

Cron Job With Sidekiq

Basically cron jobs use for scheduling the tasks to be executed at specific time in future or to be execute periodically. For example, if we want to send out a notice at every morning, we want to send mail at every Monday etc.

In this blog we will know how to setup the cron jobs in our [Ruby on Rails application](#) in which we will need to use Redis server and Sidekiq in Ruby on Rails.

If you need to know how to configure Redis Server and sidekiq in your rails project then go through below link:

<http://tecorb.com/2014/10/07/install-and-use-redis-server/>

Lets proceed for adding cron jobs with the help of Sidekiq in our Ruby on Rails Application:

Step 1: First of all we need to add `gem "sidekiq-cron"` into our Gem file and do **bundle install**.

Step 2: Create a `workers` directory inside your `app/workers`

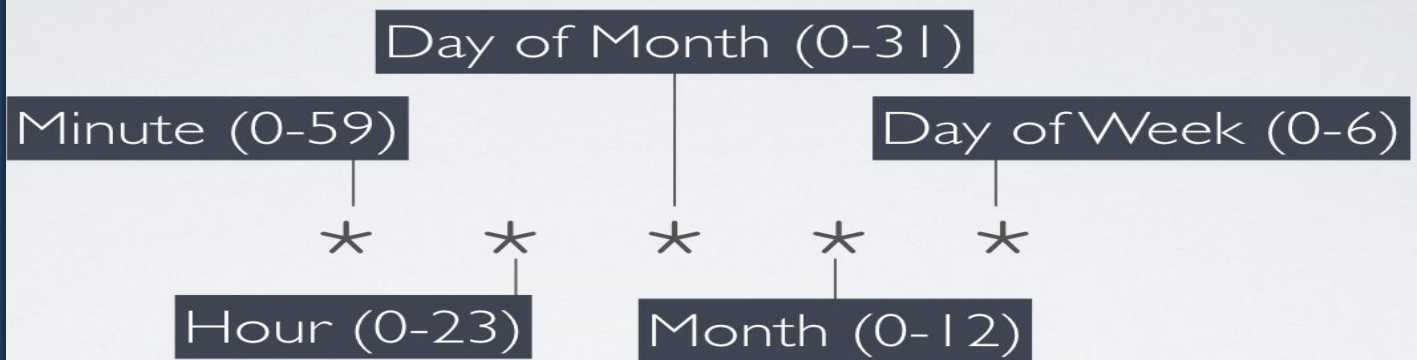
Step 3: Create a `cron_wroker.rb` file inside `app/workers/ cron_wroker.rb` directory and write a perform action in this class after including Sidekiq: Worker:

```
class CronWorker
  include Sidekiq::Worker
  def perform()
    # Write code for your cron job here.
  end
end
```

Step 4: Create a `schedule.yml` file inside config directory

```
generate_random_job:
  cron: " * * 12 * * 1 "
  class: "CronWorker"
  queue: default
```

Here " * * * * * " is simple cron format yet powerful and flexible way to define time and frequency of various actions:



```
0 9-18 * * 1-5
```

