

# CUMC Badminton Web Service

Team: CUMC Web Services

Zheyuan Liu, Yiru Gong, Brian Lee, Zhuolun Huang, Youyuan Kong

# Project: CUMC Badminton System

A web based application for badminton court appointment, partner finding, chatting, and badminton forum postings.

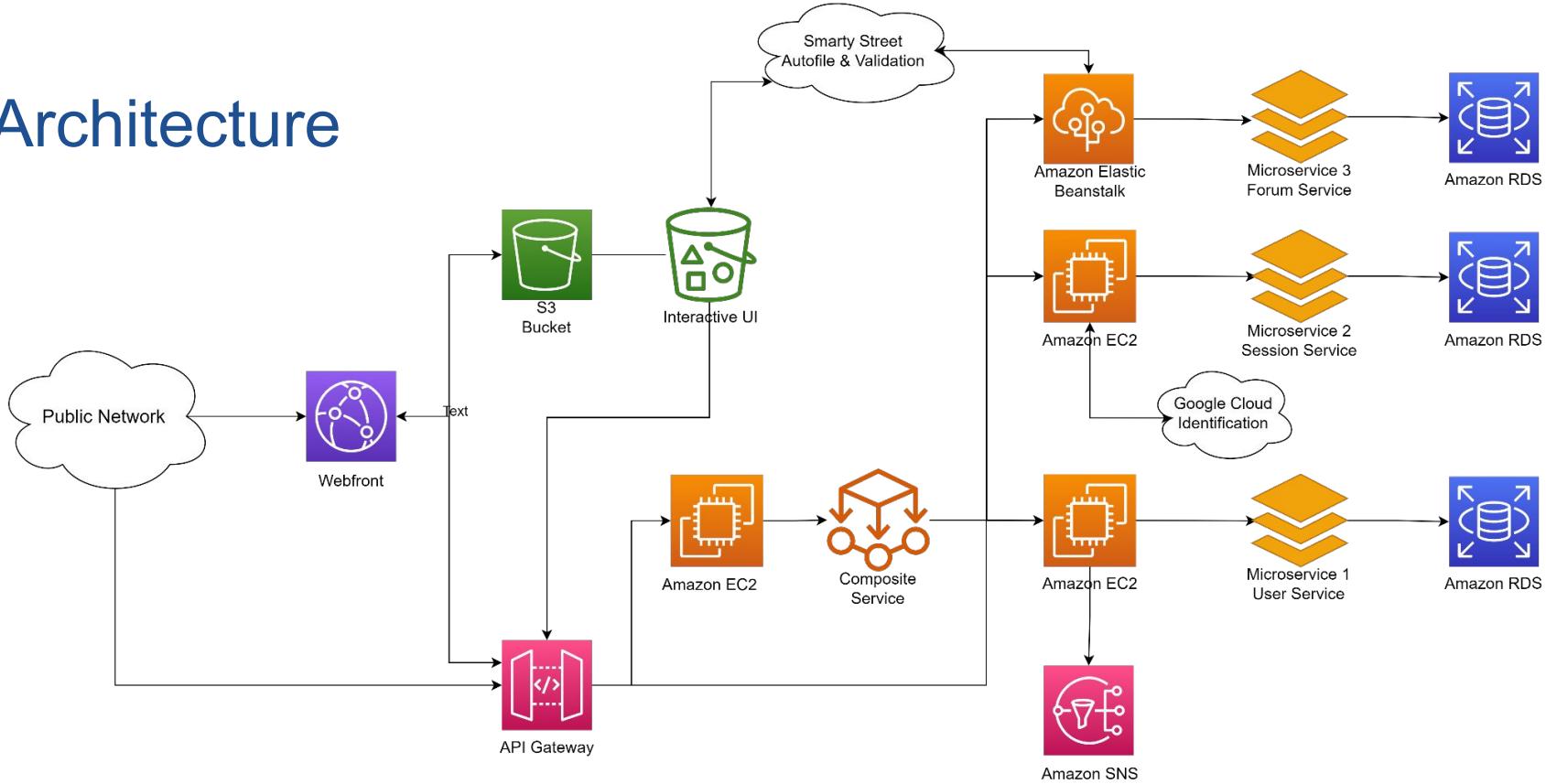
## AWS Applications

- Frontend: CloudFront, S3
- Backend: EC2 and Elastic Beanstalk
- Database: RDS
- API resources: API Gateway
- Composite Microservice
- SNS and IAM

