

Design Doc: Leisure Moment

Author: Xintong Bai

Github: <https://github.com/mingxinwb/leisure-moment>

Nov 17, 2017

Status: finished

[Overview](#)

[Major Use Cases](#)

[High-level Stack Diagram](#)

[Detailed Design](#)

[Features](#)

[Navbar](#)

[New Leisure Moment](#)

[Leisure Moment List](#)

[Leisure Moment Detail](#)

[UI Design](#)

[Nickname Wrapper before entering home page](#)

[Home Page](#)

[Leisure Monent Detail Page](#)

[Components Models](#)

[Component: Leisure Monent List](#)

[Component: Leisure Monent Detail](#)

[Component: New Leisure Monent](#)

[Component: Conversation Panel](#)

[Component: NavBar](#)

[REST API Process](#)

[Socket IO Communication](#)

[Socket.io at Client-side](#)

[Socket.io at Server-side](#)

Overview

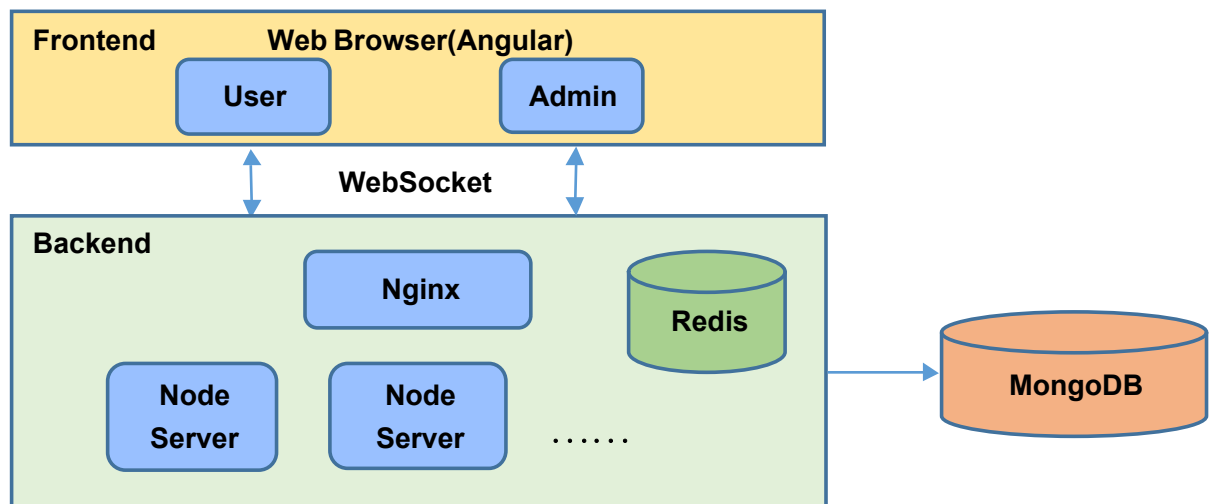
Leisure Moment is a full-stack system supporting a real-time conversation room for each leisurely moment. This document covers the details of the implementation of Leisure Moment Submit and Real-time Conversation from an engineering perspective.

Major Use Cases

1. When entering Leisure Moment, user is supposed to enter a nickname.
2. User can submit new leisure moment. Supported type are text, image and video/audio.
3. User can click one of the leisure moment from the leisure moment list and join its real-time conversation room.
4. User can chat with others through the conversation panel, they can send their comments with unique color, emojis and images to the conversation panel.
5. User can press the heart button as like for any leisure moments.
6. User can browse pre-stored leisure moment list and they can also see the number of likes and comments on each leisure moment.
7. User's chat history will be recorded for reference.
8. Admin can manually manage their account: add a unique icon, manage their posts (CRED) and browse the notification of who likes/comments your leisure moment. They can also press the favorite button for any leisure moments and browse their favorites history later.

