Action Cableで簡易チャッ トを作ってみた

Kazuhiro NISHIYAMA

第87回 Ruby関西 勉強会 2019/07/13 株式会社Ruby開発

自己紹介

- 🔵 西山和広
- 이 Ruby のコミッター
- 🌕 twitter, github など:@znz
- 🬕 株式会社Ruby開発 www.ruby-dev.jp

目的

- 🔵 Ruby 関西中継が止まっていた
 - USTREAM も終了
- 外部サーバーに保存せずにユーザー登録 などなく視聴可能なライブ配信のみした い
- できればチャットもあると良いかも→ Ruby 勉強会なので Rails で

ライブ配信

- YouTube Live
 - 必ず保存されそう (公開するかどうかは選べそう)
 - スマホからの配信はチャンネル登録ユーザー数が増えないとできない
- その他のサービス
 - 視聴にアカウントが必要だったり
 - サービスの主な用途がゲーム配信だったり

nginx-rtmp-module

- 自前ライブ配信サーバが作成可能
- ◎ 録画を残すかどうかも設定次第
- HLS + video.js で視聴可能
 - 試したブラウザー全てで視聴可能
 - Windows, macOS, iOS, Android
 - (Linux は未確認)

チャット

- ライブ配信へのコメント機能
- ◎ 何を使っても良いのなら Firebase が楽そう だった
- Ruby 勉強会なので Action Cable を使って みることに

なぜ Rails 6?

- サンプル的にできるだけデフォルト構成で シンプルに作りたい
- デフォルトが CoffeeScript ではない
 - 新規で採用する理由はあまりない

View の選択

使ってみたかったから

- React (redux なし)
- material-ui 4

環境構築

- ⊚ gem install rails --pre
- 🥯 yarn も入れておく

rails new

- o rails new chat-\$(date +%Y%m%d)
 --webpack=react
- または rails new の後で bin/rails webpacker:install:react

yarn を入れ忘れていたら、後から webpacker:install

埋め込むページ作成

- rails g controller pages index
- routes 変更: root to: 'pages#index'
- app/views/pages/index.html.erb に React の呼び出し埋め込み <%= javascript_pack_tag 'hello_react' %>

channel 作成

- rails g channel chat
- ChatChannel クラスができる
- rails g controller と同様に rails g channel chat speak などでメソッドも生成可能

送受信テスト準備 (Rails 側)

- ChatChannel に def receive(data) を 追加 ActionCable.server.broadcast('chat_channel', data)
- subscribed で stream_from 'chat_channel'

送受信テスト準備 (JS 側)

- ♠ chat_channel.js の received(data)
 に console.log(data);
- JavaScript console から send で送信して 確認

微調整

- ◎ 送信時刻追加
- material-ui で入力欄追加
- ◆ faker を使ってランダムなデフォルトの名前を設定

アイコン表示

- gravatar でアイコン表示
- サーバー側でしかわからない送信元 IP アドレスも使って、同じ名前でも同じアイコンにならないように

モデルなどを作成

- rails g model message name body sent at:timestamp
- rails g job MessageBroadcast
- ◎ broadcast を job 経由に
 - はっきりとした説明を見つけられなかったが、 アプリケーションサーバーが複数台になった時 に received で broadcast せずに job を経由す る必要がありそう

最近のメッセージ表示

◆ hidden_field_tag で to_json した文字
列を埋め込み

- Son.parse(document.getElementById('recent_messages').value)で取り出し
- ちゃんとエスケープされる方法を選択
- あまり良い方法ではないが、開発速度重視

最近の基準

- 1時間以内
- 50件まで
- 🌕 リロードするとここまでになる
- ◎ 開きっぱなしなら無制限に追加していく

送信中メッセージ表示

- 空欄アイコンで表示
- ◎ 空欄じゃないアイコンに変わったら受信完 了

微調整

- 入力欄が空欄の時は送信ボタンを無効化
- IP アドレスとリクエスト ID も保存 → アイコンに反映

デプロイ

● VPS のサーバーにデプロイ

production で動かない

- O Uncaught TypeError: r is not a function で動かない
- https://github.com/rails/rails/issues/ 35501に同じ現象が書いてあったが未解決
- 動かすことを優先して development で動かすことに

動画埋め込み

単独 HTML ファイルで試していた video.js 埋め込み

config.hosts 設定

nginx 設定

- 普通の reverse proxy 設定
- ◎ WebSocket も proxy するように設定
- 🎱 本題ではないので詳細は省略

Cloudflare 設定

- Full (Strict)
 - チャットは完全暗号化
 - ライブ配信の視聴も完全暗号化
- wss (暗号化ありの WebSocket) も問題な く通る

trusted_proxies 設定

- Cloudflare 経由にすると remote_ip が取れなくなったので config.action_dispatch.trusted_p roxies を設定
- https://www.cloudflare.com/ips/
 - https://www.cloudflare.com/ips-v4
 - https://www.cloudflare.com/ips-v6

微調整

- favicon 設定
- title 設定

WireGuard

- WireGuard とは
 - 高速軽量な VPN
 - まだ本番運用には適さない
- ◎ ライブ配信の送信側を暗号化するのに利用
- 🌕 本題ではないので詳細は省略

さらに機能追加

- ◎ 接続しているユーザー一覧アイコン表示
- reload video ボタン追加

まとめ

- Action Cable で簡単にリアルタイム通信が 作成可能
- 環境構築はちょっと面倒 (yarn が必須など)
- ◆ 本番環境で使うには WebSocket が必須な どちょっと制限あり