## Frameworks

- Angular TypeScripts
- Python Flask
- MySQL Server

# Architecture

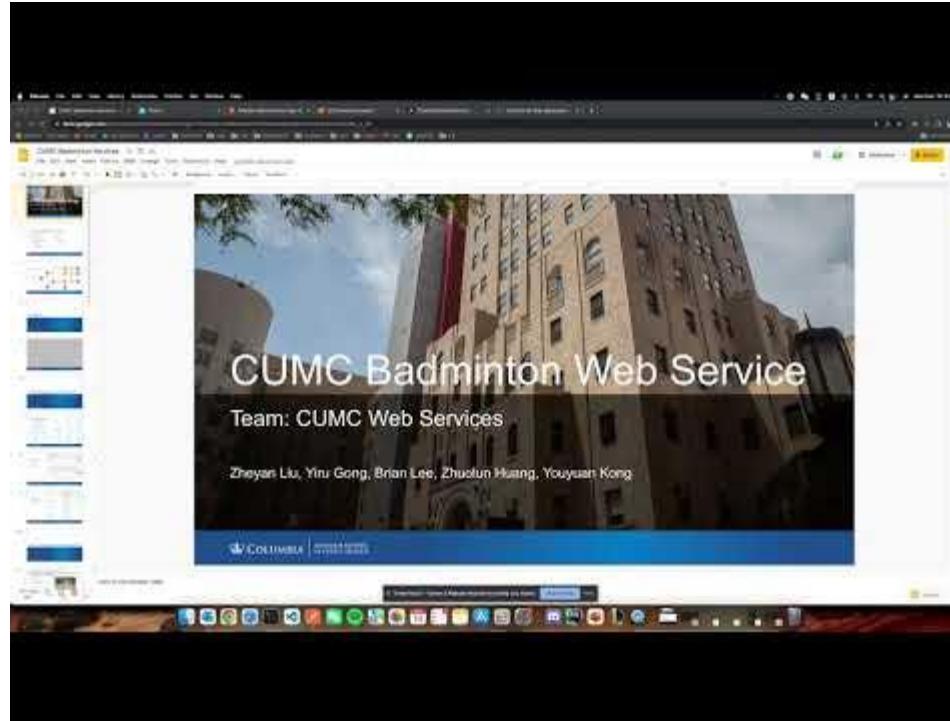


# UI Demo

# Web App cloudfont url

<https://d60lv0um3jt5y.cloudfront.net/>

## Video walkthrough

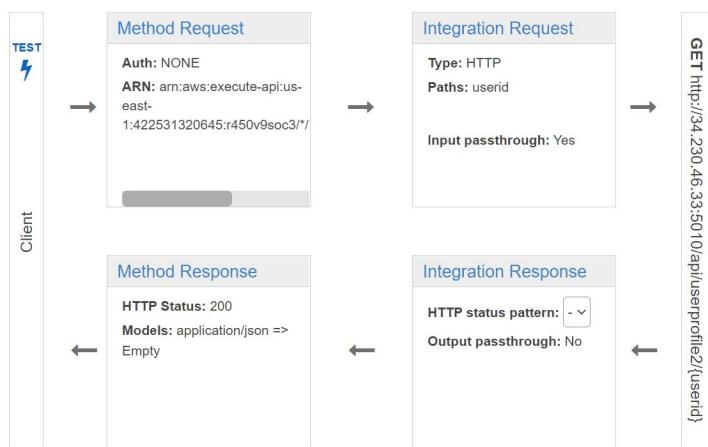


# API Gateway and CloudFront

# API Gateway Paths

- Embedded http Resources
- Enable CORS
- Differentiate from Web URLs
- Deployed with API Stages

/api/userprofile2/{userid} - GET - Method Execution



▼ /	▼ /googlelogin	▼ /user
▼ /api	OPTIONS	▼ /{userid}
▼ /forum	POST	▼ /add_partner
▼ /labels	▼ /session	OPTIONS
GET	OPTIONS	POST
OPTIONS	POST	▼ /chatting
▼ /locations	▼ /user	▼ /history
GET	▼ /{userid}	OPTIONS
OPTIONS	GET	POST
▼ /myposts	OPTIONS	▼ /sending
▼ /user_id	▼ /{sessionid}	OPTIONS
▼ /{user_id}	▼ /enroll	POST
GET	▼ /{userid}	▼ /delete_partner
OPTIONS	OPTIONS	OPTIONS
▼ /newlocation	▼ /quit	POST
OPTIONS	▼ /{userid}	▼ /{userid_to}
POST	GET	GET
▼ /newpost	OPTIONS	OPTIONS
▼ /user_id	▼ /user	GET
▼ /{user_id}	▼ /{userid}	OPTIONS
OPTIONS	GET	OPTIONS
POST	OPTIONS	POST
▼ /user	▼ /{user_id}	▼ /partner
▼ /{userid}	GET	GET
▼ /add_partner	OPTIONS	OPTIONS
▼ /invitation	▼ /{user_id}	GET
GET	OPTIONS	OPTIONS

<https://r450v9soc3.execute-api.us-east-1.amazonaws.com/stage1>

# CloudFront

- Origins
- Behaviors
- Invalidations

E3SHJ08L2UUC28

[View metrics](#)

General    **Origins**    Behaviors    Error pages    Geographic restrictions    Invalidations    Tags

## Origins

[Edit](#)

[Delete](#)

[Create origin](#)

Filter origins by property or value

< 1 >

	Origin name	Origin domain	Origin path	Origin type	Origin Shi...
<input type="radio"/>	r450v9soc3.e...	r450v9soc3.execute-api.us-east-...	/stage1	Custom Origin	-
<input type="radio"/>	e6156-cumc...	e6156-cumc-badminton-web.s3....		S3	-

General    **Origins**    **Behaviors**    Error pages    Geographic restrictions    Invalidations    Tags

## Behaviors

[Save](#)

[Move up](#)

[Move down](#)

[Edit](#)

[Delete](#)

[Create behavior](#)

Filter behaviors by property or value



	Preced...	Path pattern	Origin or o...	Viewer pro...	Cache poli...	Origin req...	Response ...
<input type="radio"/>	0	/index.html/*	e6156-cum...	Redirect H...	Managed-Cachin	-	-
<input type="radio"/>	1	/stage1/*	r450v9soc...	Redirect H...	Managed-Cachin	-	-
<input type="radio"/>	2	Default (*)	e6156-cum...	Redirect H...	Managed-Cachin	-	-

<https://d60lv0um3jt5y.cloudfront.net>

# S3 Buckets

- Bucket policy
- Bucket CORS

## Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket.

[Edit](#) [Delete](#)

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "AddPerm",  
            "Effect": "Allow",  
            "Principal": "*",  
            "Action": "s3:GetObject",  
            "Resource": "arn:aws:s3:::e6156-cumc-badminton-web/*"  
        }  
    ]  
}
```

## Cross-origin resource sharing (CORS)

The CORS configuration, written in JSON, defines a way for client web applications that are loaded in one

