Locatopedia

Documentation for Locatopedia backend.

GET /api/wiki/

```
{{BASE_URL}}/api/wiki?hash=4W2pF8ErT0Utxp5HLU4rikyKgVA=&version=1
```

This endpoint retrieves wiki information based on a specified hash and version. It allows users to access details about a specific wiki entry.

Request

- **Method**: GET
- **URL**: `{{BASE_URL}}/api/wiki`
- **Query Parameters**:
  - `hash` (string): A unique identifier for the wiki entry.
  - `version` (integer): The version number of the wiki entry to be retrieved.

Expected Responses

- **200 OK**: The request was successful, and the server returned the requested wiki information. The response body will contain the following fields:
  - `hash`: The unique identifier for the wiki entry.
  - `auid`: Author unique identifier.
  - `title`: The title of the wiki entry.
  - `subtitle`: The subtitle of the wiki entry.
  - `content`: The main content of the wiki entry.
  - `email_user`: The email of the user who created the entry.
  - `date`: The date when the entry was created or last updated.
  - `like_count`: The number of likes the entry has received.
  - `user_liked`: Indicates whether the current user has liked the entry.
  - `cuids`: An array of unique identifiers related to the entry.
- **204 No Content**: The request was successful, but there is no content to return. This may occur if the specified hash does not correspond to any existing wiki entry.
- **404 Not Found**: The specified wiki entry could not be found based on the provided hash and version.
- **500 Internal Server Error**: The server encountered an unexpected condition that prevented it from fulfilling the request. This indicates a

problem on the server side.

Response Structure

The response body contains the following attributes:

```json
{
  "hash": "",          // (string) The unique hash of the
  "auid": "",         // (string) The author unique identi
  "title": "",        // (string) The title of the wiki er
  "subtitle": "",     // (string) The subtitle of the wiki
  "content": "",      // (string) The main content of the
  "email_user": "",   // (string) The email of the user wh
  "date": "",         // (string) The date when the entry
  "like_count": "",   // (string) The number of likes the
  "user_liked": true, // (boolean) Indicates if the curren
  "cuids": [""]       // (array of strings) An array of ur
```

Notes

- All string fields may return empty values if not applicable.
- The `user_liked` field will return a boolean indicating the like status for the current user.
- The `cuids` array may contain multiple identifiers related to the wiki entry.

Ensure to handle each response status appropriately in your application.

PARAMS

hash	4W2pF8ErT0Utxp5HLU4rikyKgVA=
version	1

GET /api/wiki/all

{{BASE_URL}}/api/wiki/all

This endpoint retrieves all entries from the wiki. It returns a list of wiki entries, each containing various details.

Request Method: GET
Endpoint: `{{BASE_URL}}/api/wiki/all`
Expected Response Status: `200 OK`

Input Parameters

This endpoint does not require any input parameters.

Response Structure

The response is a JSON array containing objects with the following structure:

```json
[
    {
        "hash": "string",
        "title": "string",
        "subtitle": "string",
        "img_url": "string",
        "auid": "string"
    }
]
```

- **hash**: A unique identifier for the wiki entry.
- **title**: The title of the wiki entry.
- **subtitle**: A brief subtitle or description of the wiki entry.
- **img_url**: A URL pointing to an image associated with the wiki entry.
- **auid**: An identifier for the author of the wiki entry.

GET /api/wiki/validnewcuids

{{BASE_URL}}/api/wiki/validnewcuids?cuids=8a39708d4c87fff,8a39708d4c8ffff,8a39708d4ca7fff,8a39708d4caffff,8a39708d4cb7fff,8a39708d4cd7fff,8a39708d4cf7fff,8a39708d4d9ffff

This endpoint validates a list of unique identifiers (CUIDs) provided as a query parameter. It checks if the specified CUIDs are valid and returns a boolean response indicating their validity.

Request

Method: GET
URL: `{{BASE_URL}}/api/wiki/validnewcuids?cuids={cuids}`

Query Parameters:

- `cuids` (string): A comma-separated list of CUIDs to validate.

Response

Status Code: 200
Content-Type: application/json

Response Body

The response body contains a JSON object with the following structure:

```json
{
  "valid": boolean
}
```

- `valid` (boolean): Indicates whether the provided CUIDs are valid. A value of `true` signifies that all CUIDs are valid, while `false` indicates that at least one of the CUIDs is invalid.

