API Documentation

This the endpoints documentation showing the request and response.

Character Endpoints

1. Get All Characters

URL: `/characters`

Method: `GET`

Description: Retrieve all characters, with optional sorting and filtering.

Query Parameters:

- `sortBy` (string, optional) - Field to sort by (e.g., `first_name`, `last_name`).
- `order` (`ASC` | `DESC`, optional) - Sort order (default is `ASC`).
- `gender` (string, optional) - Filter by gender.
- `status` (string, optional) - Filter by status.
- `location` (string, optional) - Filter by location

Response Body:

```
[
  {
    "id": 1,
    "first_name": "John",
    "last_name": "Doe",
    "status": "ACTIVE",
    "state_of_origin": "California",
    "gender": "MALE",
    "location": {
      "name": "Earth",
      "latitude": 40.7128,
      "longitude": -74.0060,
      "created": "2023-06-04T12:34:56.000Z"
    },
    "episodes": [
      {
        "name": "Episode 1",
        "release_date": "2023-05-01T00:00:00.000Z",
        "episode_code": "E01",
        "characters": [],
        "episode_comments": [],
        "created_at": "2023-05-01T00:00:00.000Z"
      }
    ],
    "created_at": "2023-06-04T12:34:56.000Z"
  }
]
```

2. Get Character by ID

URL: `/characters/:id`

Method: `GET`

Description: Retrieve a character by its ID.

Response Body:

```
{
  "id": 1,
  "first_name": "John",
  "last_name": "Doe",
  "status": "ACTIVE",
  "state_of_origin": "California",
  "gender": "MALE",
  "location": {
    "name": "Earth",
    "latitude": 40.7128,
    "longitude": -74.0060,
    "created": "2023-06-04T12:34:56.000Z"
  },
  "episodes": [
    {
      "name": "Episode 1",
      "release_date": "2023-05-01T00:00:00.000Z",
      "episode_code": "E01",
      "characters": [],
      "episode_comments": [],
      "created_at": "2023-05-01T00:00:00.000Z"
    }
  ],
  "created_at": "2023-06-04T12:34:56.000Z"
}
```

3. Create Character

URL: `/characters`

Method: `POST`

Description: Create a new character.

Request Body:

```
{
  "first_name": "John",
  "last_name": "Doe",
  "status": "ACTIVE",
  "state_of_origin": "California",
  "gender": "MALE",
  "location": {
    "name": "Earth",
    "latitude": 40.7128,
    "longitude": -74.0060,
    "created": "2023-06-04T12:34:56.000Z"
  },
  "episodeIds": [1, 2]
}
```

Response Body:

```
{
  "id": 1,
  "first_name": "John",
  "last_name": "Doe",
  "status": "ACTIVE",
  "state_of_origin": "California",
  "gender": "MALE",
  "location": {
    "name": "Earth",
    "latitude": 40.7128,
    "longitude": -74.0060,
    "created": "2023-06-04T12:34:56.000Z"
  },
  "episodes": [
    {
      "name": "Episode 1",
      "release_date": "2023-05-01T00:00:00.000Z",
      "episode_code": "E01",
      "characters": [],
      "episode_comments": [],
      "created_at": "2023-05-01T00:00:00.000Z"
    }
  ],
  "created_at": "2023-06-04T12:34:56.000Z"
}
```

4. Update Character

URL: `/characters/:id`

Method: `PUT`

Description: Update an existing character.

Request Body:

```
{
  "first_name": "John",
  "last_name": "Doe",
  "status": "ACTIVE",
  "state_of_origin": "California",
  "gender": "MALE",
  "location": {
    "name": "Earth",
    "latitude": 40.7128,
    "longitude": -74.0060,
    "created": "2023-06-04T12:34:56.000Z"
  },
  "episodeIds": [1, 2]
}
```

Response Body:

```
{
  "id": 1,
  "first_name": "John",
  "last_name": "Doe",
  "status": "ACTIVE",
  "state_of_origin": "California",
  "gender": "MALE",
  "location": {
    "name": "Earth",
    "latitude": 40.7128,
    "longitude": -74.0060,
    "created": "2023-06-04T12:34:56.000Z"
  },
  "episodes": [
    {
      "name": "Episode 1",
      "release_date": "2023-05-01T00:00:00.000Z",
      "episode_code": "E01",
      "characters": [],
      "episode_comments": [],
      "created_at": "2023-05-01T00:00:00.000Z"
    }
  ],
  "created_at": "2023-06-04T12:34:56.000Z"
}
```

5. Delete Character

URL: `/characters/:id`

Method: `DELETE`

Description: Delete a character by its ID.

Response Body: `204 No Content`

Episode Endpoints

1. Get All Episodes Sorted by Release Date

URL: `/episodes`

Method: `GET`

Description: Retrieve all episodes sorted by their release date.

