# Project Release Notes

Release 4.4 (April 2022)

# What's new

## Geo logging

Track and store app users' locations. Share their location information with other services, such as sending live location notifications or maintaining an audit trail of sites visited by the app users during work hours. For example, cab aggregators can track their cab drivers' location during work hours and use the location log for mandatory audits and payments.

Read more...

---

## Authentication schemes

Create reusable authentication schemes for use across several settings such as remote objects, geo logging, and custom buttons. Use the new 'Authentication scheme' menu on the left navigation bar to access the settings.

Read more...

## Remote objects

Define rich Flex objects using REST APIs to access records in external or remote objects. You can seamlessly use remote objects in all the app settings where a Flex

object can be used. The apps store, access, and update records directly on the remote source as though they were stored on Flex.

For example, you can define a remote object in Flex for accessing customer information from an existing CRM system in your app for enrollment and payment workflow.

Read more...

---

# Enhancements

## Inline-editing in list actions

Allow your app users on the web client to create, edit, and delete records directly in the list. You can also hide duplicate records in the list. Use the record settings in the list designer to enable this feature.

Read more...

## Multi-select reference fields

Allow app users on the web client to search and simultaneously add one or more child records by selecting multiple reference values from the search result. For example, in an asset management flow, they can search for Dell laptops in their inventory, and add one or more models to their assets list.

Read more...

---

# Export records from list actions

Enable your web client users to quickly export records from a list action to an MS Excel file directly. For instance, field executives can export a list of work orders they closed during the week, month, or quarter. They can then use this data to generate charts and reports in MS Excel.

[Read more...](#)

# Geo logging

The Geo logging feature allows you to track and record your users' movements.  You can use the location information to build apps that are capable of responding to location-specific needs in real-time.

For example, dispatchers can use the location tracking information to quickly track down and assign the field technician closest to the location of the caller. Similarly, cab aggregators can track their cab drivers' location during work hours and use the location log for mandatory audits and payments.

You can set up a schedule for activating location tracking. For example, you can set a schedule to automatically stop tracking your field workforce outside of working hours. You can optionally override the global tracking at the profile level and turn off geo logging for profiles such as the back-office workforce that don't need it to be tracked.

## Enabling Geo logging

Follow the steps below to enable geo logging globally:

1.

2. Expand the profile settings from the left navigation bar.

3. Click account settings. The settings page opens on the right.

4. Click the settings tab and scroll down to the location tracking setup section.

5. Enable the location tracking option.
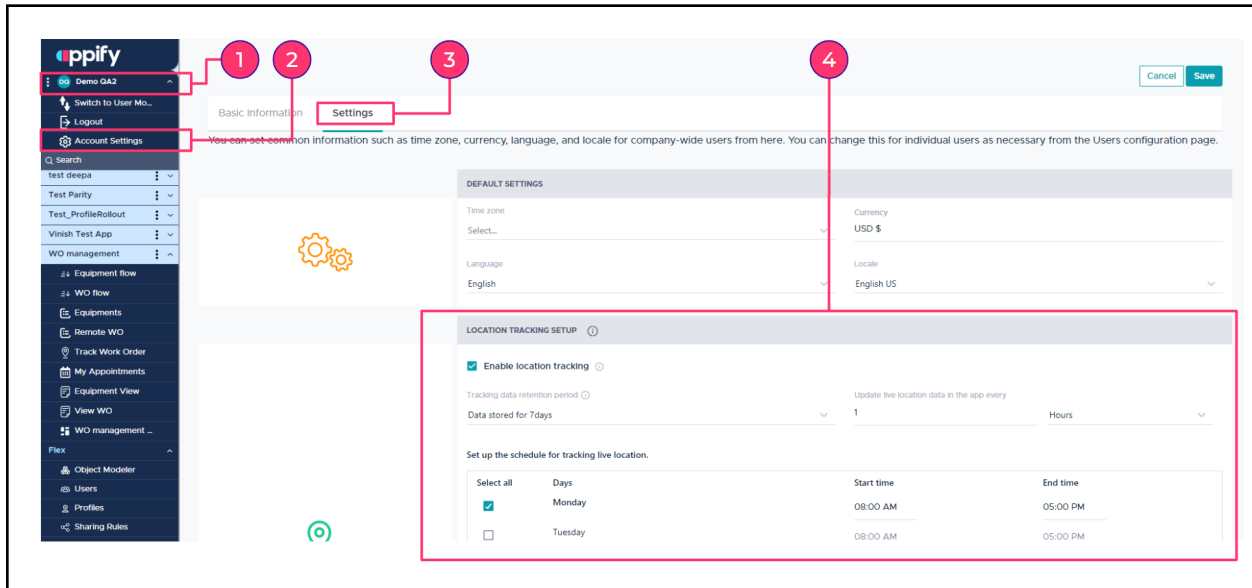
Enabling geo logging globally

| | |
|---|---|
| 1. Profile<br>2. Account settings | 3. Settings tab<br>4. Location tracking setup |

## Additional settings

You can also customize the settings to fine-tune the geo logging feature to your needs.