```
{  
    "AllowedHeaders": [  
        "*"  
    ],  
    "AllowedMethods": [  
        "PUT",  
        "POST",  
        "DELETE"  
    ],  
    "AllowedOrigins": [  
        "http://e6156-cumc-badminton-web.s3-website-us-east-1.amazonaws.com"  
    ],  
    "ExposeHeaders": [  
        "x-amz-server-side-encryption",  
        "x-amz-request-id",  
        "x-amz-id-2"  
    ],  
    "MaxAgeSeconds": 3000  
}
```

Amazon S3 > Buckets > e6156-cumc-badminton-web

## e6156-cumc-badminton-web [Info](#)

Publicly accessible

[Objects](#) [Properties](#) [Permissions](#) [Metrics](#) [Management](#) [Access Points](#)

### Objects (18)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory [\[?\]](#) to get a list of all objects in you

[C](#) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#)

[Find objects by prefix](#)

<input type="checkbox"/>	Name	Type
<input type="checkbox"/>	<a href="#">3rdpartylicenses.txt</a>	txt
<input type="checkbox"/>	<a href="#">assets/</a>	Folder
<input type="checkbox"/>	<a href="#">favicon.ico</a>	ico
<input type="checkbox"/>	<a href="#">google-logo.702aadcc5aa45a.png</a>	png
<input type="checkbox"/>	<a href="#">index.html</a>	html
<input type="checkbox"/>	<a href="#">main.09c820d375617420.js</a>	js
<input type="checkbox"/>	<a href="#">main.2c5c6ce45e15c6eb.js</a>	js
<input type="checkbox"/>	<a href="#">main.3ec090096ef8bdee.js</a>	js
<input type="checkbox"/>	<a href="#">main.5ae4bd8747692c44.js</a>	js
<input type="checkbox"/>	<a href="#">main.60a770615da0bc6f.js</a>	js
<input type="checkbox"/>	<a href="#">main.954053ec07d9eb35.js</a>	js
<input type="checkbox"/>	<a href="#">main.c6acaa24760aacd0.js</a>	js
<input type="checkbox"/>	<a href="#">main.ee33c6f79adeaf19.js</a>	js
<input type="checkbox"/>	<a href="#">polyfills.4ed20f14c8d5f068.js</a>	js
<input type="checkbox"/>	<a href="#">polyfills.d69ae09104006704.js</a>	js
<input type="checkbox"/>	<a href="#">runtime.696fb306f7122e5c.js</a>	js
<input type="checkbox"/>	<a href="#">styles.31f2520e1ebd6e10.css</a>	css
<input type="checkbox"/>	<a href="#">styles.9f2f217cc4cbc796.css</a>	css

# MS1:Chatting and Partner Service

# Chatting service and partner service: Check profile, send/accept partner request, check chat history and send message

- Check partner status
- Delete partner
- Search users by their email
- Send partner request
- Accept/reject partner request
- Check chat history
- Send message



Beanstalk: E6156ms1-env.eba-jjys9bmm.us-east-1.elasticbeanstalk.com

# DB design

invitations	chatting_form	partners
<ul style="list-style-type: none"><li>userid_from int</li><li>userid_to int</li><li>Content longtext</li><li>response tinyint(1)</li><li>Invitation_ID int</li></ul>	<ul style="list-style-type: none"><li>userid_from int</li><li>userid_to int</li><li>Content longtext</li><li>Time datetime</li><li>Chat_ID int</li></ul>	<ul style="list-style-type: none"><li>Userid_to int</li><li>Userid_from int</li></ul>

# Partner Service: Check profile, send/accept partner request

COLUMBIA BADMINTON SERVICES

Home Sessions ▾ Forums ▾ Profile ▾ Chat ▾

Logout

My partner

View my partner

View invitations

Userid from:	Username:	Email:	Content:	Add/Reject:
5	banana	test1@test.com	Hello.77?	Add Reject
1	yk888	test5@test.com	Hello, how are you doing?	Add Reject
12	KK	keviantk@gmail.com	try to be a good partner	Add Reject

Search Profile

Email(any part of email): com

Search number: 5

check profile

Userid:	Email:	Username:	Sex:	Preference	Credits	Birthday	Partner number	Send Invition
1	test5@test.com	yk888	Male	Double	90	2011-02-02	2	Invite
2	test4@test.com	zebra	Female		60	2014-01-01	1	Invite
3	test3@test.com	orange	Female		0	2015-01-01	4	Invite
4	test2@test.com	apple	Male		40	2017-01-01	3	Invite
5	test1@test.com	banana	Male		100	2022-01-01		Invite

# API Demo(EC2)

## Send invitation to be a partner

6156-MS1 / Send a New Invitation

GET http://44.204.161.68:5010/api/user/2/partner/invitation

Params Authorization Headers (6) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

This request does not have a body

Body Cookies Headers (6) Test Results

Pretty Raw Preview Visualize JSON

```
1 "success": true,
2 "data": [
3   {
4     "Invitation_ID": 1,
5     "userid_from": 1,
6     "userid_to": 2,
7     "Content": "Hello, how are you doing?",
8     "response": 0
9   },
10   {
11     "Invitation_ID": 6,
12     "userid_from": 6,
13     "userid_to": 2,
14     "Content": "Hello,66?",
15     "response": 0
16   },
17   {
18     "Invitation_ID": 9,
19     "userid_from": 1,
20     "userid_to": 2,
21     "Content": "Hello, how are you doing?",
22     "response": 0
23 }
```

Cookies

## Get my partner info

6156-MS1 / get\_partner

GET http://44.204.161.68:5010/api/user/9/partner

Params Authorization Headers (6) Body Pre-request Script

Query Params

KEY	VALUE
Key	Value

Body Cookies Headers (6) Test Results

Pretty Raw Preview Visualize JSON

```
1 "success": true,
2 "data": [
3   {
4     "userid": 1,
5     "email": "test@test.com",
6     "username": "yk888",
7     "sex": "Female",
8     "preference": "Single",
9     "credits": 90,
10    "birthday": "2011-02-09",
11    "partner": 2
12  },
13  {
14    "userid": 2,
15    "email": "test4@test.com",
16    "username": "zebra",
17    "sex": "Female",
18    "preference": null,
19    "credits": 60,
20    "birthday": "2014-01-01",
21  }
22 ]
```

## Search profile

6156-MS1 / search users\_by\_email

POST http://44.204.161.68:5010/api/user/1/search\_pro

Params Authorization Headers (8) Body Pre-request Script

none form-data x-www-form-urlencoded raw binary GraphQL

```
1 ...
2 ...
3 ...
4 ...
```

Body Cookies Headers (6) Test Results

Pretty Raw Preview Visualize JSON

```
1 "success": true,
2 "data": [
3   {
4     "userid": 1,
5     "email": "test5@test.com",
6     "username": "yk888",
7     "sex": "Female",
8     "preference": "Single",
9     "credits": 90,
10    "birthday": "2011-02-09",
11    "partner": 2
12  },
13  {
14    "userid": 2,
15    "email": "test4@test.com",
16    "username": "zebra",
17    "sex": "Female",
18    "preference": null,
19    "credits": 60,
20    "birthday": "2014-01-01",
21  }
22 ]
```

# API Demo(EC2)

Get chatting history

Get invitation not responded

Sending Message

POST <http://44.204.161.68:5010/api/user/12/chatting/history>

Body

```
1 ... "userid_to":1
2
3
```

Body Cookies Headers (6) Test Results

Pretty Raw Preview Visualize JSON

```
1
2   "success": true,
3   "data": [
4     {
5       "Chat_ID": 1,
6       "userid_from": 1,
7       "userid_to": 12,
8       "Content": "Hello, how are you doing?",
9       "Time": "2022-10-16 21:35:14",
10      "my_username": "KK",
11      "its_username": "yk888"
12    },
13    {
14      "Chat_ID": 5,
15      "userid_from": 1,
16      "userid_to": 12,
17      "Content": "all along the watchtowers",
18      "Time": "2022-12-19 04:06:53",
19      "mv_username": "KK".

```

GET <http://44.204.161.68:5010/api/user/2/partner/invitation>

Body

This request does not have a body.

Body Cookies Headers (6) Test Results

Pretty Raw Preview Visualize JSON

```
1
2   "success": true,
3   "data": [
4     {
5       "userid_from": 1,
6       "content": "Hello, how are you doing?",
7       "email": "test5@test.com",
8       "username": "yk888"
9     },
10    {
11      "userid_from": 3,
12      "content": "Hello2?",
13      "email": "test3@test.com",
14      "username": "orange"
15    },
16    {
17      "userid_from": 4,
18      "content": "Hello3?",
19      "email": "test4@test.com",
20      "username": "purple"
21    }
22  ]
23
```

POST <http://44.204.161.68:5010/api/user/1/chatting/sending>

Body

```
1
2   "userid_to":2,
3   "content": "hello"
4
```

Body Cookies Headers (6) Test Results

Pretty Raw Preview Visualize JSON