Response Body:

```
[
  {
    "id": 1,
    "name": "Episode 1",
    "release_date": "2023-05-01T00:00:00.000Z",
    "episode_code": "E01",
    "characters": [
      {
        "id": 1,
        "first_name": "John",
        "last_name": "Doe",
        "status": "ACTIVE",
        "state_of_origin": "California",
        "gender": "MALE",
        "location": {
          "name": "Earth",
          "latitude": 40.7128,
          "longitude": -74.0060,
          "created": "2023-06-04T12:34:56.000Z"
        },
        "episodes": [],
        "created_at": "2023-06-04T12:34:56.000Z"
      }
    ],
    "episode_comments": [
      {
        "id": 1,
        "comment": "Great episode!",
        "ip_address_location": "192.168.1.1",
        "created_at": "2023-06-04T12:34:56.000Z"
      }
    ],
    "created_at": "2023-05-01T00:00:00.000Z"
  }
]
```

2. Get Episodes by Character ID

URL: `/episodes/character/:characterId`

Method: `GET`

Description: Retrieve all episodes that a specific character appears in.

Response Body:

```
[
  {
    "id": 1,
    "name": "Episode 1",
    "release_date": "2023-05-01T00:00:00.000Z",
    "episode_code": "E01",
    "characters": [
      {
        "id": 1,
        "first_name": "John",
        "last_name": "Doe",
  }
        "status": "ACTIVE",
        "state_of_origin": "California",
        "gender": "MALE",
        "location": {
          "name": "Earth",
          "latitude": 40.7128,
          "longitude": -74.0060,
          "created": "2023-06-04T12:34:56.000Z"
        },
        "episodes": [],
        "created_at": "2023-06-04T12:34:56.000Z"
      }
    ],
    "episode_comments": [
      {
        "id": 1,
        "comment": "Great episode!",
        "ip_address_location": "192.168.1.1",
        "created_at": "2023-06-04T12:34:56.000Z"
      }
    ],
    "created_at": "2023-05-01T00:00:00.000Z"
  }
```

```
]
```

3. Add Comment to Episode

URL: `/episodes/:id/comment`

Method: `POST`

Description: Add a comment to a specific episode.

Request Body:

```
{
  "comment": "Amazing episode!",
  "ip_address_location": "192.168.1.1"
}
```

Response Body:

```
{
  "id": 1,
  "name": "Episode 1",
  "release_date": "2023-05-01T00:00:00.000Z",
  "episode_code": "E01",
  "characters": [
    {
      "id": 1,
      "first_name": "John",
      "last_name": "Doe",
    "status": "ACTIVE",
    "state_of_origin": "California",
    "gender": "MALE",
    "location": {
      "name": "Earth",
      "latitude": 40.7128,
      "longitude": -74.0060,
      "created": "2023-06-04T12:34:56.000Z"
    },
    "episodes": [],
    "created_at": "2023-06-04T12:34:56.000Z"
  }
],
"episode_comments": [
  {
    "id": 1,
    "comment": "Great episode!",
    "ip_address_location": "192.168.1.1",
    "created_at": "2023-06-04T12:34:56.000Z"
  },
  {
    "id": 2,
    "comment": "Amazing episode!",
      "ip_address_location": "192.168.1.1",
      "created_at": "2023-06-04T12:34:56.000Z"
    }
  ],
  "created_at": "2023-05-01T00:00:00.000Z"
}
```

Comment Endpoints

1. Get All Comments Sorted by Creation Date

URL: /comments

Method: GET

Description: Retrieve all comments sorted by their creation date in descending order.

Response Body:

```
[
  {
    "id": 1,
    "comment": "Great episode!",
    "ip_address_location": "192.168.1.1",
    "created_at": "2023-06-04T12:34:56.000Z",
    "episode": {
      "id": 1,
      "name": "Episode 1",
      "release_date": "2023-05-01T00:00:00.000Z",
      "episode_code": "E01",
      "created_at": "2023-05-01T00:00:00.000Z"
    }
  },
  {
    "id": 2,
    "comment": "Amazing episode!",
    "ip_address_location": "192.168.1.2",
    "created_at": "2023-06-05T12:34:56.000Z",
    "episode": {
      "id": 2,
      "name": "Episode 2",
      "release_date": "2023-06-01T00:00:00.000Z",
      "episode_code": "E02",
      "created_at": "2023-06-01T00:00:00.000Z"
    }
  }
]
```

Error Responses:

404 Not Found:

```
{
  "statusCode": 404,
  "message": "Episode not found",
  "error": "Not Found"
}
```

400 Bad Request:

```
{
  "statusCode": 400,
  "message": "Error adding comment to episode",
  "error": "Bad Request"
}
```

500 Internal Server Error:

```
{
  "statusCode": 500,
  "message": "Error retrieving comments",
  "error": "Internal Server Error"
}
```