See the available settings below.

| Sl.No | Field | Description |
|---|---|---|
| 1 | Tracking data retention period | Set the duration for record retention. For example, you can set the retention period to one week to provide data for weekly audits. |

| | | |
|---|---|---|
| 2 | Live location update interval | You can define how often the location information of the app user is captured. The interval can be set in minutes and hours. |
| 3 | Set a schedule for location tracking | You can set a schedule to automatically activate the location tracking only during the specified timeslot on specific days. For example, activate tracking only from Monday to Friday. |
| 4 | API setup to share tracking data information on external services | You can send the geo logging information to REST API services such as billing, payroll, or audit services. |

| Additional settings | |
|---|---|
| 1. Option to enable location tracking | 4. Update interval |
| 2. Retention period | 5. API set for sharing logging data with other REST services |
| 3. Schedule for activating tracking | |

## Profile-level override

When you enable geo logging globally for all your users, you can override it at the profile level and disable it for users in that profile. For example, you can leave it enabled for users in the "Field force" profile but disable it for users in the "Managers" profile.

Follow these steps to disable geo logging for a profile:

1. Click the profiles menu on the left navigation pane. The Profiles page opens on the right.

2. Expand the profile for which

3. Click the profile for which you need to update the geo-tracking settings. The settings for the profiles appear on the right.

4. Use the Enable location tracking option to turn off geo-tracking for all profile users.



| Profile-level override | |
|---|---|
| 1. Profiles menu | 3. Profile details |
| 2. Profile | 4. Enable location tracking option |

## App user notes

The app user needs to allow location tracking on the device for geo logging to work.

# Authentication schemes

Create authentication schemes for use in settings such as remote objects, geo logging, and custom buttons. Use the new 'Authentication scheme' menu on the left navigation bar to access the settings.

Choose from two types of authentication methods - basic and custom. Set up username and password-based schemes using the basic authentication method. Use the custom method to set up authentication APIs for use in API calls.

Authentication schemes that you create here will automatically appear on all the screens across Studio where authentication settings are available.
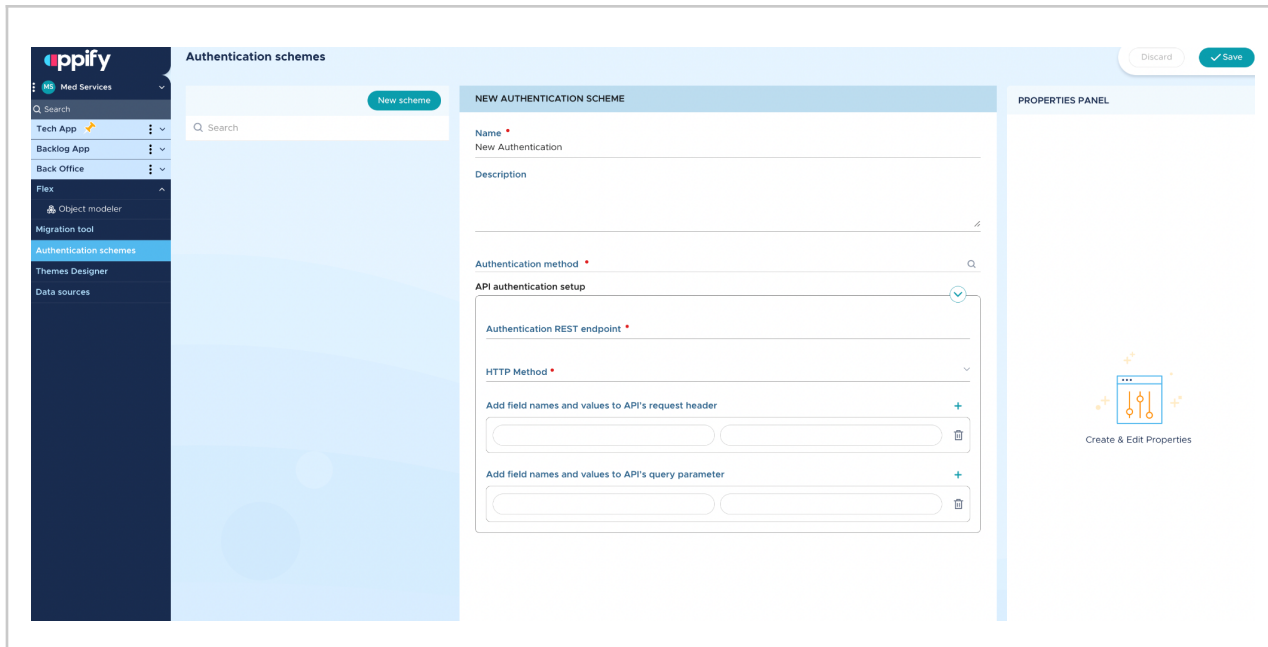
## Setting up custom authentication schemes