```
1
2   "success": true,
3   "message": "sent it successfully!"
4
```

# Chatting Service: Retrieve Past Messages/Send New Message

CUMC Badminton Services

Home Sessions ▾ Forums ▾ Profile ▾ Chat ▾

 Logout

My chat history

UserID: 12  
Userid\_to: 1

**Find chat history**

Message from yk888 to KK  
Hello, how are you doing?  
2022-10-16 21:35:14

Message from yk888 to KK  
all along the watchtowers  
2022-12-19 04:06:53

Message from KK to yk888  
chatting  
2022-12-19 04:49:58

Message from KK to yk888  
try  
2022-12-19 04:50:52

Message from KK to yk888  
sending to 1  
2022-12-19 04:55:08

Message from KK to yk888  
try  
2022-12-19 05:00:46

Input(ID number): 1

Chat: hello

**Send**

# EC2 Deployment

Set up environment variable in bash\_profile

```
# .bash_profile
# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/.local/bin:$HOME/bin
export PATH
export aws secret access key=OVL3rpYtqJlch1IeVUlj+jK7858bW7g+y73eMtII
export DBHOST=ms1db.ceqs4fzqr3cn.us-east-1.rds.amazonaws.com
export DBPW=Kevinsekai232323**
export DBUSER=root
export KEY_ID=AKIA473A3FFIWE67K6VD
export MS1_URL=http://54.167.191.86:5011/
export PORT=3306
export PYTHONUNBUFFERED=1
export region=us-east-1
export TopicARN=arn:aws:sns:us-east-1:89301943329:
export MS2_URL=http://54.167.191.86:5011/
```

MS2 EC2 instance running on port 5010

```
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5010
 * Running on http://172.31.91.64:5010
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 480-896-267
```

i-0be11f9dbcc19f505 (6156\_MS1)

Public IPs: 44.204.161.68 Private IPs: 172.31.91.64

# Middleware(SNS)

## 1. Use IAM and boto3 to implement SNS

```
import boto3
import os
from botocore.exceptions import ClientError

# Keviant
class SNS:
    logger = logging.getLogger(__name__)

    sns_client = boto3.client('sns',
        region_name=os.environ.get("region"),
        aws_access_key_id=os.environ.get("key_id"),
        aws_secret_access_key=os.environ.get("aws_secret_access_key")
    )
```

## 2. Create, subscribe topics and publish message

```
@app.after_request
def after_decorator(rsp):
    print('After request I should do...')
    for each in SNS.list_topics(SNS.sns_client, SNS.logger)['Topics']:
        if(TopicARN==each['TopicArn']):
            SNS.sns_client.publish(TopicArn=TopicARN,
                Message=content,
                Subject="Notification from the Badminton Club")

    created = True
    if created == False:
        topic_name=SNS.create_topic(SNS.sns_client, SNS.logger, str(id))
        response = SNS.subscribe(SNS.sns_client, SNS.logger, TopicARN, "email", email)
        print(response)
```

## 3. Explore the created topic

The screenshot shows the AWS SNS Topics page. A topic named '12' is selected. The 'Details' section shows the topic's name ('12'), ARN ('arn:aws:sns:us-east-1:893019433329:12'), owner ('893019433329'), and type ('Standard'). Below the details, there are tabs for 'Subscriptions', 'Access policy', 'Data protection policy', 'Delivery retry policy (HTTP/S)', 'Delivery status logging', and 'Encryption'. The 'Subscriptions' tab is active, showing one confirmed subscription for 'keviant@gmail.com' with ID '2d33d090-855d-4bfe-8892-085c130392f'.

## 4. Show the email

The screenshots show three separate emails from 'AWS Notifications' with subject lines 'Notification from the Badminton Club'. Each email contains a link to a partner invitation page. The first two emails are from 'AWS Notifications' and the third is from 'keviant@gmail.com'. The bottom of each email includes standard reply and forward buttons.

# MS2: Auth and Session Service

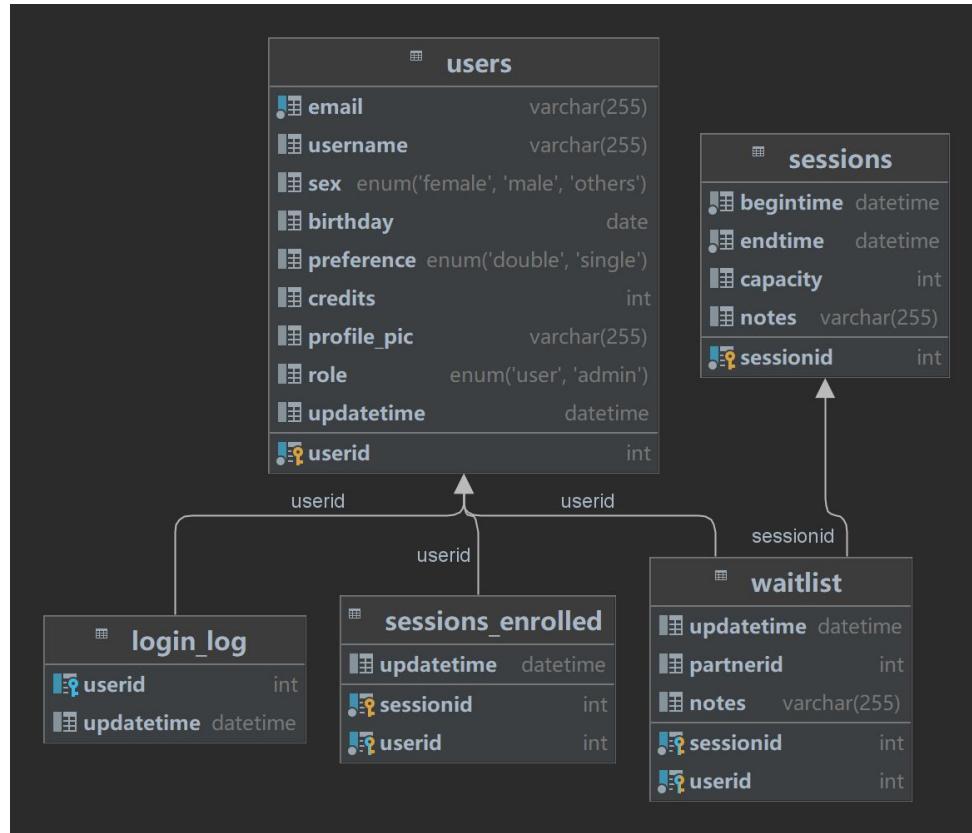
# Auth service and Session management: Log in, Join/Quit session waitlist, Approve to session

- Log in with Google account
- Get/Edit userprofile
- Show available sessions
- Join and quit session waitlist
- (Admin feature) Approve user to session  
and remove them from waitlist



Beanstalk: E6156ms3-env.eba-jjys9bmm.us-east-1.elasticbeanstalk.com

# DB design



# Google Oauth

The image shows two screenshots related to Google OAuth configuration.

**Left Screenshot: Google API Client Configuration**

This screenshot shows the "Client ID for Web application" section of the Google API console. It includes:

- Name:** E6156 course project
- Description:** The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.
- Authorized JavaScript origins:** A list of URLs where the client can be used:
  - https://127.0.0.1:5000
  - http://127.0.0.1:5000
  - https://127.0.0.1:5011
  - https://d50lv0um3t5y.cloudfront.net
  - http://localhost:4200
- Authorized redirect URIs:** A list of URLs where the client can be redirected after authentication:
  - http://127.0.0.1:5000/login/callback
  - http://127.0.0.1:5000/callback
  - http://127.0.0.1:5011/api/login/callback
  - https://54.167.191.86:5011/api/login/callback
  - https://d50lv0um3t5y.cloudfront.net

**Right Screenshot: Google Sign-in Dialog**

This screenshot shows a "Sign in with Google" dialog box. It includes:

- Sign in with Google** button
- Sign in - Google Accounts** window title
- Email or phone** input field
- Forgot email?** link
- Create account** button
- Next** button
- English (United States)**, **Help**, **Privacy**, and **Terms** links at the bottom

# API demo

## Google login call back

6156-MS2 / auth / Google Login

POST https://r450v9soc3.execute-api.us-east-1.amazonaws.com/stage1/api/googlelogin

Params Authorization Headers (9) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

1 `email": "test91@test91.com", "name": "test91", "picture": "fake picture"`

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1
2
3   "data": {
4     "birthday": null,
5     "credits": 100,
6     "email": "test91@test91.com",
7     "preference": null,
8     "profile.pic": "fake picture",
9     "role": "User",
10    "sex": null,
11    "updatedate": "Sun, 18 Dec 2022 07:53:56 GMT",
12    "userid": 14,
13    "username": "test91"
14  },
15  "message": "Hi test91, you have logged in",
16  "success": true
17
```

