pixChat

pixChat

これはあまりにも酷いアイデアで

単なる遊びなので

絶対に付けてはいけないぞw

(´・ω・`)

Demo

www.192.168.1.97.xip.io:3000

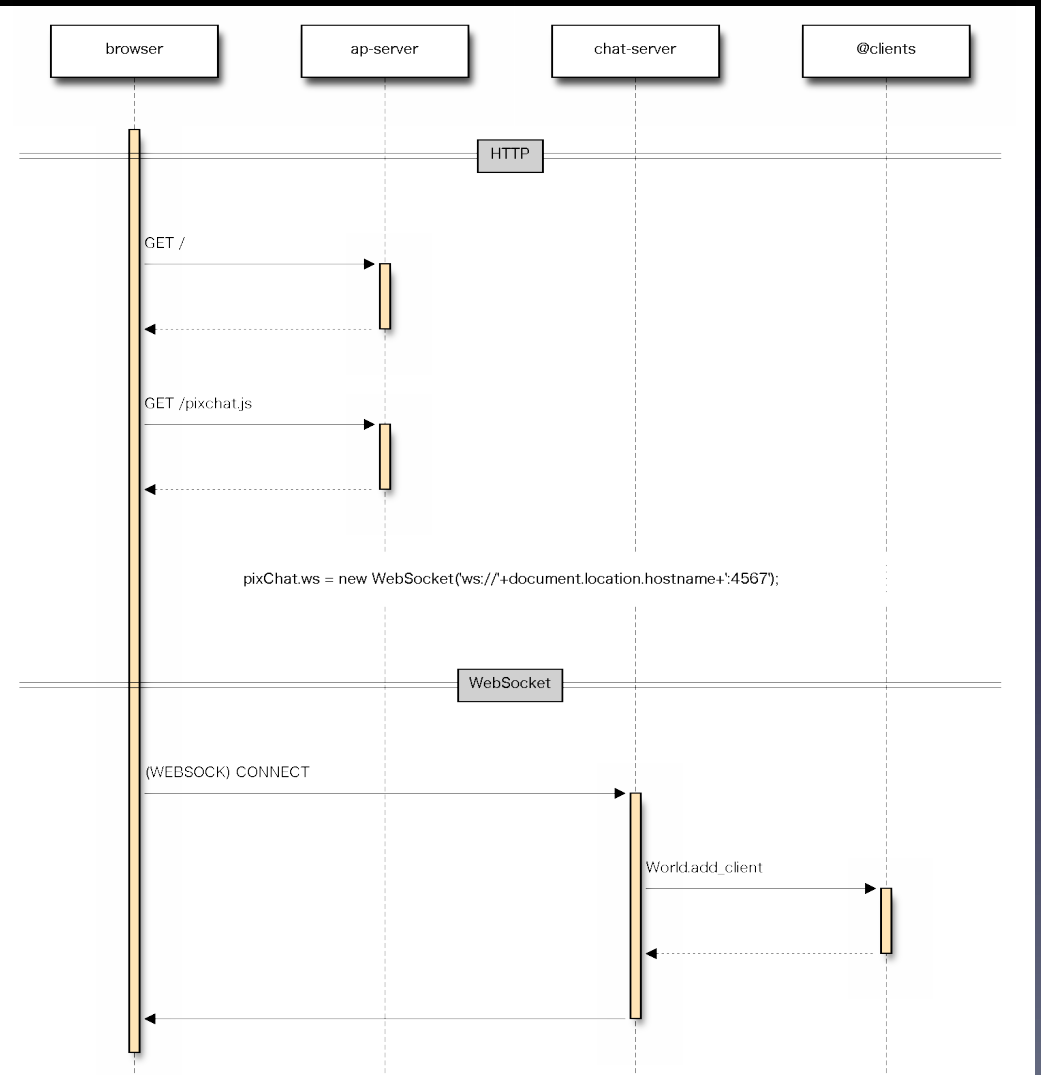
pixiv

How?

