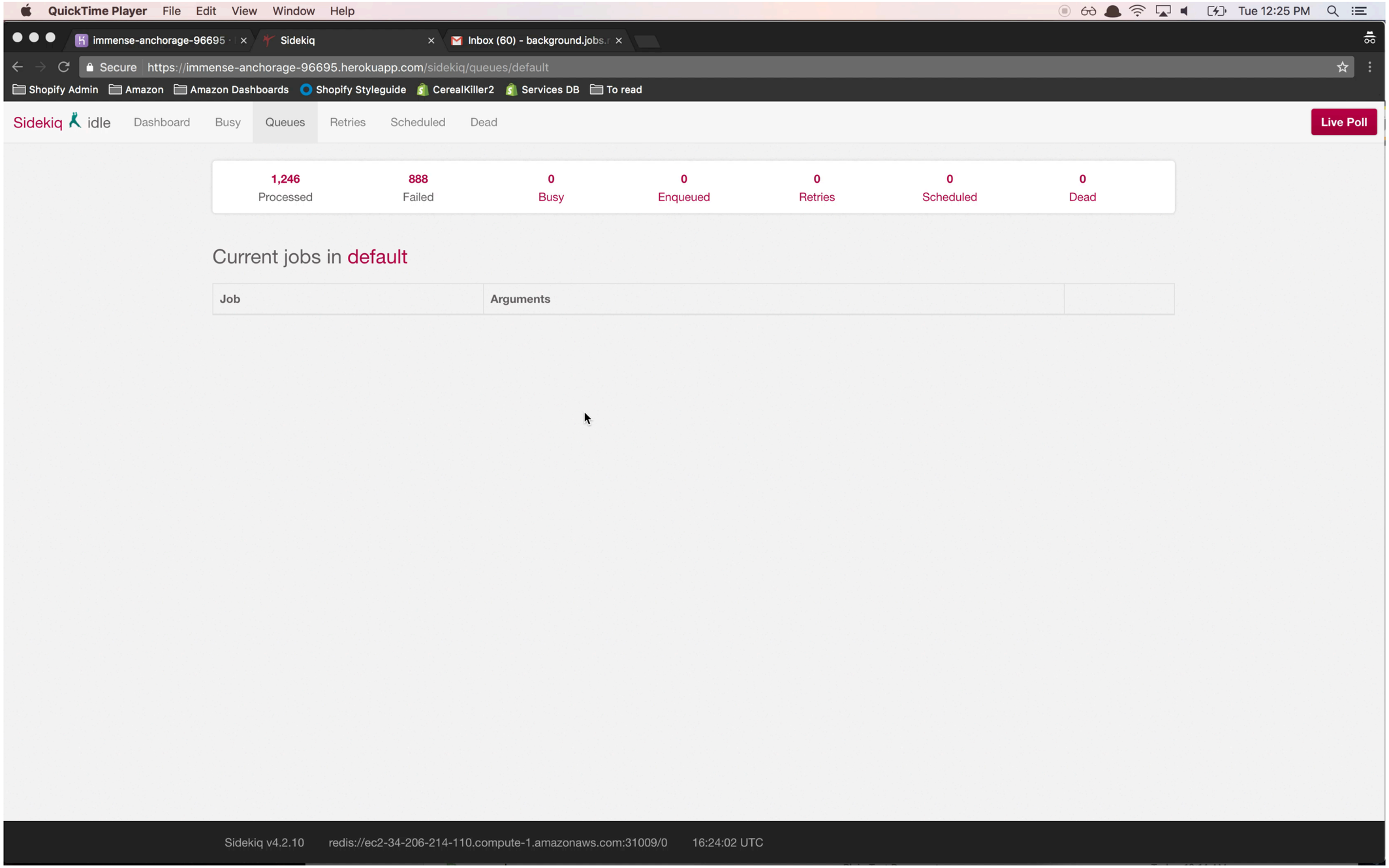
Sidekiq

```
gem 'foreman'
```

Procfile

```
web: bundle exec puma -C config/puma.rb   # command to start your server
worker: bundle exec sidekiq -c 3          # start sidekiq
clock:  bundle exec clockwork clock.rb    # start clockwork
```




QuickTime Player File Edit View Window Help
Tue 12:25 PM
immense-anchorage-96695
Sidekiq
Inbox (60) - background.jobs.r
Secure https://immense-anchorage-96695.herokuapp.com/sidekiq/queues/default
Shopify Admin
Amazon
Amazon Dashboards
Shopify Styleguide
CerealKiller2
Services DB
To read
Sidekiq idle
Dashboard
Busy
Queues
Retries
Scheduled
Dead
Live Poll
1,246 Processed
888 Failed
0 Busy
0 Enqueued
0 Retries
0 Scheduled
0 Dead
Current jobs in default
Job
Arguments
Sidekiq v4.2.10
redis://ec2-34-206-214-110.compute-1.amazonaws.com:31009/0
16:24:02 UTC

jenna.blumenthal@shopify.com

@jennaleeblume

github.com/jennaleeb/automom

References

https://github.com/mperham/sidekiq/

https://robots.thoughtbot.com/action-mailer-and-active-job-sitting-in-a-tree

http://blog.scoutapp.com/articles/2016/02/16/which-ruby-background-job-framework-is-right-for-you

https://ryanboland.com/blog/writing-your-first-background-worker/

http://stackoverflow.com/questions/38315548/rails-async-active-job-doesnt-execute-code-while-inline-does

https://www.youtube.com/watch?v=GBEDvF1_8B8