Ruby on Rails Action Cable Tutorial for Developing Real Time Web Application

Action Cable is a powerful feature introduced in Rails 5. Using action cable one can develop real-time web applications. Action cable uses Web Sockets to create full duplex communication between application’s server and client.

Using action cable feature we can create any real-time functionality like live chatting, which updates chats, show new messages, notifications, etc without reloading of the page. Action cable basically keeps **ruby on rails** server data and client data updated using web sockets which make the application more feasible to use.

Action cable is not supported by RAILS VERSION > 5.

PREREQUISITES:

This **ruby gem** requires jQuery. If jQuery is not present in the project then simply add “jquery_rails” gem to gemfile.rb and redis-server in the system.

STEPS REQUIRED TO USE ACTION CABLE:

Add following in Gemfile.rb file :

```
gem 'redis', '~> 4.0'
```

Also add follwing in application.js file :

```
//= require rails-ujs
//= require jquery
//= require turbolinks
//= require_tree
```

Thats all about installation process.

Let’s see an **EXAMPLE**:

For creating **Notification** model, controller and views using scaffold command just run:

```
rails g scaffold Notification name:string
```

Then create a **Channel** , We call this channel “WebNotifications”.

```
rails generate channel WebNotifications
```

And we add some CoffeeScript files to show notifications to all users in **javascripts/page.coffee**:

```
App.room = App.cable.subscriptions.create "WebNotificationsChannel",
  received: (data) ->
```

```
     $('#notification div').append '<li>' + data['notification'] + '</li>'
      $('#notifications-count,.notifications-count').text data['count']
```

And finally we need to add the below code to the channels **web_notifications_channel.rb**:

```
class WebNotificationsChannel < ApplicationCable::Channel
 def subscribed
  stream_from "web_notifications_channel"
 end
 def unsubscribed
 end
end
```

Changes in **notifications_controller.rb** file To call the action cable:

```
def create
  @notfication = Notfication.new(notfication_params)
  respond_to do |format|
   if @notfication.save
    format.html { redirect_to @notfication, notice: 'Notfication was successfully created.' }
    format.json { render :show, status: :created, location: @notfication }
   else
    format.html { render :new }
    format.json { render json: @notfication.errors, status: :unprocessable_entity }
   end
```

```
    end
    ActionCable.server.broadcast 'web_notifications_channel', notification: @notfication.name, count: Notification.all.count
  end
```

Changes in **app/views/layouts/application.html.erb** file:

I have added simple notification navigation in the application layouts.

```
<ul class="nav navbar-nav navbar-right">
  <li class="dropdown">
    <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">Notification (<b class="notifications-count">2</b>)</a>
    <ul class="dropdown-menu notify-drop">
      <div class="notify-drop-title">
        <div class="row">
          <div class="col-md-6 col-sm-6 col-xs-6">Notifications (<b id="notifications-count">2</b>)</div>
          <div class="col-md-6 col-sm-6 col-xs-6 text-right"><a href="" class="rIcon allRead" data-tooltip="tooltip" data-placement="bottom" title="tümü okundu."><i class="fa fa-dot-circle-o"></i></a></div>
        </div>
      </div>
      <!-- end notify title -->
      <!-- notify content -->
      <div class="drop-content" id="notification">
        <% Notfication.all.each do |notification| %>
          <li>
            <div class="col-md-3 col-sm-3 col-xs-3">
```

```
              <%= notification.name %>
            </div>
          </li>
        <% end %>
      </div>
    </ul>
  </li>
</ul>
```

We need to do simple configuration setup in the **cable.yml** file**.**

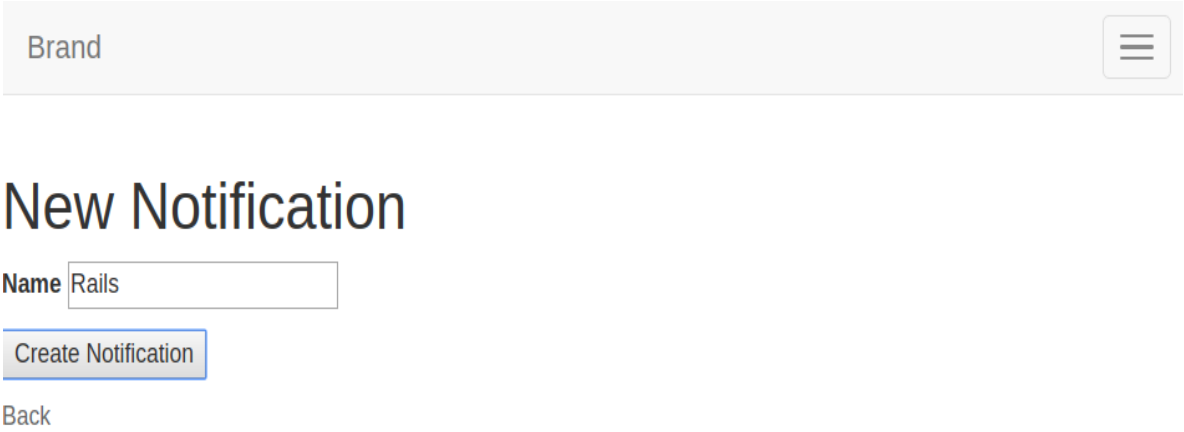
```
redis: &redis
  adapter: redis
  url: redis://localhost:6379/1

production: *redis
development: *redis
test: *redis
```

To start the application first we need to start the redis-server the command to start the redis-server is:

```
redis-server
```

The Redis-server will be started then you can serve your application.

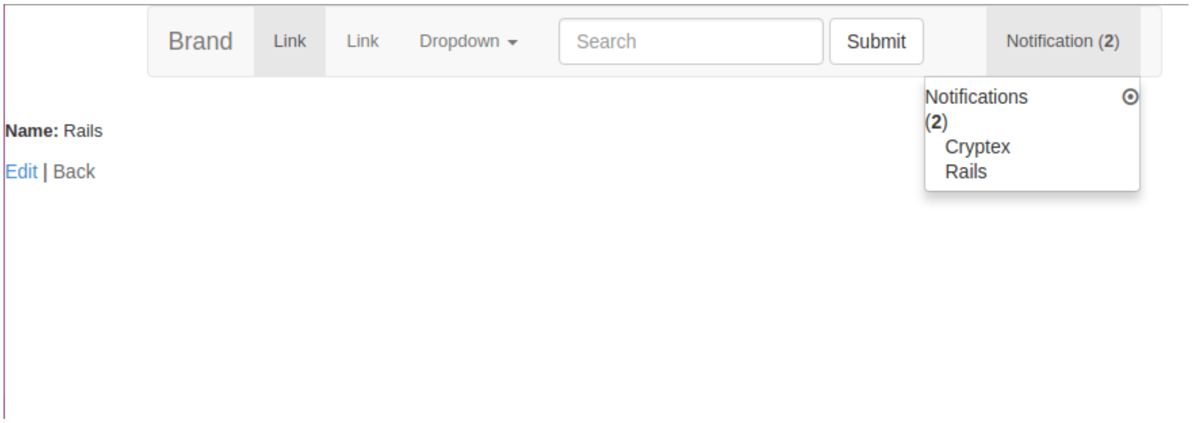
Creating new Notifications :



New Notification

Name Rails

Result will be show like that By using the action cable it will update the **Notification** to **all live users.**



Name: Rails

Notifications (**2**)
Cryptex
Rails

Hire Ruby on Rails web developers, using **Action Cable** we can show the live notifications to all users. This notifications can be of any type like Messaging, Sending push notification etc.