- Client: JavaScript + JSON over WebSocket
- Server: sinatra-websocket

pixiv

On Page Load



- Load Page from AP Server
- Included JS Loads
- Connect to Chat-Server via WebSocket
- Chat-Server registers client

JavaScript

```
init: function() {
  pixChat.ws = ws = new WebSocket('ws://'+document.location.hostname+':4567');
  ws.onopen    = function()  { pixChat.show('connecting...'); };
  ws.onclose   = function()  { pixChat.show('disconnected.'); }
  ws.onmessage = function(m) { pixChat.parse(m.data); };
  $(document).on('click', '.pixchat-talker > input[type=button]', pixChat.clickTalk);
},
```

Server

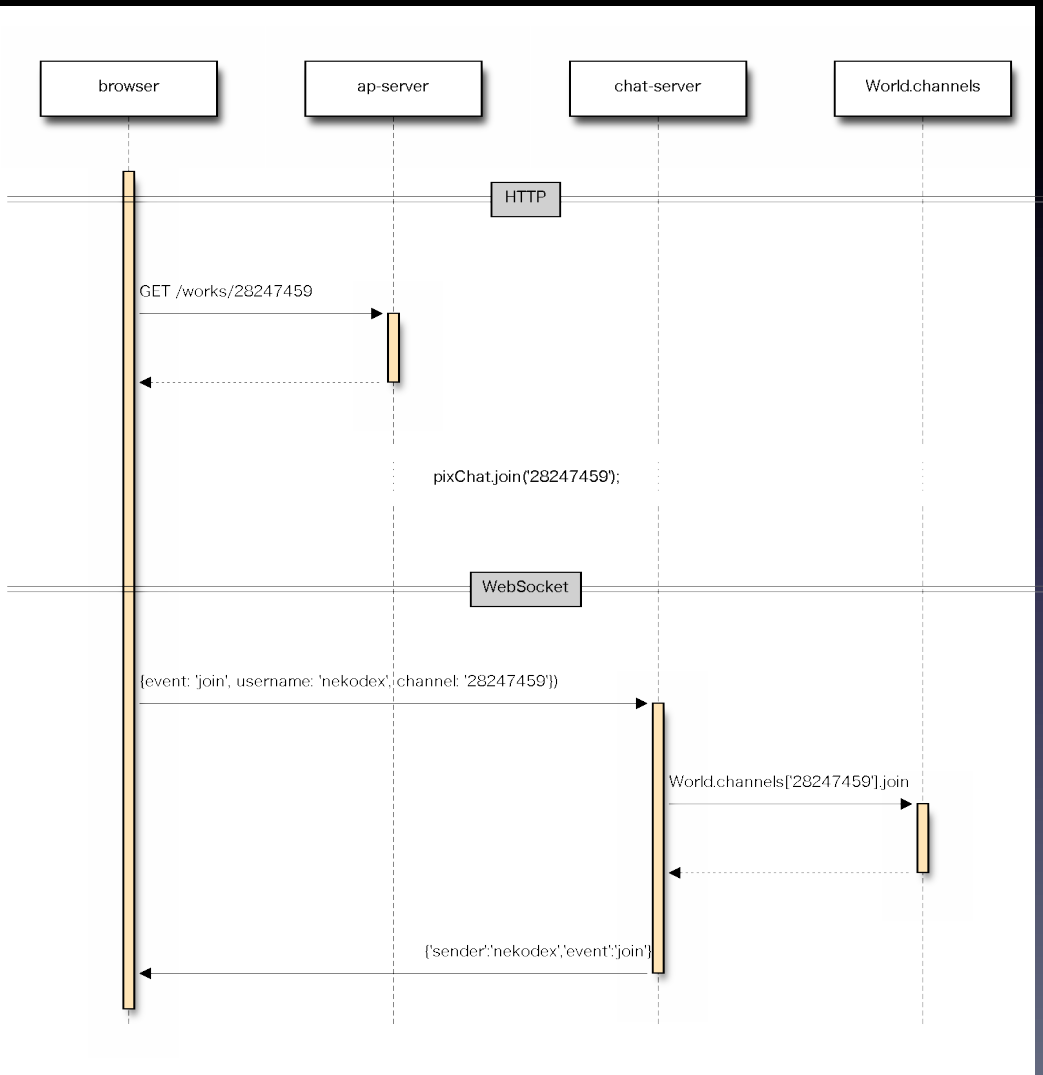
```ruby
#!/usr/bin/env ruby
require 'sinatra'
require 'sinatra-websocket'
require 'json'
require 'cgi' # for CGI.escapeHTML

set :server, 'thin'

$world = World.new

get '/' do
  return unless request.websocket?
  request.websocket do |ws|
    ws.onopen do
      $world.add_client Client.new(ws)
      warn("-> [SOCK] socket connected, #{$world.connected_count} connected")
    end
    ws.onmessage do |data|
      EM.next_tick {
        $world.parse ws, data
      }
    end
    ws.onclose do
      $world.remove_client($world.find_client(ws))
    end
  end
end
```

On Work View



- Load Work from AP Server
- JS triggers channel join
- Chat-Server registers client in channel

