Scavenger Hunt API Reference

This document provides a comprehensive guide to the Scavenger Hunt API endpoints for frontend developers. All endpoints are accessible through the base URL of the Elastic Beanstalk environment.

Base URL: `http://scavenger-hunt-env.eba-2dvq4pug.us-east-1.elasticbeanstalk.com/api`

Table of Contents

Users

Create a User

Creates a new user in the system.

- **URL**: `/users`
- **Method**: `POST`
- **Content-Type**: `application/json`

Request Body:

```
{
  "name": "John Doe",
  "email": "john.doe@example.com"
}
```

Success Response: - **Code**: 201 Created - **Content**:

```
{
  "message": "User created successfully",
  "user": {
    "userId": "550e8400-e29b-41d4-a716-446655440000",
    "name": "John Doe",
    "email": "john.doe@example.com"
  }
}
```

Error Responses: - **Code**: 400 Bad Request - **Content**: `{ "message": "User with this email already exists" }` - **Code**: 500 Internal Server Error - **Content**: `{ "message": "Failed to create user" }`

Get User by ID

Retrieves a user by their ID.

- **URL**: `/users/:userId`
- **Method**: `GET`
- **URL Parameters**: `userId=[string]`

Success Response: - **Code**: 200 OK - **Content**:

```json
{
  "userId": "550e8400-e29b-41d4-a716-446655440000",
  "name": "John Doe",
  "email": "john.doe@example.com"
}
```

Error Responses: - **Code**: 404 Not Found - **Content**: `{ "message": "User with ID 550e8400-e29b-41d4-a716-446655440000 not found" }` - **Code**: 500 Internal Server Error - **Content**: `{ "message": "Failed to get user" }`

Teams

Create a Team

Creates a new team and assigns the creator as the team owner.

- **URL**: `/teams`
- **Method**: `POST`
- **Content-Type**: `application/json`

Request Body:

```json
{
  "name": "Team Awesome",
  "userId": "550e8400-e29b-41d4-a716-446655440000",
  "categoryId": "550e8400-e29b-41d4-a716-446655440001"
}
```

Success Response: - **Code**: 201 Created - **Content**:

```json
{
  "message": "Team created successfully",
  "team": {
    "teamId": "550e8400-e29b-41d4-a716-446655440002",
    "name": "Team Awesome",
    "code": "ABC123",
    "categoryId": "550e8400-e29b-41d4-a716-446655440001"
  }
}
```

Error Responses: - **Code**: 400 Bad Request - **Content**: `{ "message": "Team with name Team Awesome already exists" }` - **Code**: 404 Not Found - **Content**: `{ "message": "Category with ID 550e8400-e29b-41d4-a716-446655440001 not found" }` - **Code**: 500 Internal Server Error - **Content**: `{ "message": "Failed to create team" }`

List All Teams

Retrieves a list of all teams.

- **URL**: `/teams`
- **Method**: `GET`

Success Response: - **Code**: 200 OK - **Content**:

```
[
  {
    "teamId": "550e8400-e29b-41d4-a716-446655440002",
    "name": "Team Awesome",
    "code": "ABC123",
    "categoryId": "550e8400-e29b-41d4-a716-446655440001"
  },
  {
    "teamId": "550e8400-e29b-41d4-a716-446655440003",
    "name": "Team Fantastic",
    "code": "XYZ789",
    "categoryId": "550e8400-e29b-41d4-a716-446655440001"
  }
]
```

Error Response: - **Code**: 500 Internal Server Error - **Content**: `{ "message": "Failed to list teams" }`

Join a Team

Adds a user to an existing team using the team code.

- **URL**: `/teams/join`
- **Method**: `POST`
- **Content-Type**: `application/json`

Request Body:

```
{
  "userId": "550e8400-e29b-41d4-a716-446655440000",
  "teamId": "550e8400-e29b-41d4-a716-446655440002",
  "code": "ABC123"
}
```

Success Response: - **Code**: 200 OK - **Content**:

```
{
  "message": "User added to team successfully"
}
```

Error Responses: - **Code**: 400 Bad Request - **Content**: `{ "message": "User is already a member of this team" }` - **Content**: `{ "message": "Team is already completed" }` - **Content**: `{ "message": "Invalid team code" }` - **Code**: 404 Not Found - **Content**: `{ "message": "Team with ID 550e8400-e29b-41d4-a716-446655440002 not found" }` - **Code**: 500 Internal Server Error - **Content**: `{ "message": "Failed to join team" }`

Get Team Items

Retrieves the items assigned to a specific team.