Example Responses

- { "valid": true}
- { "valid": false}

This endpoint is useful for applications that need to verify the legitimacy of CUIDs before proceeding with further operations.

This endpoint allows you to validate a list of CUIDs (Customer Unique Identifiers) to check if they are valid.

PARAMS

cuids	8a39708d4c87fff,8a39708d4c8ffff,8a39708d4ca7fff,8a39708d4caffff,8a39708d4cb7fff,8a39708d4cd7fff,8a39708d4cf7fff,8a39708d4d9ffff

GET /api/dggstools/generate-auid-hash

```
{{BASE_URL}}/api/dggstools/generate-auid-hash?cuids=8a39708d4c87fff,8a39708d4c8ffff,8a39708d4ca7fff,8a39708d4caffff,8a39708d4cb7fff,8a39708d4cd7fff,8a39708d4cf7fff,8a39708d4d9ffff
```

This endpoint generates an AUID hash based on the provided `cuids`. It returns a `200 OK` status for successful requests. If the `cuids` parameter is not provided, a `400 Bad Request` status will be returned.

Request Parameters

- **cuids** (string, required): A comma-separated list of unique identifiers for which the AUID hash needs to be generated.

Response

On a successful request, the response will be in JSON format and will include:

- **auid_comp_b64** (string): A base64 encoded representation of the computed AUID.

- **hashed_b64** (string): A base64 encoded representation of the hashed value.

Example Response

json

```
{
  "auid_comp_b64": "",
  "hashed_b64": ""
}
```

PARAMS

cuids	8a39708d4c87fff,8a39708d4c8ffff,8a39708d4ca7fff,8a39708d4caffff,8a39708d4cb7fff,8a39708d4cd7fff,8a39708d4cf7fff,8a39708d4d9ffff cuids of cells from wich to get hash and auid

GET /api/dggstools/cuids-from-auid

{{BASE_URL}}/api/dggstools/cuids-from-auid?auid=eJyLs0g0tjQ3sEgxSbYwT0tLU7EHgTQ4yz4Rq2gSQtQ-BYmdhsS2T7EE6wAAgbMcbA==

This endpoint retrieves CUIDs associated with a given AUID. It is a GET request that requires the `auid` parameter to be included in the query string.

Request Parameters

- **auid** (string, required): The AUID for which CUIDs are to be fetched. It should be URL-encoded.

Expected Response Codes

- **200 OK**: The request was successful, and the response will contain a JSON object with an array of CUIDs.
- **400 Bad Request**: The request is not authorized if the `auid` parameter is missing.

Example Response

json

```
{
  "cuids": [""]
}
```

PARAMS

auid	eJyLs0g0tjQ3sEgxSbYwT0tLU7EHgTQ4yz4Rq2gSQtQ-BYmdhsS2T7EE6wAAgbMcbA==

GET /api/dggstools/hash

{{BASE_URL}}/api/dggstools/hash?auid=eJyLs0g0tjQ3sEgxSbYwT0tLU7EHgTQ4yz4Rq2gSQtQ-BYmdhsS2T7EE6wAAgbMcbA==

This endpoint retrieves a hashed value based on the provided `auid` parameter. It is used to generate a hash that can be utilized for various purposes, such as data integrity verification.

Request Parameters

- **auid** (string, required): A unique identifier that is used to generate the hash. It must be provided in the query string of the request.

Expected Response Codes

- **200 OK**: The request was successful, and the response will contain the hashed value.
- **400 Bad Request**: The request is not authorized due to the absence of the `auid` parameter.

Response Schema

The response will be in JSON format and follows the schema below:

```json
{
  "hashed_b64": "string"
}
```

- **hashed_b64**: A base64-encoded string representing the hashed value. This field may be empty if the hashing process did not yield a result.

PARAMS

auid	eJyLs0g0tjQ3sEgxSbYwT0tLU7EHgTQ4yz4Rq2gSQtQ-BYmdhsS2T7EE6wAAgbMcbA==

POST /api/wiki/auth/like 🔒

{{BASE_URL}}/api/wiki/auth/like

This endpoint allows users to toggle like a specific wiki entry by providing a unique hash identifier.

Request

Method: POST
URL: `{{BASE_URL}}/api/wiki/auth/like`

Request Body

The request body must be in JSON format and include the following parameters:

- **hash** (string): A unique identifier for the wiki entry that the user wishes to like.
- **email** (string): The email address of the user performing the action.