## Join waitlist as individual/partners

6156-MS2 / session\_waitlist / join session waitlist

POST http://54.167.191.86.5011/api/session/2/enroll/3

Params Authorization Headers (9) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL Text

1

Body Cookies Headers (6) Test Results

Pretty Raw Preview Visualize JSON

```
1
2
3   "success": false,
4   "message": "Failed. Your partner or you have already joined this waitlist"
```

## Quit waitlist

6156-MS2 / session\_waitlist / quit session waitlist

GET http://54.167.191.86.5011/api/session/4/quit/1

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE
Key	Value

Body Cookies Headers (6) Test Results

Pretty Raw Preview Visualize JSON

```
1
2
3   "success": false,
4   "message": "You are not in the waitlist"
```

# UI demo

## Join waitlist

Outlook Gmail Colab Google

localhost:4200 says You have joined the waitlist

OK

CUMC Badminton Services

Home Sessions Forums Profile

Available Sessions

Choose how you want to join the waitlist:

Register for myself  
 Register as partners

Begin time	End time	Capacity	Waitlist size	Notes	Register
2022-12-20 18:30:00	2022-12-20 19:30:00	6	2		<a href="#">Register</a>
2022-12-21 18:30:00	2022-12-21 19:30:00	6	5		<a href="#">Register</a>
2022-12-22 18:30:00	2022-12-22 19:30:00	6	2		<a href="#">Register</a>
2022-12-23 18:30:00	2022-12-23 19:30:00	6	3		<a href="#">Register</a>
2022-12-27 18:30:00	2022-12-27 19:30:00	6	2		<a href="#">Register</a>
2022-12-28 18:30:00	2022-12-28 19:30:00	6	1		<a href="#">Register</a>
2022-12-29 18:30:00	2022-12-29 19:30:00	6	7		<a href="#">Register</a>
2022-12-29 19:30:00	2022-12-29 20:30:00	6	4		<a href="#">Register</a>
2023-01-19 19:30:00	2023-01-19 20:30:00	6	4		<a href="#">Register</a>
2023-01-20 19:30:00	2023-01-20 20:30:00	6	1		<a href="#">Register</a>
2023-01-29 19:30:00	2023-01-29 20:30:00	6	1		<a href="#">Register</a>
2023-01-30 19:30:00	2023-01-30 20:30:00	6	1		<a href="#">Register</a>

## Quit waitlist

Outlook Gmail Colab Google

localhost:4200 says You have quit the waitlist

OK

CUMC Badminton Services

Home Sessions Forums Profile

Enrolled session

Begin time	End time	Capacity	Enrolled	Notes
2023-02-20 19:30:00	2023-02-20 20:30:00	6	1	
2023-02-14 19:30:00	2023-02-14 20:30:00	6	1	
2023-03-20 19:30:00	2023-03-20 20:30:00	6	1	

Waitlist

Begin time	End time	Capacity	Wait size	Notes	Quit
2022-12-21 18:30:00	2022-12-21 19:30:00	6	5		<a href="#">Quit</a>
2022-12-23 18:30:00	2022-12-23 19:30:00	6	4		<a href="#">Quit</a>
2022-12-27 18:30:00	2022-12-27 19:30:00	6	2		<a href="#">Quit</a>
2022-12-29 19:30:00	2022-12-29 20:30:00	6	4		<a href="#">Quit</a>
2023-01-29 19:30:00	2023-01-29 20:30:00	6	1		<a href="#">Quit</a>

## Show enrolled and waitlist

ReCUMC Badminton Services

Home Sessions Forums Profile Chat

Logout

Enrolled session

Begin time	End time	Capacity	Enrolled	Notes
2023-02-20 19:30:00	2023-02-20 20:30:00	6	1	
2023-02-14 19:30:00	2023-02-14 20:30:00	6	1	
2023-03-20 19:30:00	2023-03-20 20:30:00	6	1	

Waitlist

Begin time	End time	Capacity	Wait size	Notes	Quit
2023-01-10 18:30:00	2023-01-10 19:30:00	6	4		<a href="#">Quit</a>
2023-01-28 18:30:00	2023-01-28 19:30:00	6	6		<a href="#">Quit</a>
2023-02-11 19:30:00	2023-02-11 20:30:00	6	2		<a href="#">Quit</a>

# API demo

## Show available sessions

6156-MS2 / session\_waitlist / show available sessions

POST https://i450v9soc3.execute-api.us-east-1.amazonaws.com/stage1/api/session

Body

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
```

Pretty Raw Preview Visualize JSON

