# API

Since Tiki24 an API has been available, leveraging swagger-api/swagger-ui

See it in action here: https://doc.tiki.org/api/

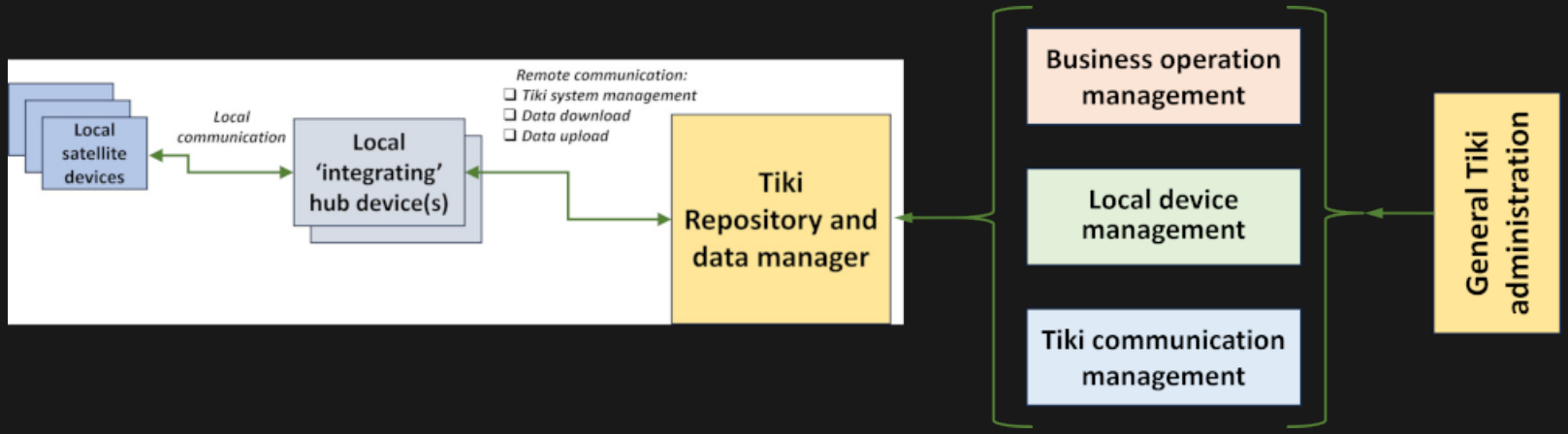
First commit: https://gitlab.com/tikiwiki/tiki/-/merge_requests/1028

# TIKI 27+

Significant updates made in Tiki27, particularly to support Internet of Things (IoT) deployments, enhancing and adding support for:

- Trackers
- File galleries

The schematic below illustrates a generic IoT system with Tiki at its centre, where the left-hand side shows the different types of field deployed devices and their communications, and the right-hand side segments the different types of Tiki user that can each be supported with various types of reporting and analysis:




More information on the use of the API for IoT deployment can be found here with:

- details about example software for field deployed devices that automates the upload of data to Tiki using the API, and how
- customised reporting and analysis can be configured with, for example, automated notification emails for sensed alarm conditions.

## TIKI 24+

A self-documented REST API is available since Tiki 24. This new feature is exposing the most commonly used elements of the system, notably:

- Categories
- Comments
- Groups
- Search
- Trackers
- Translation
- Users
- and Wiki

To start using Tiki API, you may need to refer to this documentation which details its endpoints.

## REQUIREMENTS

`.htaccess` file must be enabled to make the `/api/` URL work, which is standard practice to have SEFURLs.

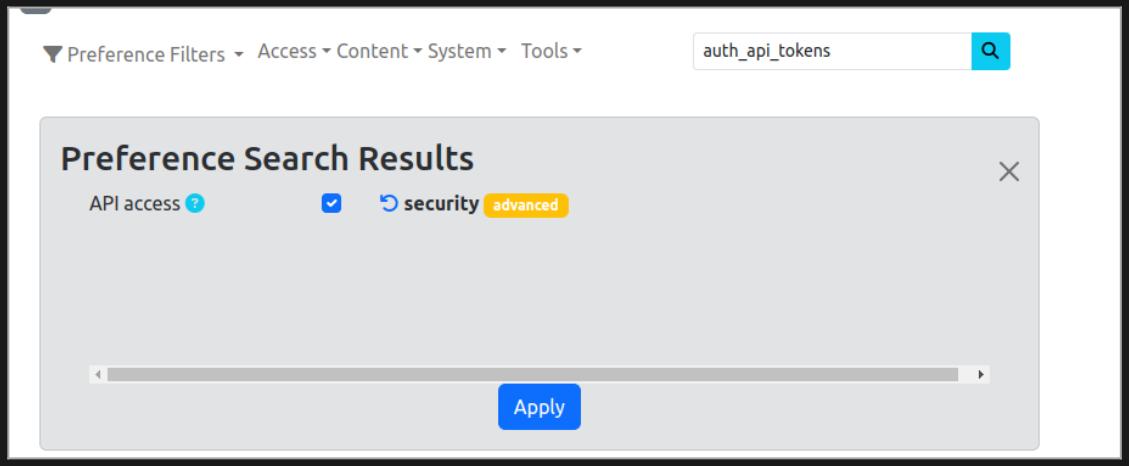
# GETTING STARTED

Enable the preference `auth_api_tokens` via the admin page.

## Enable API access

In the Control Panel, the screenshot below shows how to enable `auth_api_tokens` labeled as API access:

- Check the box "API access"
- Click "Apply" to save your choice.