Example Request Body

```json
{
  "hash": "TlDy5lMOz4sRH-Ulfvj28eveDFs=",
  "email": "841723@unizar.es"
}
```

Expected Responses

- **204 No Content**: This response indicates that the like action was successful and the specified hash was found.
- **404 Not Found**: This response indicates that the provided hash does not correspond to any existing wiki entry.

Response Schema

The response for a successful like action (204) does not contain a body. However, it is important to note that no additional data is returned in the case of a successful request.

```json
{
  "status": "204",
  "message": "No Content"
}
```

AUTHORIZATION Bearer Token

Token	{{vault:AUTH_TOKEN}}

Body raw (json)

json

```
{
    "hash": "TlDy5lMOz4sRH-Ulfvj28eveDFs=",
    "email": "841723@unizar.es"
}
```

GET /api/wiki/auth/liked 🔒

{{BASE_URL}}/api/wiki/auth/liked?limit=10

This endpoint retrieves a list of liked wiki entries for the authenticated user, with an optional limit on the number of entries returned.

Request

Method: GET
URL: `{{BASE_URL}}/api/wiki/auth/liked?limit=10`

Expected Responses

- **200 OK**: This response indicates that the request was successful and returns a JSON array of liked wiki entries.

Response Schema

The response is an array of objects, each representing a liked wiki entry. Each object contains the following fields:

- **hash** (string): A unique identifier for the liked wiki entry.
- **title** (string): The title of the liked wiki entry.
- **subtitle** (string): A subtitle or description of the liked wiki entry.
- **img_url** (string): A URL pointing to an image associated with the liked wiki entry.

Example Response

json

```
[
  {
    "hash": "",
    "title": "",
    "subtitle": "",
    "img_url": ""
  }
]
```

Request

Method: GET
URL: `{{BASE_URL}}/api/wiki/auth/liked?limit=10`

Query Parameters

- **limit** (integer, required): The maximum number of liked wiki entries to return. This parameter controls how many entries are included in the response.

Expected Responses

- **200 OK**: This response indicates that the request was successful and returns a list of liked wiki entries.

Response Schema

The response contains an array of objects, each representing a liked wiki entry with the following structure:

json

```
[
  {
    "hash": "",
    "title": "",
    "subtitle": "",
    "img_url": ""
  }
]
```

- **hash** (string): A unique identifier for the liked wiki entry.
- **title** (string): The title of the liked wiki entry.
- **subtitle** (string): A brief description or subtitle of the liked wiki entry.
- **img_url** (string): The URL of the image associated with the liked wiki entry.

Example Response

json

```
[
  {
    "hash": "exampleHash",
    "title": "Example Title",
    "subtitle": "Example Subtitle",
    "img_url": "http://example.com/image.jpg"
  }
]
```

Request

Method: GET
URL: `{{BASE_URL}}/api/wiki/auth/liked?limit=10`

Expected Responses

- **200 OK**: This response indicates that the request was successful and returns an array of liked wiki entries.

Response Schema

The response contains an array of objects, each representing a liked wiki entry with the following properties:

- **hash** (string): A unique identifier for the wiki entry.
- **title** (string): The title of the wiki entry.
- **subtitle** (string): A subtitle or additional information about the wiki entry.
- **img_url** (string): A URL pointing to an image associated with the wiki entry.

Example Response

json

```json
[{"hash":"","title":"","subtitle":"","img_url":""}]
```

AUTHORIZATION Bearer Token

Token	{{vault:AUTH_TOKEN}}

PARAMS

limit	10

GET /api/wiki/auth/created

{{BASE_URL}}/api/wiki/auth/created?limit=10

This endpoint retrieves a list of created wiki entries for the authenticated user, with an optional limit on the number of entries returned.

Request

Method: GET
URL: `{{BASE_URL}}/api/wiki/auth/created?limit=10`

Query Parameters

- **limit** (integer, optional): The maximum number of created wiki entries to return. This parameter controls how many entries are included in the response.

Expected Responses

- **200 OK**: This response indicates that the request was successful and returns a JSON array of created wiki entries.

Response Schema

The response contains an array of objects, each representing a created wiki entry with the following structure:

```json
[
  {
    "hash": "",
    "title": "",
    "subtitle": "",
    "img_url": ""
  }
]
```

- **hash** (string): A unique identifier for the created wiki entry.
- **title** (string): The title of the created wiki entry.
- **subtitle** (string): A brief description or subtitle of the created wiki entry.
- **img_url** (string): The URL of the image associated with the created wiki entry.