```
{"success": true,
"data": [
  {
    "sessionid": 1,
    "beginime": "2022-12-20 18:30:00",
    "endtime": "2022-12-20 19:30:00",
    "notes": null,
    "capacity": 8,
    "enrolled": 2,
    "is_registered": 1
  },
  {
    "sessionid": 2,
    "beginime": "2022-12-21 18:30:00",
    "endtime": "2022-12-21 19:30:00",
    "notes": null,
    "capacity": 8,
    "enrolled": 5,
    "is_registered": 1
  },
  {
    "sessionid": 3,
    "beginime": "2022-12-22 18:30:00",
    "endtime": "2022-12-22 19:30:00",
    "notes": null,
    "capacity": 8,
    "enrolled": 3,
    "is_registered": 0
  },
  {
    "sessionid": 4,
    "beginime": "2022-12-23 18:30:00",
    "endtime": "2022-12-23 19:30:00",
    "notes": null
  }
]}
```

## Show approved sessions

6156-MS2 / session\_waitlist / show approved session by user

GET http://127.0.0.1:5011/api/session/approved/user/3

Body

```
This request does not have a body.
```

Pretty Raw Preview Visualize JSON

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
```

```
{"success": true,
"message": "Here is your approved sessions",
"data": [
  {
    "sessionid": 1,
    "beginime": "2022-12-20 18:30:00",
    "endtime": "2022-12-20 19:30:00",
    "capacity": 8,
    "approved_cnt": 2,
    "notes": null
  },
  {
    "sessionid": 3,
    "beginime": "2022-12-22 18:30:00",
    "endtime": "2022-12-22 19:30:00",
    "capacity": 8,
    "approved_cnt": 3,
    "notes": null
  },
  {
    "sessionid": 4,
    "beginime": "2022-12-23 18:30:00",
    "endtime": "2022-12-23 19:30:00",
    "capacity": 8,
    "approved_cnt": 4,
    "notes": null
  }
]}
```

# Middleware and admin features

If the user is admin, it will randomly approve user from waitlist

The screenshot shows a Postman request for 'Admin session waitlist approve' at `http://127.0.0.1:5011/api/admin/session/approve/5`. The 'Body' tab is selected, containing the JSON payload:

```
1 {  
2   "success": true,  
3   "message": "Following users joined the session: [1, 9, 10]"  
4 }
```

The response body is also JSON, indicating success:

```
1 {  
2   "success": true,  
3   "message": "Following users joined the session: [1, 9, 10]"  
4 }
```

Otherwise, it will raise error

The screenshot shows a Postman request for 'Admin session waitlist approve' at `http://127.0.0.1:5011/api/admin/session/approve/5`. The 'Body' tab is selected, containing the JSON payload:

```
1 {  
2 }
```

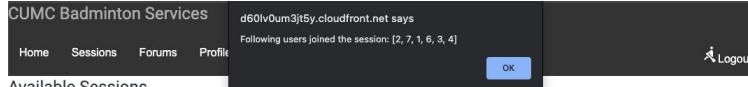
The response body is an HTML error page with status 500:

```
1 <!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">  
2 <html>  
3 <head>  
4   <title>NameError: You are not an admin // Werkzeug Debugger</title>  
5   <link rel="stylesheet" href=?_debugger_&yes&cmd=resource&f=style.css" type="text/  
6   <!-- We need to make sure this has a favicon so that the debugger does  
7   not accidentally trigger a request to /favicon.ico which might  
8   change the application's state. -->  
9   <link rel="shortcut icon" href=?_debugger_&yes&cmd=resource&f=console.png">  
10  <script src=?_debugger_&yes&cmd=resource&f=debugger.js"></script>  
11  <script type="text/javascript">  
12    var TRACEBACK = 4449744160,  
13        CONSOLE_MODE = false.  
14 
```

Implementation using before and after request

```
@app.before_request  
def before_decorator():  
    print('Before request I should do...')  
    # Verify it is an admin or a user  
    print(request.url_rule)  
    if str(request.url_rule)[:11] == "/api/admin/":  
        if not C05Resource.if_admin(int(request.data)):  
            raise NameError('You are not an admin')  
  
@app.after_request  
def after_decorator(rsp):  
    print('After request I should do...')  
    print(rsp.data)  
    return rsp
```

# UI demo



## Available Sessions

Choose how you want to join the waitlist

- Register for myself
- Register as partners

Hi admin, you can approve people from waitlist to session.

Begin time	End time	Capacity	Waitlist size	Notes	Register	Approve
2023-01-10 18:30:00	2023-01-10 19:30:00	6	2		<button>Register</button>	<button>Approve</button>
2023-01-27 18:30:00	2023-01-27 19:30:00	6	4		<button>Register</button>	<button>Approve</button>
2023-01-28 18:30:00	2023-01-28 19:30:00	6	5		<button>Register</button>	<button>Approve</button>
2023-01-29 18:30:00	2023-01-29 19:30:00	6	7		<button>Register</button>	<button>Approve</button>
2023-01-29 19:30:00	2023-01-29 20:30:00	6	5		<button>Register</button>	<button>Approve</button>
2023-02-11 19:30:00	2023-02-11 20:30:00	6	7		<button>Register</button>	<button>Approve</button>
2023-02-20 19:30:00	2023-02-20 20:30:00	6	0		<button>Register</button>	<button>Approve</button>
2023-02-14 19:30:00	2023-02-14 20:30:00	6	0		<button>Register</button>	<button>Approve</button>
2023-03-20 19:30:00	2023-03-20 20:30:00	6	0		<button>Register</button>	<button>Approve</button>

