

API Documentation

Overview

This document provides detailed information on all API endpoints in the application, including Campaign, Statement, User, Authentication, Payment, and Upload APIs. Each endpoint description includes the request method, URL, request parameters, expected responses, and sample response data.

1 Campaign API Group

1.1 Get All Campaigns

- Endpoint: `../api/campaigns`
- Method: `GET`
- Description: Retrieve a list of all campaign data.
- Response:
 - Status: 200 OK
 - Body: Array of Campaign objects

Sample Response:

```
[
  {
    "_id": "ObjectId",
    "id": "String",
    "name": "string",
    "status": "string",
    "description": "string",
    "pictureFiles": ["UploadThingPictureFile"],
    "companyName": "string",
    "website": "string",
    "founderName": "string",
    "email": "string",
    "linkedInProfile": "string",
    "companyStage": "string",
    "industry": "string",
    "sector": "string",
    "amountRaised": 10000,
    "targetAmount": 50000,
    "teamSize": 10,
    "headquartersLocation": "string",
    "productAvailable": true,
    "location": "string",
    "incorporationDate": "string",
    "investors": ["string"],
    "companyNumber": "string",
    "companyVision": "string",
    "endInDate": "string"
  }
]
```

1.2 Create Campaign

- Endpoint: `../api/campaign`
- Method: `POST`
- Description: Create a new campaign.
- Request Body: JSON object containing campaign details
- Response:
 - Status: 201 Created
 - Message: Success message if the campaign is created successfully

1.3 Get Campaign by ID

- Endpoint: `../api/campaign/<id>`
- Method: `GET`
- Description: Retrieve campaign details by specific ID.
- Response:
 - Status: 200 OK
 - Body: Campaign object

Sample Response:

```
{
  "_id": "ObjectId",
  "id": "String",
  "name": "string",
  "status": "string",
  "description": "string",
  "pictureFiles": ["UploadThingPictureFile"],
  "companyName": "string",
  ...
}
```

1.4 Update Campaign

- Endpoint: `../api/campaign/<id>`
- Method: `PUT`
- Description: Update a campaign by specific ID.
- Request Body: JSON object with updated campaign details
- Response:
 - Status: 200 OK
 - Message: Success message if updated successfully

1.5 Delete Campaign

- Endpoint: `../api/campaign/<id>`

- Method: **DELETE**
 - Description: Delete a campaign by specific ID.
 - Response:
 - Status: 200 OK
 - Message: Success message if deleted successfully
-

2. Statement API Group

2.1 Get All Statements

- Endpoint: **../api/statements**
- Method: **GET**
- Description: Retrieve all statement data.
- Response:
 - Status: 200 OK
 - Body: Array of Statement objects

Sample Response:

```
[
  {
    "_id": "ObjectId",
    "statement_id": "string",
    "user_id": "string",
    "campaign_id": "string",
    "campaignName": "string",
    "amount": 1000,
    "session_id": "string",
    "date": "string",
    "successAt": "string",
    "status": "string"
  }
]
```

2.2 Get Statement by ID

- Endpoint: `../api/payment/statement/<id>`
- Method: `GET`
- Description: Retrieve a statement by specific ID.
- Response:
 - Status: 200 OK
 - Body: Statement object

2.3 Delete Statement

- Endpoint: `../api/statement/<id>`
- Method: `DELETE`
- Description: Delete a statement by specific ID.
- Response:
 - Status: 200 OK
 - Message: Success message if deleted successfully

2.4 Get Statements by User ID

- Endpoint: `../api/statement/byuserid/<id>`
 - Method: `GET`
 - Description: Retrieve statements by specific user ID.
 - Response:
 - Status: 200 OK
 - Body: Statement object
-

3. User API Group

3.1 Get All Users

- Endpoint: `../api/users`
- Method: `GET`
- Description: Retrieve all user data.
- Response:
 - Status: 200 OK
 - Body: Array of User objects

Sample Response:

```
[
  {
    "user_id": "string",
    "name": "string",
    "email": "string",
    "email_verified": true,
    "family_name": "string",
    "given_name": "string",
    "nickname": "string",
    "picture": "string",
    "created_at": "string",
    "updated_at": "string",
    "last_login": "string",
    "last_ip": "string",
    "logins_count": 10
  }
]
```

3.2 Get User by ID

- Endpoint: `../api/user/<id>`
- Method: `GET`
- Description: Retrieve a user by specific ID.
- Response:
 - Status: 200 OK
 - Body: User object

3.3 Update User Data

- Endpoint: `../api/user/patch`
 - Method: `POST`
 - Description: Update user data.
 - Request Body: JSON object with updated user details
 - Response:
 - Status: 200 OK
 - Message: Success message if updated successfully
-

4. Authentication API Group

4.1 Login

- Endpoint: `../api/auth/login`
- Method: `GET`
- Description: Redirects to login page.

4.2 Logout

- Endpoint: `../api/auth/logout`
 - Method: `GET`
 - Description: Redirects to logout page.
-

5. Payment API Group

5.1 Payment Checkout

- Endpoint: `../api/payment/checkout`
- Method: `POST`
- Description: Initiates payment checkout process.
- Response:
 - Status: 302 Redirect
 - Message: Redirects to the payment checkout page

5.2 Payment Webhook

- Endpoint: `../api/payment/webhook`
 - Method: `POST`
 - Description: Endpoint for Stripe to confirm session updates.
 - Response:
 - Status: 200 OK
 - Message: Confirmation of the webhook received
-

6. Uploadthing API Group

6.1 Upload Pictures

- Endpoint: `../api/uploadthing`
- Method: `POST`
- Description: Uploads pictures to the cloud.
- Request Body: Form-data with image files
- Response:
 - Status: 201 Created
 - Message: Success message if uploaded successfully