API Documentation

POST /api/auth/register

Description

Registers a new user.

Request Body (JSON)

```
{
  "email": "user@domain.com",
  "name": "User Name",
  "password": "password",
  "university": "University"
}
```

Response

- `200 OK`: User successfully registered.
- `401 Unauthorized`: A user with the same email already exists.

POST /api/auth/login

Description

Logs in a user with email and password.

Request Body (JSON)

```
{
  "email": "user@domain.com",
  "password": "password"
}
```

Response

- `200 OK`: Returns an access token.
- `401 Unauthorized`: Incorrect credentials.
- `400 Bad Request`: Malformed data.

POST /api/auth/loginWithToken

Description

Logs in using a previously issued authentication token.

Request Body (JSON)

```
{
  "email": "user@domain.com",
  "token": "access_token"
}
```

Response

- **`200 OK`**: Valid token.
- **`400 Bad Request`**: Invalid or expired token.

POST /api/auth/logout

Description

Logs out the user.

Request Body (JSON)

```
{
  "email": "user@domain.com",
  "token": "access_token"
}
```

Response

- **`200 OK`**: Successfully logged out.
- **`400 Bad Request`**: Token not found or invalid.

GET /api/users/me

Description

Returns details of the authenticated user.

Request Parameters (Query)

- `token`: Authentication token.

Response

- **`200 OK`**: User information.
- **`400 Bad Request`**: Invalid or missing token.

POST /api/auth/forgotPassword

Description

Sends a password recovery email.

Request Body (JSON)

```
{
  "email": "user@domain.com"
}
```

Response

- **`200 OK`**: Recovery email sent.
- **`400 Bad Request`**: Error recovering password.

POST /api/exchanges

Description

Creates a new language exchange.

Request Body (JSON)

```
{
  "token": "access_token",
  "nativeLanguage": "Spanish",
  "targetLanguage": "English",
  "educationalLevel": "High School",
  "academicLevel": "Basic",
  "beginDate": "01-01-2025",
  "endDate": "01-06-2025",
  "quantityStudents": 10,
  "university": "Example University"
}
```

Response

- **`201 Created`**: Exchange successfully created.
- **`400 Bad Request`**: Invalid token or processing error.

GET /api/exchanges

Description

Retrieves available exchanges with search parameters.

Request Parameters (Query)

- `nativeLanguage`, `targetLanguage`, `educationalLevel`, `academicLevel`, `beginDate`, `endDate`, `quantityStudentsMin`, `quantityStudentsMax`, `university`, `userId`, `status`.

Response

- **`200 OK`**: List of exchanges.

GET /api/exchanges/

Description

Retrieves a specific exchange by its ID.

Request Parameters (Query)

- `exchangeId`: Exchange ID.

Response

- **`200 OK`**: Exchange details.
- **`404 Not Found`**: Exchange not found.

POST /api/messages

Description

Sends a message between two users.

Request Body (JSON)

```
{
  "idUserSender": "sender_id",
  "idUserRecipient": "recipient_id",
  "message": "Hello!",
  "typeMessage": "text",
  "token": "access_token"
}
```

Response

- **`201 Created`**: Message sent.
- **`400 Bad Request`**: Invalid request body.
- **`401 Unauthorized`**: Invalid or expired token.

PUT /api/notifications/read

Description

Marks all messages in a conversation as read.

Request Body (JSON)

```
{
  "idUserSender": "sender_id",
  "idUserRecipient": "recipient_id",
  "token": "access_token"
}
```

Response

- **`200 OK`**: Messages marked as read.
- **`400 Bad Request`**: Invalid JSON format.
- **`401 Unauthorized`**: Invalid or expired token.
- **`204 No Content`**: No new messages available.

GET /api/messages/conversation

Description

This endpoint retrieves all messages exchanged between two users. The parameters `idUserSender`, `idUserRecipient`, and `token` must be sent as URL parameters.

Successful Response

- **Status Code:** `200 OK`
- **Response Body:** Returns a list of messages exchanged between the specified users.

Error Responses

- **`401 Unauthorized`**: If the provided token is invalid or has expired.
- **`204 No Content`**: If no messages are found between the users.

PUT /api/exchanges/

Description

This endpoint updates the details of a specific exchange. The exchange ID must be sent as a URL parameter, and the request body must contain a JSON object with the fields to be updated. A valid authentication token is required in the request.

Successful Response

- **Status Code:** `200 OK`
- **Response Body:** Returns the updated exchange.

Error Responses

- **`400 Bad Request`**: If there is an error in the request payload.
- **`401 Unauthorized`**: If the provided token is invalid or has expired.
- **`404 Not Found`**: If the specified exchange is not found.

GET /api/messages/conversation

Description

This endpoint retrieves all messages exchanged between two users. The parameters `idUserSender`, `idUserRecipient`, and `token` must be sent as URL parameters.

Successful Response

- **Status Code:** `200 OK`
- **Response Body:** Returns a list of messages exchanged between the specified users.

Error Responses

- **`401 Unauthorized`**: If the provided token is invalid or has expired.
- **`204 No Content`**: If no messages are found between the users.

PUT /api/exchanges/

Description

This endpoint updates the details of a specific exchange. The exchange ID must be sent as a URL parameter, and the request body must contain a JSON object with the fields to be updated. A valid authentication token is required in the request.

Successful Response

- **Status Code:** `200 OK`
- **Response Body:** Returns the updated exchange.