## Admin approve session

# EC2 Deployment

Set up environment variable in bash\_profile

```
# .bash_profile
# Get the aliases and functions
if [ -f './.bashrc' ]; then
    . ./.bashrc
fi

# User specific environment and startup programs
PATH=$PATH:$HOME/.local/bin:$HOME/bin

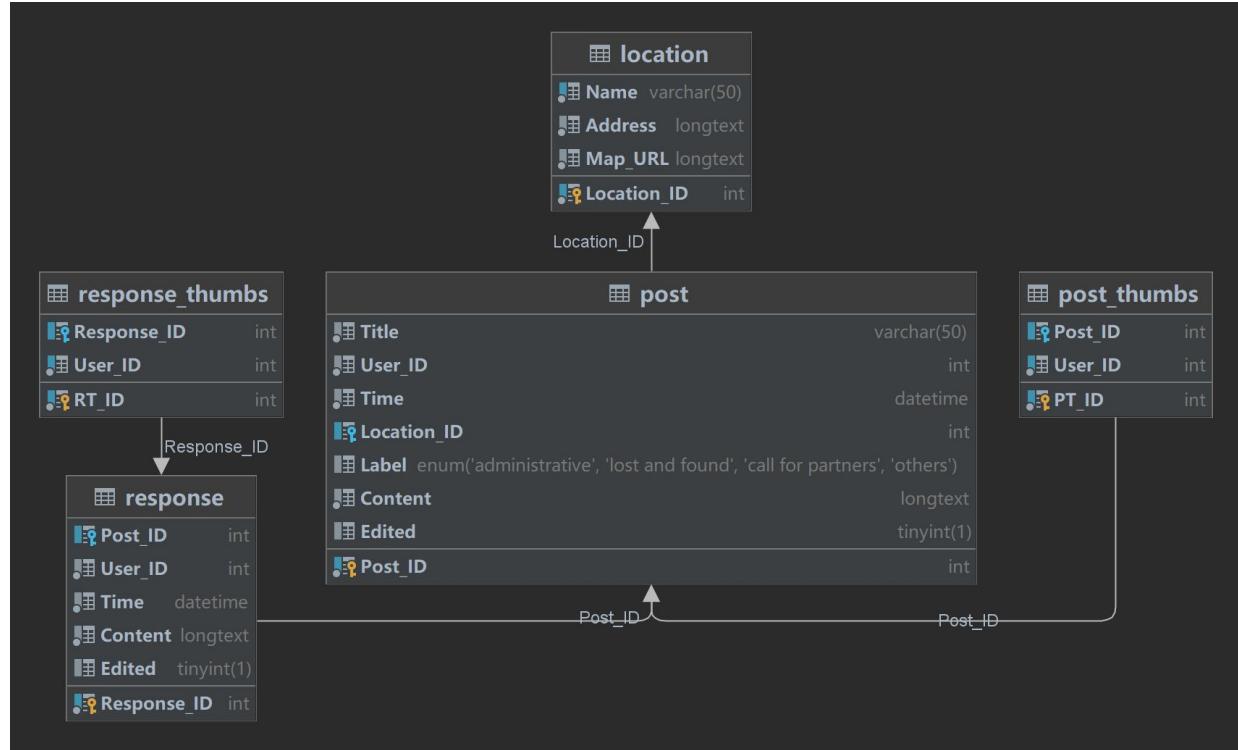
export PATH
export AWS_ACCESS_KEY_ID=...
export DBNAME=jmapp0200
export DBHOST=ea6156.cfjfinuvyknb.us-east-1.rds.amazonaws.com
export MS1_URL=http://44.204.1.61.68:5010/
export MS2_URL=http://54.167.191.86:5011/
export MS3_URL=http://e6156ms3-env.eba-jjye9bmm.us-east-1.elasticbeanstalk.com/
export WEB_APP_URL=http://localhost:4200/
export GOOGLE_CLIENT_ID=432898219349-ufa3r8g4885c1iu70vfeh0lb.apps.googleusercontent.com
export GOOGLE_CLIENT_SECRET=GOOG1-CQ83q3byFCF5lyEXDNz-5SLRpm...
export AWS_REGION=us-east-1
export key_id=AKIA73A3PFYWZ67K6VD
export region=us-east-1
export Topic_ARN=arn:aws:sns:us-east-1:893019433329:
```

MS2 EC2 instance running on port 5011

```
[ec2-user@ip-172-31-88-230 ~]$ python3 CUMC-Badminton-Services-MS2/src/application.py
 * Serving Flask app 'application'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5011
 * Running on http://172.31.88.230:5011
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 248-208-809
```

# MS3: Forum Service

# DB design



# Community Space: Posting, Liking and Commenting

Users are allowed to make posts and share announcements for the community to interact with, such as reacting or responding to them, and view them by popularity, recency, category, or any combination of them.

Everyone has access to editing their own posts and responses.

The screenshot shows a forum interface with a green header bar containing the title "Forum" and the subtitle "A place to share your love with Badminton". Below the header, there are navigation tabs: "All Posts", "Lost and Found", "Others", "Call for Partners", and "Administrative". A large green "New Post" button is located on the right side of the header. The main content area displays three posts:

- test cloudfront** (8 posts, December 16, 2022) - Located at Morningside Dodge Fitness Center. It has 0 likes and a "Call for Partners" button.
- Badminton at Blue Gym** (4 posts, October 16, 2022) - Located at Morningside Dodge Fitness Center. It has 2 likes and a "Call for Partners" button.
- Friday night badminton?** (2 posts, October 14, 2022) - Located at CUIMC 50 Haven Athletic Center. It has 3 likes and an "Others" button.

At the bottom of the post list, there is a navigation bar with page numbers 1, 2, 3, 4 (highlighted in green), and 5.

Beanstalk: E6156ms3-env.eba-jjys9bmm.us-east-1.elasticbeanstalk.com

[Elastic Beanstalk](#) > [Environments](#) > E6156ms3-env

**E6156ms3-env**  
E6156ms3-env.eba-ijjys9bmm.us-east-1.elasticbeanstalk.com [e-bqsfzjybjqe]  
Application name: E6156-MS3

**Health**



Ok

[Causes](#)

**Running version**  
MS3.v9  
[Upload and deploy](#)

**Platform**  
  
 Python 3.8 running on 64bit  
 Amazon Linux 2/3.4.2  
[Change](#)

**Recent events**

Time	Type	Details
2022-12-19 11:44:38 UTC+0800	INFO	Environment health has transitioned from Info to Ok. Configuration update completed 70 seconds ago and took 76 seconds.
2022-12-19 11:43:25 UTC+0800	INFO	Environment update completed successfully.
2022-12-19 11:43:25 UTC+0800	INFO	Successfully deployed new configuration to environment.
2022-12-19 11:42:50 UTC+0800	INFO	Instance deployment completed successfully.
2022-12-19 11:42:47 UTC+0800	INFO	Instance deployment successfully generated a 'Profile'.

MS3 Beanstalk / [get all posts](#)

POST [{{bs\\_url}}/api/forum/user\\_id/:user\\_id](#)

Authorization Headers (8) Body **JSON** Pre-request Script Tests Settings

form-data x-www-form-urlencoded raw binary GraphQL

```

"label": 2,
"sort": 1,
"limit": 2,
"page": 1,
"mypost": 0

```

Cookies Headers (6) Test Results

Status: 200 OK Time: 1649 ms Size: 938 B [Save Response](#)

Raw Preview Visualize JSON

```

"success": true,
"data": [
  {
    "Post_ID": 5,
    "Title": "postman edits post",
    "User_ID": 8,
    "Time": "2022-12-16 21:48:05",
    "Location_ID": 2,
    "Location": "Morningside Dodge Fitness Center",
    "Map_URL": "https://goo.gl/maps/SAQEFMUKiDQksuIXA",
    "Label": "Lost and Found",
    "Content": "some edited content",
    "Edited": 1,
    "Thumbs": 0,
    "is_Thumbed": 0
  },
  {
    "Post_ID": 10,
    "Title": "add post local",
    "User_ID": 11,
    "Time": "2022-12-16 03:51:26",
    "Location_ID": 4,
    "Location": "new loc",
    "Map_URL": "https://www.google.c

```

Pretty Raw Preview Visualize JSON

```

1
2   "success": false,
3   "message": "Post exists but the current user cannot edit it."
4

```

# smarty API

An announcement could be tagged with a location, and its address is verified before the it is officially posted. We provide both a address suggestion service in the front end, and a validation service in the backend for viewers to pin the address on Google Maps.

