

DevNotes Server API Documentation

Overview.....	1
Authentication.....	1
Authentication Headers.....	1
Endpoints.....	1
Authentication Endpoints.....	1
POST /signin.....	1
POST /signout.....	2
POST /validatetoken.....	2
Notes Endpoints.....	3
GET /notes.....	3
GET /notes/{id}.....	3
POST /notes.....	4
POST /notes/dto.....	4
PUT /notes/{id}.....	5
DELETE /notes/{id}.....	5
Users Endpoints.....	6
GET /users.....	6
Tags Endpoints.....	6
GET /tags.....	6
POST /tags.....	7
DELETE /tags/{id}.....	7
Data Models.....	7
DevNote.....	7
DevNoteUser.....	8
DevNoteTag.....	8
CreateDevNoteDto.....	8

Notes.....	8
------------	---

Overview

The DevNotes Server is a RESTful API for managing development notes with user authentication, tagging system, and location-based features. The server uses an ASP.NET Core Minimal API with Entity Framework Core and SQLite database.

Authentication

The API uses session-based authentication with tokens. Most endpoints require authentication except for signin, signout, validate token, and the DTO notes endpoint.

Authentication Headers

For authenticated endpoints, include the session token in your requests under the key *X-Session-Token*.

Endpoints

Authentication Endpoints

POST /signin

Authenticate a user and receive a session token.

Request Body:

```
{
  "UserName": "string",
  "Password": "string"
}
```

Response (200 OK):

```
{
  "token": "session-token-string",
  "Id": "user-guid"
}
```

Response (401 Unauthorized): Invalid credentials

POST /signout

Invalidate the current session token.

Request Body:

```
{  
  "Token": "session-token-string"  
}
```

Response (200 OK): Session terminated successfully

POST /validateToken

Validate a session token and return associated user information.

Request Body:

```
{  
  "Token": "session-token-string"  
}
```

Response (200 OK):

```
{  
  "Id": "user-guid",  
  "Name": "username"  
}
```

Response (400 Bad Request): Invalid or expired token

Response (404 Not Found): User not found

Notes Endpoints

GET /notes

Retrieve all notes with associated user and tag information.

Authentication: Required

Response (200 OK):

```
[
  {
    "Id": "note-guid",
    "Title": "Note Title",
    "Body": "Note content",
    "CreatedById": "user-guid",
    "CreatedBy": {
      "Id": "user-guid",
      "Name": "username"
    },
    "Tags": [
      {
        "Id": "tag-guid",
        "Name": "tag-name",
        "Colour": 12345
      }
    ],
    "CreatedAt": "2023-01-01T00:00:00Z",
    "LastEdited": "2023-01-01T00:00:00Z",
    "LevelPath": "path/to/level",
    "WorldX": 1.5,
    "WorldY": 2.5,
    "WorldZ": 3.5
  }
]
```

GET /notes/{id}

Retrieve a specific note by ID.

Authentication: Required

Parameters:

- id (path): GUID of the note

Response (200 OK): Single note object (same structure as above)

Response (404 Not Found): Note not found

POST /notes

Create a new note.

Authentication: Required

Request Body:

```
{
  "Id": "note-guid",
  "Title": "Note Title",
  "Body": "Note content",
  "Tags": [
    {
      "Id": "tag-guid",
      "Name": "tag-name",
      "Colour": 12345
    }
  ],
  "LevelPath": "path/to/level",
  "WorldX": 1.5,
  "WorldY": 2.5,
  "WorldZ": 3.5
}
```

Response (201 Created): Created note object with Location header

POST /notes/dto

Create a new note using simplified DTO format.

Authentication: Required (supports public key authentication)

Request Body:

```
{
  "Title": "Note Title",
  "Body": "Note content",
  "CreatedByUserName": "username",
  "TagNames": ["tag1", "tag2"],
  "LevelPath": "path/to/level",
  "WorldX": 1.5,
  "WorldY": 2.5,
  "WorldZ": 3.5
}
```

Response (201 Created): Created note object **Response (400 Bad Request):** User not found or tag not found

PUT /notes/{id}

Update an existing note (replaces all fields).

Authentication: Required

Parameters:

- id (path): GUID of the note to update

Request Body:

```
{
  "Id": "note-guid",
  "Title": "Note Title",
  "Body": "Note content",
  "Tags": [
    {
      "Id": "tag-guid",
      "Name": "tag-name",
      "Colour": 12345
    }
  ],
  "LevelPath": "path/to/level",
  "WorldX": 1.5,
  "WorldY": 2.5,
  "WorldZ": 3.5
}
```

```
"WorldZ": 3.5
}
```

Response (200 OK): Updated note object

Response (404 Not Found): Note not found

Response (400 Bad Request): One or more tags not found

DELETE /notes/{id}

Delete a note.

Authentication: Required

Parameters:

- **id (path):** GUID of the note to delete

Response (204 No Content): Note deleted successfully

Response (404 Not Found): Note not found

Users Endpoints

GET /users

Retrieve all users.

Authentication: Required

Response (200 OK):

```
[
  {
    "Id": "user-guid",
    "Name": "username"
  }
]
```

Tags Endpoints

GET /tags

Retrieve all tags.

Authentication: Required

Response (200 OK):

```
[
  {
    "Id": "tag-guid",
    "Name": "tag-name",
    "Colour": 12345
  }
]
```

POST /tags

Create a new tag.

Authentication: Required

Request Body:

```
{
  "Id": "tag-guid",
  "Name": "tag-name",
  "Colour": 12345
}
```

Response (201 Created): Created tag object with Location header

DELETE /tags/{id}

Delete a tag.

Authentication: Required

Parameters:

- **id (path):** GUID of the tag to delete

Response (204 No Content): Tag deleted successfully

Response (404 Not Found): Tag not found

Data Models

DevNote

- **Id:** GUID - Unique identifier
- **Title:** String - Note title
- **Body:** String - Note content
- **CreatedById:** GUID - ID of user who created the note
- **CreatedBy:** DevNoteUser - User object (populated in responses)
- **Tags:** Array of DevNoteTag - Associated tags
- **CreatedAt:** DateTime - Creation timestamp (UTC)
- **LastEdited:** DateTime - Last modification timestamp (UTC)
- **LevelPath:** String - Game level or location path
- **WorldX:** Float - X coordinate in world space
- **WorldY:** Float - Y coordinate in world space
- **WorldZ:** Float - Z coordinate in world space

DevNoteUser

- **Id:** GUID - Unique identifier
- **Name:** String - Username

DevNoteTag

- **Id:** GUID - Unique identifier
- **Name:** String - Tag name
- **Colour:** Integer - Color value for UI display

CreateDevNoteDto

- **Title:** String - Note title
- **Body:** String - Note content
- **CreatedByUserName:** String - Username of note creator
- **TagNames:** Array of String - Tag names to associate

- **LevelPath:** String - Game level or location path
- **WorldX:** Float - X coordinate
- **WorldY:** Float - Y coordinate
- **WorldZ:** Float - Z coordinate

Notes

- All timestamps are in UTC format
- GUIDs are used for all entity identifiers
- The server uses SQLite database with automatic migration/creation
- Session tokens persist until the server shuts down, in which case they will need to be refreshed through /signin
- Tags must be created before being associated with notes