# Signup

POST /api/profile/signup

Request: {

'email': str, 'password': str,

'phone': str [opt],

'use\_phone': bool [opt],

'use\_msg': bool [opt],

}

Response: 201: {'profile\_id': profile.id}; 4xx: {'error': {'message': str}}

# Login

POST /api/auth

Request: {'email': str, 'password': str}

Response: 200: {'access\_token': str, 'profile': int}; 4xx: {'error': {'message': str}}

Comments: ‘access\_token’ is JWT token. Use it in request header:

"Authorization: JWT <access\_token>"

# Delete profile

DELETE /api/profile/<int:profile\_id>

jwt required

Request: {}

Response: 204; 4xx: {'error': {'message': str}}

# Update profile

PUT /api/profile/<int:profile\_id>

jwt required

Request: {

'phone': str,

'use\_phone': bool,

'use\_msg': bool,

}, any field is optional

Response: 204; 4xx: {'error': {'message': str}}

# Fetch profile data

GET /api/profile/<int:profile\_id>

Response:

200: {

'profile\_id': int,

'phone': str,

'use\_phone': bool,

'use\_msg': bool

};

4xx: {'error': {'message': str}}

# Create Friend record

POST /api/profile/<int:profile\_id>/friends/new

jwt required

Request: {

'name': str, 'breed': str, 'sex': str, 'age': str, 'location': str, 'status': str

}

Response: 201: {'friend\_id': friends.id}; 4xx: {'error': {'message': str}}

# Fetch Friends list

GET /api/profile/<int:profile\_id>/friends

jwt required

Response: 200: [{'friend\_id':int}, …], 4xx: {'error': {'message': str}}

# Fetch Friend data

GET /api/friend/<int:friend\_id>

Response: 200: {

'friend\_id': int,

'name': str,

'breed': str,

'sex': str,

'age': str,

'location': str,

'status': str

};

4xx: {'error': {'message': str}}

# Delete friend record

DELETE /api/friend/<int:friend\_id>

jwt required

Request: {}

Response: 204; 4xx: {'error': {'message': str}}

# Upload photo

PUT /api/friend/<int:friend\_id>/photo

jwt required

Request: {'image': {'data': str, 'type': str}}

'data': base64 encoded image body, 'type': content-type value

Response: 204; 4xx: {'error': {'message': str}}

# Download photo

GET /api/friend/<int:friend\_id>/photo

Response: 200: {'image': {'data': str, 'type': str}}; 4xx: {'error': {'message': str}}

# Match photo

POST /api/query\_match

Request: {'image': {'data': str, 'type': str}}

Response:

200: {'status': 'not found'};

200: {'status': 'found', 'profile': int, 'percent': int};

4xx: {'error': {'message': str}}

# Match photo with the best matches available

# Use this endpoint to get up to max\_best\_friends best-matching friends for a given query image.

# **Outputs**:

# friend\_ids is a list of database ids for the best matching friends, sorted from best match to worst match.

# Percents is a list of floats specifying the confidence in each match. Higher is better. In decreasing order.

POST /api/query\_matches/<int:max\_best\_friends>

Request: {'image': {'data': str, 'type': str}}

Response:

200: {'status': 'not found'};

200: {'status': 'found', 'friend\_ids': listOfInts, 'percents': listOfFloats};

4xx: {'error': {'message': str}}