MS3 Beanstalk / location lookup

POST `{{bs_url}}/api/forum/location_lookup`

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {  
2   "line1": "308 E 209",  
3   "line2": "",  
4   "city": "Bronx",  
5   "state": "NY",  
6   "zipcode": "10065"  
7 }
```

Body Cookies Headers (6) Test Results

Pretty Raw Preview Visualize JSON

```
1 {  
2   "success": true,  
3   "address": "308 E 209th St, Bronx NY 10467-3307",  
4   "message": "Valid address found via Smarty"  
5 }
```

# Composite Service to get all user information

# Composite service to get all profile

```
from flask import Flask, Response, request
from flask_cors import CORS
import json
from typing import *
from requests_futures.sessions import FuturesSession
import requests
from utils import DTEncoder
import os

app = Flask(__name__)
CORS(app)

session = FuturesSession()
def get_complete_user_profile(userid):
    rsp_data = {'user_info':None, 'partner_info':None, 'post_info':None}
    future1 = session.get(f'{os.environ.get("MS2_URL")}/api/userprofile3/{userid}')
    future2 = session.get(f'{os.environ.get("MS1_URL")}/api/user/{userid}/partner')
    future3 = session.post(f'{os.environ.get("MS3_URL")}/api/forum/user_id/{userid}', json = {
        "label": 5,
        "sort": 1,
        "limit": 1000,
        "page": 1,
        "mypost": 1
    })
    response1 = future1.result().json()
    response2 = future2.result().json()

    # if user not found
    if not response1['success']:
        rsp = {'success':False, 'data':rsp_data}
        return rsp

    if future3.result():
        response3 = future3.result().json()
        rsp_data['user_info'] = response1['data'][0]
        rsp_data['post_info'] = response3

    # if the user has a partner, get partner's infomation
    if response2['success']:
        partner_id = list(response2['data'][0].values())[0]
        future21 = session.get(f'http://localhost:5011/api/userprofile3/{partner_id}')
        response21 = future21.result().json()
        rsp_data['partner_info'] = response21['data'][0]

    rsp = {'success':True, 'data':rsp_data}
    return rsp
```

Using FutureSession method to make three MS Synchronous. Getting user info, partner info and post info at the same time.

# API demo

If the user has post and partner

6156-CS1 / get complete user profile

GET https://r450v9soc3.execute-api.us-east-1.amazonaws.com/stage1/api/composite/user\_profile\_all/9

Send Save Cookies

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1
2 "success": true,
3 "data": {
4   "user_info": {
5     "userid": 9,
6     "profile_pic": "https://lh3.googleusercontent.com/a/AEfTp5qcDX04ur_BXJ70xc9RddfAboXlm0rdq9Chw=96-c",
7     "role": "Admin",
8     "email": "zheyian.liu@columbia.edu",
9     "username": "Zheyian Liu",
10    "sex": null,
11    "preference": null,
12    "credits": 100,
13    "birthday": null
14  },
15  "partner_info": {
16    "userid": 11,
17    "profile_pic": "https://lh3.googleusercontent.com/a/AEfTp4iF3WGV5_qdJass1nPqGm5WkzU1is5078-H=s96-c",
18    "role": "User",
19    "email": "zheyian.liu1999@gmail.com",
20    "username": "Zheyian Liu",
21    "sex": null,
22    "preference": null,
23    "credits": 100,
24    "birthday": null
25  },
26  "post_info": {
27    "success": false,
28    "data": [],
29    "message": "No post found in the described category"
30  }
31}
32
```

If the user do not have post and partner

6156-CS1 / get complete user profile

GET https://r450v9soc3.execute-api.us-east-1.amazonaws.com/stage1/api/composite/user\_profile\_all/6

Send Save Cookies

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON

```
1
2 "success": true,
3 "data": {
4   "user_info": {
5     "userid": 6,
6     "profile_pic": null,
7     "role": "User",
8     "email": "test@test.com",
9     "username": "mushroom",
10    "sex": "Male",
11    "preference": null,
12    "credits": 100,
13    "birthday": "2021-01-01"
14  },
15  "partner_info": null,
16  "post_info": {
17    "success": false,
18    "data": [],
19    "message": "No post found in the described category"
20  }
21}
22
```

# UI demo



Zheyen Liu

Information	
Email	Birthday
zheyen.liu1999@gmail.com	
Sex	Preference
Credits	Total posts
100	1
Partner Information	
Username	Email
Zheyen Liu	zheyen.liu@columbia.edu
Preference	Credits
100	

- MS1: Partner Infomation
- MS3: Total posts
- MS2: The rest

# EC2 Deployment

```
[ec2-user@ip-172-31-88-230 ~]$ python3 CUMC-Badminton-Services-CS1/application.py
 * Serving Flask app 'application'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5013
 * Running on http://172.31.88.230:5013
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 248-208-809
```

# Useful links

## Web App cloudfront url

<https://d60lv0um3jt5y.cloudfront.net/>

## Github Repo

- MS1:<https://github.com/ZhuolunHuang/CUMC-Badminton-Services-MS1.git>
- MS2:<https://github.com/zheyanliu99/CUMC-Badminton-Services-MS2.git>
- MS3:<https://github.com/YiruGONG/E6156-Microservice3-Yiru-Brian.git>
- CS1:<https://github.com/zheyanliu99/CUMC-Badminton-Services-CS1.git>
- Web App:<https://github.com/zheyanliu99/CUMC-Badminton-Services-Web-app.git>

## API public URL (EC2/Beanstalk)

- MS1:<http://44.204.161.68:5010/>
- MS2:<http://54.167.191.86:5011/>
- MS3:<http://e6156ms3-env.eba-jiys9bmm.us-east-1.elasticbeanstalk.com/>
- CS1:<http://54.167.191.86:5013/>
- API GATEWAY:<https://r450v9soc3.execute-api.us-east-1.amazonaws.com/stage1/>

# Trello

<https://trello.com/invite/b/3edEJxO1/f586d1265c9e672a531672e12347c147/cumc-badminton-service>

The Trello board displays the following lists and their contents:

- To Do**:
  - + Add a card
- Doing**:
  - + Add a card
- Done**:
  - MS2 Sprint1
    - 1/11
  - Web Browser
    - 0/4
  - MS1
    - 1/16
  - Google OAuth Sprint 2
    - 1/1
  - Composite user profile Sprint 2
    - 1/1
  - Admin features and middleware
    - 1/1
  - Profile page update
    - 1/1
  - APIGATEWAY and cloudfront
    - 1/1
  - Notification add partner Sprint 2
    - 1/1
  - Sprint 1 Amend
    - 1/1
  - MS3 Smarty
    - 5/5
  - MS3 Sprint 1
    - 23/26
- + Add a card

# Thank you