Example Response

```json
[
  {
    "hash": "exampleHash",
    "title": "Example Title",
    "subtitle": "Example Subtitle",
    "img_url": "http://example.com/image.jpg"
  }
]
```

Request

Method: GET
URL: `{{BASE_URL}}/api/wiki/auth/created?limit=10`

Query Parameters

- **limit** (integer, required): The maximum number of created wiki entries to return. This parameter controls how many entries are included in the response.

Expected Responses

- **200 OK**: This response indicates that the request was successful and returns a JSON array of created wiki entries.

Response Schema

The response is an array of objects, each representing a created wiki entry. Each object contains the following fields:

- **hash** (string): A unique identifier for the created wiki entry.
- **title** (string): The title of the created wiki entry.
- **subtitle** (string): A subtitle or description of the created wiki entry.
- **img_url** (string): A URL pointing to an image associated with the created wiki entry.

Example Response

```json
[
  {
    "hash": "",
    "title": "",
    "subtitle": "",
    "img_url": ""
  }
]
```

This endpoint retrieves a list of created wiki entries for the authenticated user, with an optional limit on the number of entries returned.

Request

Method: GET
URL: `{{BASE_URL}}/api/wiki/auth/created?limit=10`

Query Parameters

- **limit** (integer, required): The maximum number of created wiki entries to return. This parameter controls how many entries are included in the response.

Expected Responses

- **200 OK**: This response indicates that the request was successful and returns a JSON array of created wiki entries.

Response Schema

The response is an array of objects, each representing a created wiki entry. Each object contains the following fields:

- **hash** (string): A unique identifier for the created wiki entry.
- **title** (string): The title of the created wiki entry.

- **subtitle** (string): A subtitle or description of the created wiki entry.
- **img_url** (string): A URL pointing to an image associated with the created wiki entry.

Example Response

json

```
[
  {
    "hash": "",
    "title": "",
    "subtitle": "",
    "img_url": ""
  }
]
```

AUTHORIZATION Bearer Token

Token	{{vault:AUTH_TOKEN}}

PARAMS

limit	10

GET /api/wiki/auth/edited

{{BASE_URL}}/api/wiki/auth/edited?limit=10

This endpoint retrieves a list of edited wiki entries for the authenticated user, with an optional limit on the number of entries returned.

Request

Method: GET
URL: `{{BASE_URL}}/api/wiki/auth/edited?limit=10`

Query Parameters

- **limit** (integer, required): The maximum number of edited wiki entries to return. This parameter controls how many entries are included in the response.

Expected Responses

- **200 OK**: This response indicates that the request was successful and returns a JSON array of edited wiki entries.

Response Schema

The response is an array of objects, each representing an edited wiki entry. Each object contains the following fields:

- **hash** (string): A unique identifier for the edited wiki entry.
- **title** (string): The title of the edited wiki entry.
- **subtitle** (string): A subtitle or description of the edited wiki entry.
- **img_url** (string): A URL pointing to an image associated with the edited wiki entry.

Example Response

```json
[
  {
    "hash": "",
    "title": "",
    "subtitle": "",
    "img_url": ""
  }
]
```

Request

Method: GET
URL: `{{BASE_URL}}/api/wiki/auth/edited?limit=10`

Query Parameters

- **limit** (integer, required): The maximum number of edited wiki entries to return. This parameter controls how many entries are included in the response.

Expected Responses

- **200 OK**: This response indicates that the request was successful and returns a JSON array of edited wiki entries.

Response Schema

The response is an array of objects, each representing an edited wiki entry. Each object contains the following fields:

- **hash** (string): A unique identifier for the edited wiki entry.
- **title** (string): The title of the edited wiki entry.
- **subtitle** (string): A subtitle or description of the edited wiki entry.
- **img_url** (string): A URL pointing to an image associated with the edited wiki entry.

Example Response

```json
```

```
[
  {
    "hash": "",
    "title": "",
    "subtitle": "",
    "img_url": ""
  }
]
```

Request

Method: GET
URL: `{{BASE_URL}}/api/wiki/auth/edited?limit=10`

Query Parameters

- **limit** (integer, required): The maximum number of edited wiki entries to return. This parameter controls how many entries are included in the response.