1. Authentication scheme menu

2. New scheme button

3. Search bar

4. Scheme name

5. Authentication method

6. Auth method picker

7. Auth endpoint

8. The HTTP method to use for the auth API

9. Custom request headers & query params

# Remote objects

Remote Object is a powerful new feature that enables you to easily connect your Appify apps with any data source that exposes REST APIs for accessing its records. A remote object is an object on Flex, mirroring the external object's data model - the structure and definition of the object and its fields. Data for remote objects are not stored in Flex. When an app connects to the remote object to access records, the remote object, in turn, makes an API call to the external object.

Remote objects provide numerous benefits some of which are listed below:

- You can use Appify Studio to define the data model based on your business process. It will enable you to quickly update and manage changes to your data model to meet your fast-growing business needs.

- You can seamlessly connect to any existing data sources, including legacy systems, to use the data from there directly in your apps. You don't have to duplicate the data locally or deal with the complexity of various data source connectors.

- Since the data will continue to reside on the existing system, any security and compliance systems you have set up for your data will continue to work as-is.

- Take advantage of all the features available to Flex objects. Features available to Flex are automatically applicable to remote objects. For instance, you can set up notifications and trigger actions for record creation and update. This is not possible in a connector-based integration with the external data source.

If you are already using external data sources that support REST APIs, you can now switch over to remote objects without changes to your app layer.

## Adding new remote objects

When you add a new remote object, only the object details such as object name, field names, and field types are stored on Flex. The actual data is stored on the remote object,

and Apps that use the remote object create the records directly on it. You will need to define REST APIs for creating, updating, retrieving, and viewing records from the remote object.

## Steps to create a remote object

1. Go to object modeler. The objects list appears on the right.

2. Click the Add new button. The object details tab appears on the right.

3. Enter the label name.

4. Click the "This is a remote object" option to turn it on.

| Object Modeler | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | Cancel | Save |
| Object Details | Fields | Dependencies | API settings | Uniqueness | Reference controls | Advanced search | Validation rules |

**OBJECT CREATE / EDIT**

Label
Remote Account

Api Name
Iodxa_remoteaccount

Description

Tags

Categories
Select...

☑ This is a remote object ⓘ

API alias
BROADCOM_CUSTOMER

API alias for accessing it as a child object
BROADCOM_CUSTOMER

### Adding a new remote object

| | |
|---|---|
| 1. Object modeler | 5. Option to make it a remote object |
| 2. Object details tab | 6. Alias for the remote object |
| 3. Fields tab | 7. Alias for referring to the object when it is a child object in an API call. |
| 4. API settings tab | |

