Hands UP

Service Contracts

Updated: 18/08/2025

Contents

1 Service Contracts

1.1 User Sign Up

1.1.1 signup

Operation: `signup(name, surname, username, email, password)`

Input: User details object consisting of:

- name (String, required)
- surname (String, required)
- username (String, required)
- email (String, required)
- createdAt (Timestamp, optional)

Output: Successful creation response

Precondition: Email must not already exist

Postcondition: New user must be added to the system

Errors:

- 409 If email already exists
- 500 Internal server error

1.2 User Login

1.2.1 login

Operation: `login(email, password)`

Input: User details object consisting of:

- email (String, required)
- password (String, required)

Output: Successful login response

Preconditions: Email must already exist and password must match stored hash

Postcondition: User marked as logged in

Errors:

- 401 Invalid email or password
- 404 If email not registered
- 500 Internal server error

1.3 User Details

1.3.1 uniqueUsername

Operation: `uniqueUsername(username)`

Input: User details object consisting of:

- username (String, required)

Output: Object consisting of:

- exists (Boolean)

Precondition: A non-empty username must be provided

Postcondition: Returns `true` if the username exists, otherwise `false`

Errors:

- 500 Internal server error

1.3.2 uniqueEmail

Operation: `uniqueEmail(email)`

Input: User details object consisting of:

- email (String, required)

Output: Object consisting of:

- exists (Boolean)

Precondition: A valid email must be provided

Postcondition: Returns `true` if the email exists, otherwise `false`

Errors:

- 500 Internal server error

1.4 User Profile

1.4.1 getUserData

Operation: `getUserData(userID)`

Input:

- userID (Number, required)

Output: User object consisting of:

- userID (Number)
- username (String)
- name (String)
- surname (String)
- email (String)
- avatarURL (Blob, optional)
- createdAt (Timestamp, optional)

Precondition: `userID` must correspond to an existing user in the database

Postcondition: Returns user details if found

Errors:

- 404 If `userID` not found
- 500 Internal server error

1.4.2 updateUserDetails

Operation: `updateUserDetails(userID, name, surname, username, email)`

Input: User details object consisting of:

- userID (Number, required)
- name (String, required)
- surname (String, required)
- username (String, required)
- email (String, required)
- avatarURL (Blob, optional)

Outputs: Successful update response and updated user object consisting of:

- userID (Number)

- username (String)
- name (String)
- surname (String)
- email (String)
- avatarURL (Blob, optional)
- createdAt (Timestamp, optional)

Preconditions: `userID` must exist and input fields must be valid

Postcondition: User information is updated in the database

Errors:

- 404 If `userID` not found
- 500 Internal server error

1.4.3 updateUserPassword

Operation: `updateUserPassword(userID, name, surname, username, email, password)`

Input: User details object consisting of:

- userID (Number, required)
- name (String, required)
- surname (String, required)
- username (String, required)
- email (String, required)
- password (String, required)
- avatarURL (Blob, optional)

Outputs: Successful update response and updated user object consisting of:

- userID (Number)
- username (String)
- name (String)
- surname (String)
- email (String)
- avatarURL (Blob, optional)
- createdAt (Timestamp, optional)

Preconditions: `userID` must exist and input fields must be valid

Postcondition: User information and hashed password are updated in the database

Errors:

- 404 If `userID` not found
- 500 Internal server error

1.5 User Progress

1.5.1 learningProgress (get)

Operation: `learningProgress(username)`

Input:

- username (String, required)

Output: Learning progress object consisting of:

- lessonsCompleted (Number)
- streak (Number)
- signsLearned (Number)
- currentLevel (String)

Precondition: Username must exist in the system

Postcondition: Returns the correct user's learning progress

Errors:

- 404 If username not found
- 400 If username not provided
- 500 Internal server error

1.5.2 learningProgress (update)

Operation: `learningProgress(username, progressObject)`

Input: Username and learning progress object consisting of:

- username (String, required)
- lessonsCompleted (Number, required)
- streak (Number, required)
- signsLearned (Number, required)

- currentLevel (String, required)

Output: Successful update response

Precondition: Username must exist in the system

Postcondition: The user's learning stats are updated

Errors:

- 404 If username not found
- 400 If username or learning stats object not provided
- 500 Internal server error

1.6 Translate Sign

1.6.1 translate

Operation: `translate(inputVideo)`

Input:

- inputVideo (Blob, required)

Output: Translation object consisting of:

- letter (String)
- letterConfidence (String)
- number (String)
- numberConfidence (String)

Precondition: AI inference should be possible

Postcondition: Returns signs translated to English

Errors:

- 500 Internal server error