JavaScript

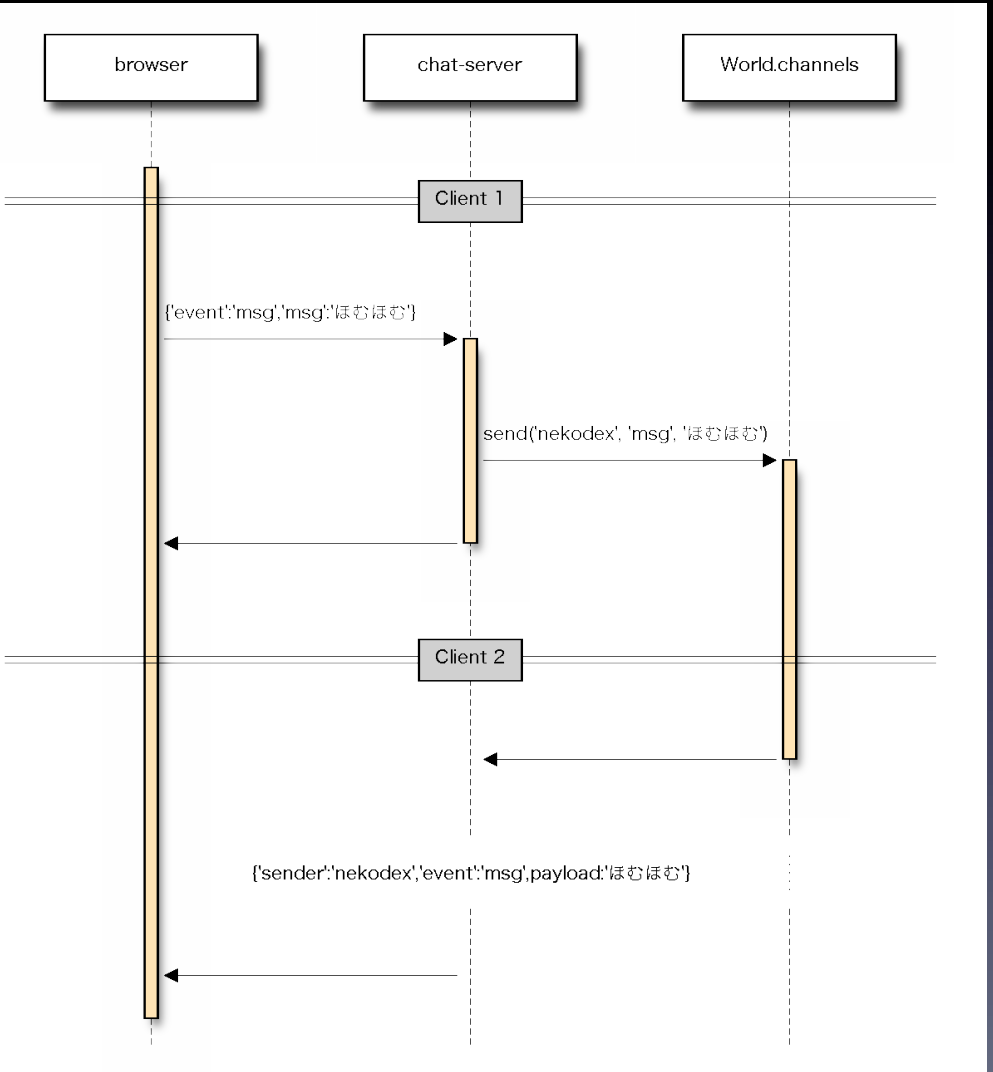
```
join: function(chan) {
  console.log('[pixChat] switching to channel '+chan);
  pixChat.send({event: 'join', username: pixiv.user.username, channel: chan});
  pixChat.channel = chan;
},
```

Server

```
def join(client)
  @clients << client
  client.channel = self.name

  send(client, 'join')
end
```

On Chat



- Message sent over WebSocket
- Chat-Server sends to other clients in same channel
- Chat-Server registers client in channel

JavaScript

```
talk: function(data) {
  pixChat.send({event: 'msg', msg: data});
}
```

Server

```
def send(sender, event, payload=nil)
  warn("[#{@name}:#{event}] <#{sender.username}> #{payload}")
  @clients.each do |client|
    payload = (CGI.escapeHTML(payload) unless payload.nil?)
    client.send({sender: sender.username, event: event, payload: payload}.to_json)
  end
end
```

Questions?

pixChat

Code:

github.com/nekodex/pixChat

Presentation by Jamie Taylor

twitter: @nekodex github: nekodex