Enable API access

# DOCUMENTATION

The Documentation is embedded in Tiki. See `/api/` on your target Tiki 24+ installation for an OpenAPI 3.0 documentation of the AP

## Documentation page

For example, let's assume that your Tiki 24+ instance is installed on `https://example.org`, then the page under

`https://example.org/api/` should look alike:

API Documentation page

## EXAMPLE: LET'S `GET` THE API VERSION

### GET/version Request

Assuming `example.org` has API feature enabled, this code snippet,

```
$ curl --request GET 'http://example.org/api/version'
```

Should return the following output as reponse:

{ "version": "24.1vcs" }

See GET/version reference in documentation.

## AUTHORIZATION

API requests should be authenticated with a token created by Tiki admin (via Admin -> Security tab). Each token gives their owner access with one and only one Tiki user. Permissions configuration is then based on that Tiki user's groups.

### Authorization token

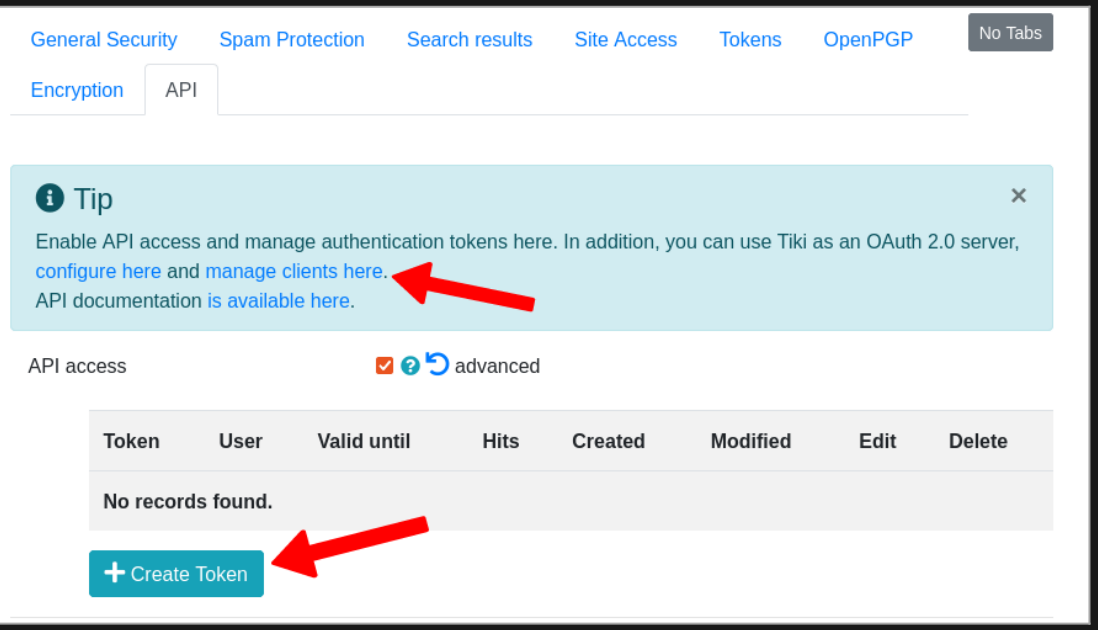
Bearer token authorization header in format:

Authorization: Bearer TOKEN

TOKENS CAN BE CREATED IN TWO WAYS:

1. Using Tiki OAuth 2.0 server. The documentation contains endpoints and parameters for different grant types.

2. Manually, in the Control Panel via Admin -> Security tab. Each token is associated with a user. Any API call using the token will act as the user observing all user's permissions.




Using Tiki as OAuth 2.0 server or Create a user token

## OAUTH 2.0 SERVER

OAuth 2 provides authorization flows for third-party applications.

### Tiki can act as an OAuth server.

In the Control Panel, Admin -> Security tab has a link to manage authenticated clients. This section creates client IDs and secrets for web, desktop or mobile applications using Tiki API.

Authorization flow can be:

1. Machine-to-machine - use client authorization grant type. Send your credentials directly to access_token endpoint to retrieve the access token.
2. End-user-to-machine - use auth flow grant type. Start by sending the user to authorize endpoint. This allows Tiki to ask target user for permission to grant access token with their user privileges. Once agreed, user is redirected back to your app/web app/machine where you do a machine-to-machine request to access_token endpoint to get the actual access token.

Access tokens generated by Tiki OAuth server are JWT encoded.

## TIKI RESTFUL API COVERAGE

CRUD operations(Create, Read, Update and Delete) are available for Category, Comments, Groups, Trackers/Fields/Items, Users an
Wiki pages.
The endpoints include:

1. Authorization flow.
2. API version.
3. Category: Object categorization and and CRUD.
4. Comments: Thread locking, moderation and CRUD.
5. Groups: User association and CRUD.
6. Search index rebuild and lookup.
7. Trackers/Fields/Items: Special features like dump/export, clone, duplicate, clear and CRUD.
8. Manage object translations.

9. User registration and CRUD operations, messaging and emailing wiki pages.
10. Wiki pages: Locking and parsing/display and CRUD.

Major items in wishlist for next versions of the API:

1. Files and file galleries (added in Tiki27)
2. Articles, blogs, other wiki-related elements.
3. Calendars.

See all the references in the documentation.
Example Tracker API usage with JavaScript here https://dev.tiki.org/API-Access-Example.

## PRE-TIKI 24 NOTES

[+]

## ALIASES

- service URL
- URL arguments