# API Testing Guide

Base URL: `http://localhost:10888/api/v1`

## Authentication

### Admin Login

- \*\*Endpoint\*\*: `POST /agency/auth`

- \*\*Auth\*\*: Basic Authentication (Admin credentials)

- \*\*Purpose\*\*: Used for admin operations and agency creation

### Member Login

- \*\*Endpoint\*\*: `POST /members/auth`

- \*\*Auth\*\*: Basic Authentication

- \*\*Purpose\*\*: Member authentication

## Agency Management

### 1. Create Agency

- \*\*Endpoint\*\*: `POST /agency/auth`

- \*\*Method\*\*: POST

- \*\*Auth\*\*: Basic Auth (Admin credentials)

- \*\*Body\*\*:

```json

{

"username": "mary",

"password": "12345678",

"email": "mary@wanderlust.com",

"phone": "86123402",

"name": {

"firstname": "mary",

"lastname": "mary",

"nickname": "mary"

}

}

```

### 2. Upload Agency Photo

- \*\*Endpoint\*\*: `POST /agency/upload-photo`

- \*\*Method\*\*: POST

- \*\*Auth\*\*: Basic Auth

```plaintext

Username: peter

Password: 12345678

```

- \*\*Body\*\*: form-data

- Key: `profilePhoto`

- Type: File

- Value: [Select image file]

### 3. Get Agency Photo

- \*\*Endpoint\*\*: `GET /agency/{username}`

- \*\*Method\*\*: GET

- \*\*Example\*\*: `GET /agency/peter`

- \*\*Response\*\*: Image file

## Hotel Management

### 1. Add Hotel

- \*\*Endpoint\*\*: `POST /hotel`

- \*\*Method\*\*: POST

- \*\*Body\*\*:

```json

{

"star": 8,

"name": "8 Seasons Hotel Hong Kong",

"accommodationType": "Hotel",

"address": "8 Finance Street, Central",

"city": "Hong Kong",

"coordinates": {

"latitude": 22.2863701,

"longitude": 114.0849434

},

"country": "Hong Kong, China",

"description": "Luxe harbour-view hotel with a pool",

"email": "info@fourseasons.com",

"facilities": [

"Pool",

"Spa",

"Free WiFi",

"Air-conditioned"

],

"lastUpdate": "2025-03-15",

"phones": "(852)-3196-8888",

"ranking": 5,

"web": "www.fourseasons.com",

"token": "1743436646980"

}

```

### 2. Update Hotel

- \*\*Endpoint\*\*: `PUT /hotel/{hotelId}`

- \*\*Method\*\*: PUT

- \*\*Example\*\*: `PUT /hotel/67ec3525250d2b71082e472b`

- \*\*Body\*\*:

```json

{

"star": 8,

"name": "8 Seasons Hotel Hong Kong",

"accommodationType": "Hotel",

"address": "8 Finance Street, Central",

"city": "Hong Kong",

"coordinates": {

"latitude": 22.2863701,

"longitude": 114.0849434

},

"country": "Hong Kong, China",

"description": "Luxe harbour-view hotel with a pool",

"email": "info@fourseasons.com",

"facilities": [

"Pool",

"Spa",

"Free WiFi",

"Air-conditioned"

],

"lastUpdate": "2025-03-15",

"phones": "(852)-3196-8888",

"ranking": 5,

"web": "www.fourseasons.com",

"token": "1743436646980"

}

```

### 3. Delete Hotel

- \*\*Endpoint\*\*: `DELETE /hotel/{hotelId}`

- \*\*Method\*\*: DELETE

- \*\*Example\*\*: `DELETE /hotel/67ec3525250d2b71082e472b`

- \*\*Body\*\*:

```json

{

"token": "1743436646980"

}

```

## Member Management

### 1. Register Member

- \*\*Endpoint\*\*: `POST /member`

- \*\*Method\*\*: POST

- \*\*Auth\*\*: None required

- \*\*Body\*\*:

```json

{

"username": "mary",

"password": "12345678",

"email": "mary@wanderlust.com",

"phone": "86123402",

"name": {

"firstname": "mary",

"lastname": "mary",

"nickname": "mary"

}

}

```

### 2. Get Favorites List

- \*\*Endpoint\*\*: `GET /favourlist`

- \*\*Method\*\*: GET

- \*\*Body\*\*:

```json

{

"token": "1743528462289",

"username": "baby",

"role": 2

}

```

## Messaging System

### 1. Send Message

- \*\*Endpoint\*\*: `POST /message`

- \*\*Method\*\*: POST

- \*\*Body\*\*:

```json

{

"token": "1743528462289",

"receiver": "agencyq",

"content": "Hello, I am interested in your hotel listings",

"type": "text"

}

```

### 2. Get Messages

- \*\*Endpoint\*\*: `GET /message`

- \*\*Method\*\*: GET

- \*\*Body\*\*:

```json

{

"token": "1743427268076"

}

```

### 3. Delete Message

- \*\*Endpoint\*\*: `DELETE /message`

- \*\*Method\*\*: DELETE

- \*\*Body\*\*:

```json

{

"token": "1743528462289",

"messageId": "67ec27def151a312661411a7"

}

```

## Testing Flow

### 1. Initial Setup

1. Start with admin login

2. Create an agency account

3. Upload agency photo

4. Verify agency photo retrieval

### 2. Hotel Management Flow

1. Add a new hotel

2. Update hotel details

3. Delete hotel

### 3. Member Operations

1. Register new member

2. Test member authentication

3. Add hotels to favorites

4. View favorites list

### 4. Communication Testing

1. Send message from member to agency

2. Retrieve messages

3. Delete messages

## Postman Collection Setup

### Environment Variables

Set up the following variables:

- `base\_url`: http://localhost:10888/api/v1

- `token`: [Your authentication token]

- `admin\_username`: [Admin username]

- `admin\_password`: [Admin password]

### Authentication Headers

For endpoints requiring Basic Auth:

1. Go to "Authorization" tab

2. Select "Basic Auth"

3. Enter credentials as required

### File Upload Testing

For photo upload:

1. Select "Body" tab

2. Choose "form-data"

3. Add key "profilePhoto"

4. Set type to File

5. Select image file to upload

## Error Testing

Test each endpoint with:

1. Invalid tokens

2. Missing required fields

3. Invalid data formats

4. Unauthorized access

5. Non-existent IDs

## Success Criteria

- 200: Successful operation

- 201: Successfully created

- 400: Bad request

- 401: Unauthorized

- 403: Forbidden

- 404: Not found

* 500: Server error

I am a member with login details created in "members" database on MongoDB. When I try to login via http://localhost:8888/api/v1/members/auth, I find that I cannot login. Instead I have to create login details created in "users" database first to login. Fix the code so that I could login only with login details created in "members" database

Update member:

member should have right to update or delete the member own account.

admin should have right to update or delete all member account.

for members.ts, the members data should be stored in users database instead of members on Mongodb, and the new members registered should be assigned as role 2. (0 for admin and 1 for agency)

Fix all the related files too, e.g. message.ts

since now agency.ts and members.ts point to the same "users" database on MongoDB, basicAuth2.ts should not be required and members.ts should use basicAuth instead of basicAuth2. Fix all files.

For message.ts, message should only be sent and received between member (role 2) with agency (role 1), but not member with member or agency or agency. Update the code.