High-level Stack Diagram

Stack	Technologies
Frontend - client	Angular.js, Socket.io
Backend - server	Node.js, Socket.io, Redis, MongoDB, Nginx



Detailed Design

Features

Navbar

- Brand Name
- Nickname
- Sign up/Sign in
- Account (dropdown list - only for Admin)
 - Icon
 - Your leisure moments
 - Notification (likes/comments)
 - Favorites

New Leisure Moment

- Enter Leisure Moment Title
- Enter Nickname
- Enter Content: type of Texts, Images, or Video/Audio
- Location (TBC)

Leisure Moment List

For each leisure moment:

- Nickname
- Location (TBC)
- Leisure Moment Title with Conversation Room(moment.id)Link
- Number of Likes with Heart Icon
- Number of Comments with Comment Icon

Leisure Moment Detail

- Nickname
- Location (TBC)
- Leisure Moment Title
- Leisure Moment Content
- Heart Icon Button as Like
- Comments Count Status
- Conversation Panel
 - Online Count Status
 - Conversation History
 - Button of Color, Emoji and Image
 - Send Comment Area
 - Send Button

UI Design

Nickname Wrapper before entering home page

get yourself a nickname :)

Home Page

Leisure Moment

Sign Up/Sign In

Search

Leisure Moment Title

Nickname

Moment Content

Submit

Walt Disney World	It's beginning to look a lot like Christmas.	39 Moments	25 Likes
Yakitori	Party cake design - Soocake.	3 Moments	8 Likes
yellowstonenps	Today's pics.	97 Moments	108 Likes

Leisure Monent Detail Page

Leisure Moment

Sign Up/Sign In

Search

It's beginning to look a lot like Christmas.

jkjklgugui jk jio jl o hyuhydytfd hjnkhuiyhuj kljdntrb ghnjk opi. jklklgugui jk jio jl o hyuhydytfd hjnkhuiyhuj kljdntrb ghnjk opi. jklklgugui jk jio jl o hyuhydytfd hjnkhuiyhuj kljdntrb ghnjk opi. jklklgugui jk jio jl o hyuhydytfd hjnkhuiyhuj kljdntrb ghnjk opi.

