Skip to main content

AITable API Introduction

The API (Application Programming Interface, Application Interface) is a communication interface created to relay data between software.

AITable official provides various API interfaces that allow users to fetch data from AITable in the form of HTTPS requests, or push data to AITable.

What can you do with AITable's API

Connect AITable to other software, achieve mutual data transfer and build efficient workflows

- Connect to robot: to bring to groups when datasheet change
- Connect the iOS App "Shortcuts": to quickly write a record from your phone to AITable
- Connect browser plugin：implements bulk upload of images from Excel to AITable
- ......

As a backend database, help build product prototype quickly and validate ideas

AITable can easily store up to 10,000 lines of data that you can view as a lightweight NoSQL database without a SQL command, easily read and write data using the AITable's API.

- Developing a Zelda Menu that can easily read menu data from AITable
- Developing a lightweight course forum website to easily read discussion data from AITable
- Developing a simple note app to store notes data easily into AITable

- ……

Open APIs

Currently, AITable supports 7 types of API interfaces: records, fields, views, attachments, spaces, work directories, and contacts, please check the API Reference for more details.

Common Parameters

The base URL of the AITable API request is `https://api.AITable.com/fusion/v1/`.

> Note that：must use https, and cannot use an HTML request.

Some of the commonly used parameters when calling an interface are listed below and how to get the corresponding parameter value.

spaceId

A single user may create or be invited to multiple spaces and each space has a Space ID (i.e. spaceId).

You can get spaceId by either way：

▶ 1. Sign in to AITable, click on the personal avatar in the lower left corner and copy the Space ID.

▶ 2. Sign in to AITable, enter the space overview, copy the Space ID.

1. Call to get the Spaces list API interface to get the required spaceId.

nodeId

Each space has a work directory that has many file nodes.

Each node has an ID (i.e. nodeId).File node type includes:File node type includes:

- Datasheet: its nodeId equivalent to datasheetId is a string beginning with `dst` , e.g. `dstZsEg3RpBvsdCgoop.`
- Folder：its nodeId is a string starting with `fod` , e.g. `fod23ha5NvyM5`.
- Form：its nodeId is a string starting with `Fom` , e.g. `fom68eghkCem0wZxk`.
- Dashboard：its nodeId is a string starting with `dsb` such as `dsbWxTei5gdTvdAfKM`.

The way to get nodeId is simple：

Open any file node (datasheet, folder, form or dashboard) and find a string starting with `dst`, `fod`, `fom` or `dsb` this is the nodeId of the file.

datasheetId

Each datasheet has a corresponding AITableID (i. e. datasheetId).Each datasheet has a corresponding AITableID (i.e. datasheetId).A dataheetId must be specified when calling the API to add and delete datasheet.

Simple way to get dataheetId:

Open any datasheet and find a string starting with `dst` in the URL address bar. This is the datasheetId of this datasheet.

dashboardId

Each dashboard has a corresponding dashboard ID (that is, dashboardId).

The way to get dashboardId is very simple:

Open any dashboard and find a string starting with `dsb` in the URL address bar. This is the dashboardId of this dashboard.

formId

Each form has a corresponding form ID (that is, formId).

The way to get formId is very simple:

Open any form and find a string starting with `fom` in the URL address bar. This is the formId of this form.

viewId

Multiple views may be created in a datasheet.Each view has a view ID (i. e. viewId).Each view has a view ID (i. e. viewId).

The way to get viewId is simple：

Open any datasheet and find a string starting with `viw` in the URL address bar. This is the viewId of this datasheet.

recordId

A wiki table consists of several lines of records, each of which has a corresponding record ID (i. e. recordId).

You can get recordId by either way：

▸ 1. Open any datasheet, expand a line of record to fetch recordId and find a string starting with "rec" in the URL address bar. This is the record Id for this line.

2. You can call Get Record API interface to get the required recordId.

fieldId

A datasheet consists of several columns, each of which corresponds to one field.A datasheet consists of several columns, each of which corresponds to one field.Each field has a field ID (i. e. fieldId).

You can get fieldId by either way：

▸ 1. Open any datasheet to confirm that the current view contains fields that want to fetch the fieldId and then open the API example panel by clicking the "API" in the upper right corner. All fields of the current view can be found under "Fields".

2. You can call Get Field API interface to get the required fieldId.

unitId

If we regard a space as a company in the real world, then teams can be seen as departments within the company. Therefore, you can create multiple teams within the space.

The Contacts of a space are composed of several teams and members. (Learn more about Team management)

- Each team can have multiple members.
- Members can also belong to multiple teams.

Roles, on the other hand, are not part of the Contacts. Each role can be associated with multiple members and teams. (Learn more about Role)

There is a unique ID called "unitId" for each member, team, and role.

You can use this "unitId" to get, update, or delete the specified member, team, or role.

For more details, please refer to the API documentation.