- **URL**: `/teams/:teamId/items`
- **Method**: `GET`
- **URL Parameters**: `teamId=[string]`

Success Response: - **Code**: 200 OK - **Content**:

```
{
  "team": {
    "teamId": "550e8400-e29b-41d4-a716-446655440002",
    "name": "Team Awesome",
    "code": "ABC123",
    "categoryId": "550e8400-e29b-41d4-a716-446655440001"
  },
  "items": [
    {
      "itemId": "550e8400-e29b-41d4-a716-446655440004",
      "name": "Red Ball",
      "description": "Find a red ball",
      "points": 10,
      "found": false
    },
    {
      "itemId": "550e8400-e29b-41d4-a716-446655440005",
      "name": "Blue Pen",
      "description": "Find a blue pen",
      "points": 5,
      "found": true
    }
  ]
}
```

Error Responses: - **Code**: 404 Not Found - **Content**: `{ "message": "Team with ID 550e8400-e29b-41d4-a716-446655440002 not found" }` - **Code**:

500 Internal Server Error - **Content**: `{ "message": "Failed to get team items" }`

Categories

List All Categories

Retrieves a list of all scavenger hunt categories.

- **URL**: `/categories`
- **Method**: `GET`

Success Response: - **Code**: 200 OK - **Content**:

```json
[
  {
    "categoryId": "550e8400-e29b-41d4-a716-446655440001",
    "name": "Office Items",
    "description": "Find items commonly found in an office"
  },
  {
    "categoryId": "550e8400-e29b-41d4-a716-446655440006",
    "name": "Outdoor Adventure",
    "description": "Find items in nature and outdoors"
  }
]
```

Error Response: - **Code**: 500 Internal Server Error - **Content**: `{ "message": "Failed to list categories" }`

Get Category by ID

Retrieves a specific category by its ID.

- **URL**: `/categories/:categoryId`
- **Method**: `GET`
- **URL Parameters**: `categoryId=[string]`

Success Response: - **Code**: 200 OK - **Content**:

```json
{
  "categoryId": "550e8400-e29b-41d4-a716-446655440001",
  "name": "Office Items",
  "description": "Find items commonly found in an office"
}
```

Error Responses: - **Code**: 404 Not Found - **Content**: `{ "message": "Category with ID 550e8400-e29b-41d4-a716-446655440001 not found" }` - **Code**: 500 Internal Server Error - **Content**: `{ "message": "Failed to get category" }`

Items

Get Item by ID

Retrieves a specific item by its ID.

- **URL**: `/items/:itemId`
- **Method**: `GET`
- **URL Parameters**: `itemId=[string]`

Success Response: - **Code**: 200 OK - **Content**:

```
{
  "itemId": "550e8400-e29b-41d4-a716-446655440004",
  "name": "Red Ball",
  "description": "Find a red ball",
  "points": 10,
  "categoryId": "550e8400-e29b-41d4-a716-446655440001"
}
```

Error Responses: - **Code**: 404 Not Found - **Content**: `{ "message": "Item with ID 550e8400-e29b-41d4-a716-446655440004 not found" }` - **Code**: 500 Internal Server Error - **Content**: `{ "message": "Failed to fetch item details" }`

Upload

Upload Item Image

Uploads an image for a specific team's item.

- **URL**: `/upload/:teamId/:itemId`
- **Method**: `POST`
- **Content-Type**: `multipart/form-data`
- **URL Parameters**:
  - `teamId=[string]`
  - `itemId=[string]`

Request Body: - Form data with a file field named `image`

Success Response: - **Code**: 200 OK - **Content**:

```
{
  "message": "Image uploaded successfully",
  "imageUrl": "https://scavenger-hunt-images-bucket.s3.amazonaws.com/550e8400-e29b-41d4-a716
  "matchResult": {
    "isMatch": true,
    "requiredLabels": ["ball", "red"],
    "detectedLabels": ["ball", "red", "round", "toy"]
  }
}
```

Error Responses: - **Code**: 400 Bad Request - **Content**: `{ "message": "No image file provided" }` - **Code**: 404 Not Found - **Content**: `{ "message": "Item with ID 550e8400-e29b-41d4-a716-446655440004 not found" }` - **Code**: 500 Internal Server Error - **Content**: `{ "message": "Failed to upload image" }`

Seed

The seed endpoints are primarily for development and testing purposes. They allow for populating the database with sample data.

Seed Categories and Items

Populates the database with predefined categories and items.

- **URL**: `/seed`
- **Method**: `POST`

Success Response: - **Code**: 200 OK - **Content**:

```
{
  "message": "Database seeded successfully",
  "categories": 2,
  "items": 10
}
```

Error Response: - **Code**: 500 Internal Server Error - **Content**: `{ "message": "Failed to seed database" }`

Health Check

Check API Health

Checks if the API is running correctly.

- **URL**: `/health`
- **Method**: `GET`

Success Response: - **Code**: 200 OK - **Content**:

```
{
  "status": "ok"
}
```

Error Handling

All endpoints follow a consistent error response format:

```
{
  "message": "Description of the error"
}
```

Common HTTP status codes: - `200` - Success - `201` - Created - `400` - Bad Request (client error) - `404` - Not Found - `500` - Internal Server Error (server error)

Authentication

The current API implementation does not include authentication. All endpoints are publicly accessible. In a production environment, it is recommended to implement proper authentication and authorization mechanisms.