Expected Responses

- **200 OK**: This response indicates that the request was successful and returns a JSON array of edited wiki entries.

Response Schema

The response is an array of objects, each representing an edited wiki entry. Each object contains the following fields:

- **hash** (string): A unique identifier for the edited wiki entry.
- **title** (string): The title of the edited wiki entry.
- **subtitle** (string): A subtitle or description of the edited wiki entry.
- **img_url** (string): A URL pointing to an image associated with the edited wiki entry.

Example Response

```json
[
  {
    "hash": "",
    "title": "",
    "subtitle": "",
    "img_url": ""
  }
]
```

AUTHORIZATION Bearer Token

Token	{{vault:AUTH_TOKEN}}

PARAMS

limit	10

PUT /api/wiki/auth/newversion

{{BASE_URL}}/api/wiki/auth/newversion

This endpoint allows users to update a new version of a wiki entry. It requires a JSON payload containing the details of the wiki entry to be updated.

Request

Method: PUT
URL: `{{BASE_URL}}/api/wiki/auth/newversion`

Request Body:

json

```
{
  "hash": "string",       // The unique identifier for the
  "title": "string",      // The title of the wiki entry (
  "subtitle": "string",   // A brief subtitle for the wiki
  "content": "string",    // The main content of the wiki
  "email_user": "string"  // The email of the user making
}
```

Expected Response Codes:

- **200 OK**: The update was successful.
- **404 Not Found**: The provided hash does not correspond to an existing wiki entry.

Response

Upon a successful update, the response will return a JSON object with the following structure:

json

```
{
  "title": "string",      // The updated title of the wiki
  "subtitle": "string",   // The updated subtitle of the w
  "content": "string",    // The updated content of the wi
  "email_user": "string", // The email of the user who up
  "date": "string"        // The date of the update
}
```

The response will include the updated details of the wiki entry, confirming the changes made.

AUTHORIZATION Bearer Token

Token	{{vault:AUTH_TOKEN}}

Body raw (json)

json

```
{"hash":"HJel76uziquFk5Lfo0l6zdODn3Y=","title":"Expo Zarag
```

POST /api/wiki/auth/add

{{BASE_URL}}/api/wiki/auth/add

This endpoint allows users to add a new entry to the wiki. Upon successful creation, it returns a status code of **201**. In case of an internal server error, it returns a status code of **500**.

Request Parameters

The request body must be in JSON format and contain the following parameters:

- **cuids** (array of strings): A list of unique identifiers associated with the user or content.
- **title** (string): The title of the wiki entry.
- **subtitle** (string): A brief subtitle or description of the entry.
- **content** (string): The main content of the wiki entry, which can include Markdown formatting.
- **imgData** (string): Base64 encoded image data to be associated with the entry.
- **emailUser** (string): The email address of the user submitting the entry.

Response Schema

On successful creation, the response will be an empty JSON array with a status code of **201**. In case of an error, the response will reflect the internal error.

Example of a typical response:

json

```
{
  "auid": "",
  "title": "",
  "subtitle": "",
  "content": "",
  "img_url": "",
  "email_user": "",
  "hash": ""
}
```

Make sure to provide valid data for each of the parameters to ensure successful processing of the request.

AUTHORIZATION Bearer Token

Token	{{vault:AUTH_TOKEN}}

Body raw (json)

json

```
{"cuids":["8a3970889697fff","8a397088bb5ffff","8a397088ba6
```

DELETE /api/wiki/auth/delete

{{BASE_URL}}/api/wiki/auth/delete

This endpoint is used to delete a specific wiki entry identified by its hash. The request requires a JSON payload containing the `hash` of the entry to be deleted.

Request Body

- **hash** (string): The unique identifier for the wiki entry that you want to delete.

Expected Response Codes

- **204 No Content**: Indicates that the deletion was successful and there is no content to return.
- **401 Unauthorized**: Indicates that the request was made by a user who is neither an admin nor the creator of the wiki entry.

Response Schema

The response for a successful deletion does not contain a body, hence the response schema is as follows:

json

```
{
  "status": "204",
  "message": "No Content"
}
```

Example Request

json

```json
{
  "hash": "HJel76uziquFk5Lfo0l6zdODn3Y="
}
```

AUTHORIZATION Bearer Token

Token	{{vault:AUTH_TOKEN}}

Body raw (json)

json

```json
{
    "hash":"HJel76uziquFk5Lfo0l6zdODn3Y="
}
```