@Walt Disney World 25 Likes 39 Moments

Let's Conversation

emoji image clear

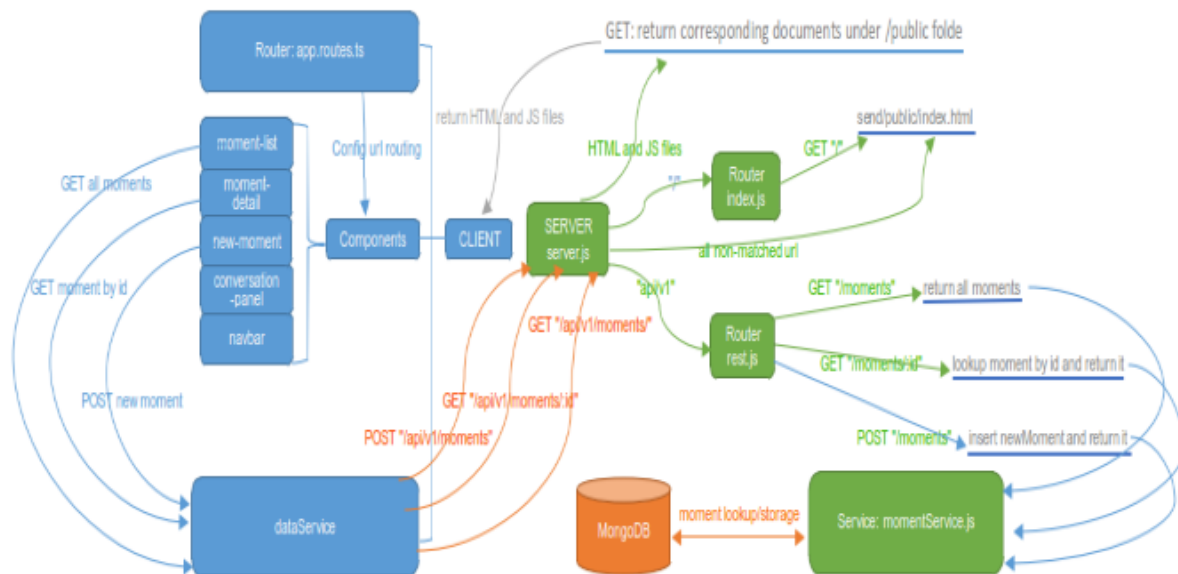
enter to send

SEND

Components Models

- Component: **moment-list** as Leisure Moment List
 - get moment list through Restful API
 - link to moment-detail by click moment
 - pass the information of nickname, moment title, moment id, likes and comments count to moment-detail page
- Component: **moment-detail** as Leisure Monent Detail
 - get one moment by moment id through Restful API
- Component: **new-moment** as New Leisure Monent
 - post newMoment to through Restful API
- Component: **Conversation Panel**
 - connect client-side and server-side
 - initialize socket.io
 - count online users
 - appear system notice: who joined / who left
 - pick font color
 - send comment, emoji and image by perssing send button or just entering the enter button
 - receive other users' comments, emojis and images with their unique color
- Component: NavBar
 - apply for the whole app page

REST API Process



Socket IO Communication

We are using socket.io as the communication protocol between client and server. The reasons are:

- Client-server communication is heavy;
- Full-duplex asynchronous messaging is preferred;
- WebSockets pass through most firewalls without any reconfiguration.

Socket.io at Client-side

Method	Description
io.connect()	connect with server
socket.emit()	send an event to server
socket.on()	listen to connect event from server-side
initSocket()	Initialize socket.io
socket.emit('login', nickname)	send user's nickname to server
socket.on('nameExisted', function())	listen to 'nameExisted' event from server-side, then sent a hint to user: 'nickname is taken, please choose another one.'
socket.on('loginSuccess'), function())	listen to 'loginSuccess' event from server-side, then appear the nickname on navbar, move away the nickname wrapper and focus on commentInput area.
socket.on('system', function(nickname, userCount, type)	listen to 'system' event from server-side, judge the type info to decide what to send in system notice in status area - e.g.: '8 users online; cici joined; bubu left;' the font color will be red.
socket.emit('postComment', comment, color)	send user's comment with color to server-side
socket.on('newComment', function(user, comment))	listen to 'newComment' event from server-side, then appear the comment into history area with user's nickname and comment.
socket.emit('img', img.target.result)	send an image with type of base64 to server-side.
socket.on('newImg', function(user, img, color))	listen to 'newImg' event from server-side and appear the image with user's nickname and unique color into history area.

Socket.io at Server-side

Method	Description
require('socket.io').listen(server)	import socket.io module and bind it to server
socket.on()	receive and handle connect event from client-side
socket.emit()	send connect event to client-side
socket.on('login', function(nickname))	receive and handle login event with nickname:
socket.emit('nameExisted')	check if users.indexOf(nickname) > -1; if yes, then send 'nameExisted' to client-side.
socket.emit('loginSuccess')	socket.userIndex=users.length; socket.nickname=nickname; users.push(nickname); after above step, then send 'loginSuccess' to client-side.

<code>io.sockets.emit('system', nickname, users.length, 'login')</code>	send the nickname to client-side all connected client-side sockets, including user-self.
<code>socket.on('disconnect', function())</code>	splice the disconnected user from users
<code>socket.broadcast.emit('system', socket.nickname, users.length, 'logout')</code>	broadcast to client-side all other connected users except user-self.
<code>socket.on('postComment', function(comment, color))</code>	listen to 'postComment' from client-side
<code>socket.broadcast.emit('newComment', socket.nickname, comment, color)</code>	send 'newComment' event to client-side all other users with user's nickname, color and comment.
<code>socket.on('img', function(imgData))</code>	listen to 'img' event from client-side
<code>socket.broadcast.emit('newImg', socket.nickname, imgData)</code>	send new image with nickname and imgData to client-side all other users.