once an hour
from 9:00AM - 6:00PM
on Monday through Friday

Step 5: Now, make a sidekiq.rb file inside initializers in config directory:

```
Sidekiq.configure_server do |config|
  config.redis = { url: 'redis://localhost:6379/0' }
  sched_file = "config/schedule.yml"
  if File.exists?(sched_file)
    Sidekiq::Cron::Job.load_from_hash YAML.load_file(sched_file)
  end
end

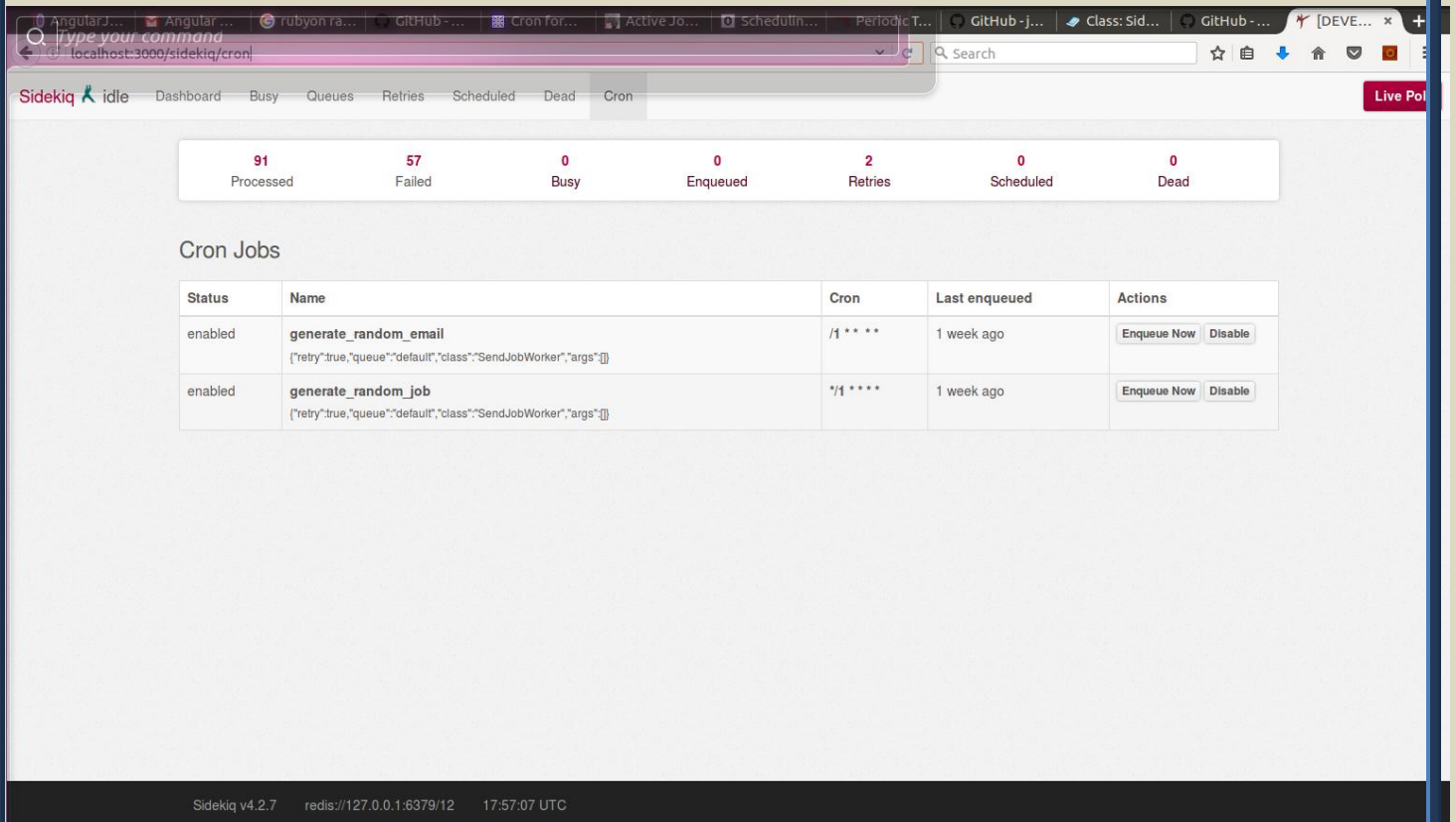
Sidekiq.configure_client do |config|
  config.redis = { url: 'redis://localhost:6379/0' }
end
```

Step 6: Now we have to start scheduling through running the *sidekiq* and *redis server*

Step 7: Add cron jobs to Sidekiq's web UI, require this in your **routes.rb** :

```
require 'sidekiq/web'  
require 'sidekiq/cron/web'  
mount Sidekiq::Web => '/sidekiq'
```

Step 8: Now run your localhost server and check your job status on Sidekiq's web UI by typing <http://localhost:3000/sidekiq/cron> you get this screen:



The screenshot shows the Sidekiq web interface. At the top, there are navigation tabs: Dashboard, Busy, Queues, Retries, Scheduled, Dead, and Cron. The Cron tab is active. Below the navigation, there is a summary bar with the following statistics:

91	57	0	0	2	0	0
Processed	Failed	Busy	Enqueued	Retries	Scheduled	Dead

Below the summary bar, there is a section titled "Cron Jobs" with a table listing the jobs:

Status	Name	Cron	Last enqueued	Actions
enabled	generate_random_email <small>[{"retry": true, "queue": "default", "class": "SendJobWorker", "args": []}]</small>	/1 * * * *	1 week ago	Enqueue Now Disable
enabled	generate_random_job <small>[{"retry": true, "queue": "default", "class": "SendJobWorker", "args": []}]</small>	*1 * * * *	1 week ago	Enqueue Now Disable

At the bottom of the interface, there is a footer with the following information: Sidekiq v4.2.7, redis://127.0.0.1:6379/12, 17:57:07 UTC.

If you are, looking to Develop and design for your project then feel free to contact us at any time

Call Us- +91 9910781148, +1 8033353593

Email: company@tecorb.com

Skype- tecorb

“We maintain transparency with our customers. Our values and business ethics has given us repeated customers. We tend are proud to be a reliable outsourcing partner for many clients across the World.”