Error Responses

- **`400 Bad Request`**: If there is an error in the request payload.
- **`401 Unauthorized`**: If the provided token is invalid or has expired.
- **`404 Not Found`**: If the specified exchange is not found.

DELETE /api/exchanges/

Description

This endpoint deletes an exchange given an ID and an authentication token.

Parameters

- **`id` (String)**: The ID of the exchange to be deleted.
- **`token` (String)**: Authentication token.

Successful Response

- **Status Code:** `200 OK`
- **Response Body:** Confirmation message of the exchange deletion.

Error Responses

- `400 Bad Request`: If the token is invalid or expired.
- `404 Not Found`: If the exchange is not found.
- `500 Internal Server Error`: If an error occurs during the deletion process.

POST /api/groups

Description

This endpoint creates a new group. Requires an authentication token and group data.

Request Body (JSON)

```
{
  "token": "access_token",
  "name": "Group Name",
  "languageLevel": 2,
  "quantity": 10,
  "language": "English"
}
```

Successful Response

- **Status Code:** `201 Created`
- **Response Body:** Message indicating successful group creation.

Error Responses

- `400 Bad Request`: If the token is invalid or expired.
- `500 Internal Server Error`: If an error occurs while creating the group.

GET /api/groups

Description

This endpoint retrieves the groups associated with a teacher's ID using an authentication token.

Parameters

- `token` **(String)**: Authentication token.

Successful Response

- **Status Code:** `200 OK`
- **Response Body:** The group associated with the teacher's ID.

Error Responses

- `400 Bad Request`: If the token is invalid or expired.

- **`404 Not Found`**: If no group is found.

GET /api/groups/

Description

This endpoint retrieves a group by its ID using an authentication token.

Parameters

- **`groupId` (String)**: The ID of the group to retrieve.
- **`token` (String)**: Authentication token.

Successful Response

- **Status Code:** `200 OK`
- **Response Body:** The requested group.

Error Responses

- **`400 Bad Request`**: If the token is invalid or expired.
- **`404 Not Found`**: If the group is not found.

PUT /api/groups/

Description

This endpoint updates an existing group with the new values provided in the request.

Parameters

- **`id` (String)**: The ID of the group to update.
- **`groupJson` (String)**: JSON body with new group details (`name`, `languageLevel`, `quantity`, `language`, `userId`).

Successful Response

- **Status Code:** `200 OK`
- **Response Body:** The updated group.

Error Responses

- **`400 Bad Request`**: If the provided data is incorrect.
- **`404 Not Found`**: If the group is not found.

DELETE /api/groups/

Description

This endpoint deletes a group given an ID and an authentication token.

Parameters

- **`id` (String)**: The ID of the group to be deleted.
- **`token` (String)**: Authentication token.

Successful Response

- **Status Code:** `200 OK`
- **Response Body:** Confirmation message of the group deletion.

Error Responses

- **`400 Bad Request`**: If the token is invalid or expired.
- **`404 Not Found`**: If the group is not found.
- **`500 Internal Server Error`**: If an error occurs during the deletion process.

POST /api/upload

Description

This endpoint uploads a file to Cloudinary.

Parameters

- **`file` (MultipartFile)**: The file to be uploaded.

Successful Response

- **Status Code:** `200 OK`
- **Response Body:** URL of the uploaded file.

Error Responses

- **`500 Internal Server Error`**: If an error occurs during file upload.

POST /api/preferences

Description

This endpoint creates user preferences like dark mode, notifications, and language using an authentication token.

Request Body (JSON)

```
{
  "token": "access_token",
  "darkMode": true,
  "notifications": true,
  "language": "English"
}
```

Successful Response

- **Status Code:** `201 Created`

Error Responses

- **`400 Bad Request`**: If the token is invalid or expired.
- **`500 Internal Server Error`**: If an error occurs while creating preferences.

GET /api/users/me/preferences

Description

This endpoint retrieves the authenticated user's preferences using a token.

Parameters

- **`token` (String)**: Authentication token.

Successful Response

- **Status Code:** `200 OK`

Error Responses

- **`400 Bad Request`**: If the token is invalid or expired.
- **`404 Not Found`**: If no preferences are found for the user.

PUT /api/users/me/preferences

Description

This endpoint updates the user's preferences.

Request Body (JSON):

```
{
  "token": "string",
  "darkMode": "boolean",
  "notifications": "boolean",
  "language": "string"
}
```

Successful Response

- **Status Code:** `200 OK`
- **Response Body:** Returns a message indicating that the preferences were successfully updated.

Error Responses

- **`400 Bad Request`**: If the token is invalid or has expired.
- **`404 Not Found`**: If the user preferences are not found.

PUT /api/auth/update

Description

This endpoint updates user details such as name, university, and profile pictures.

Request Body (JSON)

```
{
  "token": "access_token",
  "name": "User Name",
  "university": "University Name",
  "urlProfilePicture": "profile_url",
  "urlHeaderPicture": "header_url"
}
```

Successful Response

- **Status Code:** `200 OK`

Error Responses

- **`400 Bad Request`**: If data is incorrect.
- **`404 Not Found`**: If the user is not found.

GET /api/users/image

Description

This endpoint retrieves a user's profile image by email.

Parameters

- **`email` (String)**: The user's email.

Successful Response

- **Status Code:** `200 OK`

Error Responses

- **`404 Not Found`**: If the user is not found.