### Adding fields to a remote object

Adding fields to a remote object is similar to that of Flex. For remote objects, there is an option to set an alias for the field. See the section on Aliases for more information. Every remote object should have a primary key defined. Enable the primary key switch for the field to make it so.
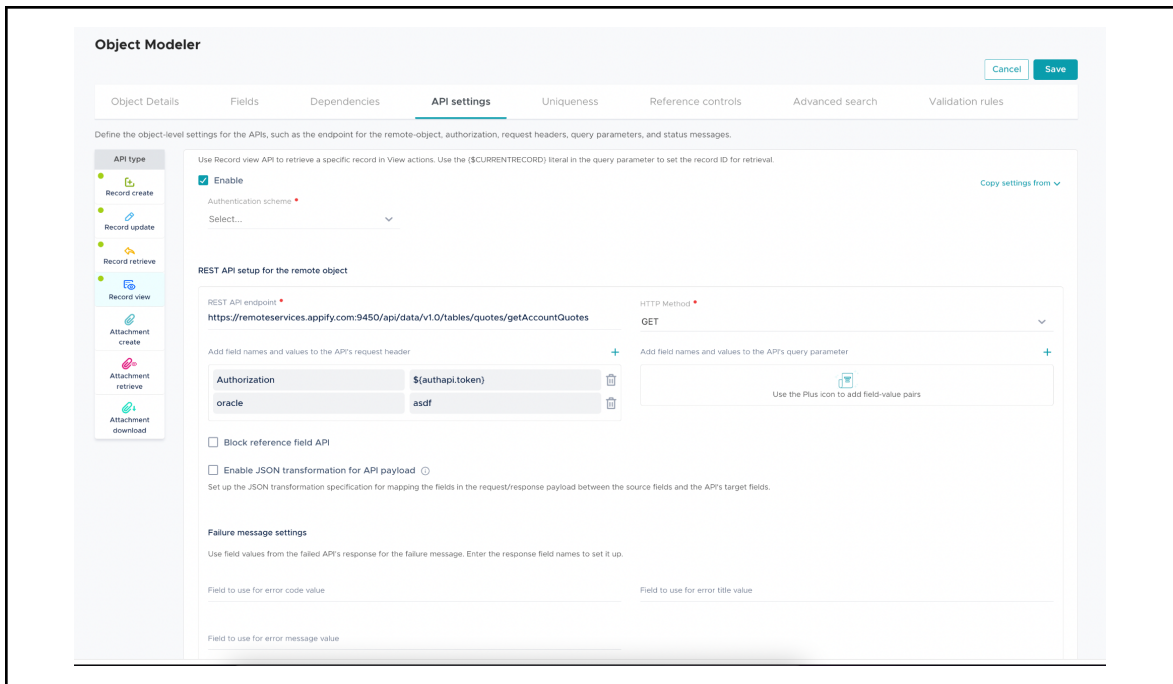
## Aliasing remote objects and fields

When the app makes an API call to the remote object, it refers to the object and fields by its Flex API name. However, the REST API service may not recognize the API name. To address such cases, set the object's name, as referred to by the API service, as an alias when you define it. The app will use the alias to refer to the object in the API call.

In an API call, a remote object can be accessed as a header object or a child object. To distinguish between the two, you can define another alias for accessing it as a child object in the "API alias for accessing it as a child object" field.

# Defining REST APIs for remote objects

You can define four types of APIs for the object. Refer to the table below for the API types and their descriptions.

| Record create | These APIs are meant for creating new records and are used in the create, fork, and checklist actions. |
|---|---|
| Record update | These APIs are meant for updating existing new records and are used in the edit and sign actions. |
| Record retrieve | These APIs are meant for retrieving a list of records and are used in the view, search, list, map, and calendar actions. They are also used in reference field lookup and in the initial setup for mobile devices. |
| Record view | These APIs are meant for retrieving a specific record and are used in the view actions. |

| REST API setup for remote objects | |
| --- | --- |
| 1. API settings tab | 6. Option to copy the API to other types |
| 2. API types menu | 7. Custom request header fields |
| 3. Authentication scheme picker | 8. Query parameters |
| 4. Endpoint | 9. JSON transformation spec |
| 5. Method to use for the API | 10. Failure message setup |

## Steps to add an API for a remote object:

1. Click the API settings tab. The API settings page appears below the tab.

2. Click the API type you need to define.

3. Choose an authentication scheme from the drop-down.

4. Enter the API's REST endpoint.

5. Choose an HTTP method.

You can customize the API call by setting request headers, query parameters, and the failure message. This is similar to setting up the API for custom buttons. A green dot on the API type menu indicates that the API is set up for that type.

> **Note:**
>
> When the JSON request or response body between the app and the API endpoint don't match, use JSON transformation spec to map the fields between the two. For help with setting it up, please reach out to your support contact at Appify.

## Literal supported in the API settings.

You can use the below literals in the API settings:

| | |
|---|---|
| ${USERINFO.*field_name*} | You can use it in the request header, query params, and the endpoint URL. |
| ${AUTHAPI.*response_field*} | In the request header, for sending auth-tokens. For example. You use the literal ${AUTHAPI.auth-token} to pass the authenticated token in the API call. |
| ${CURRENTRECORD.*field_name*} | Use it in the Endpoint URL for accessing the current record in View record API. |

## Object conversion

You can convert existing Flex objects to remote objects by enabling the 'This is a remote object' option on the object details tab. After the switch, the Flex object is deleted. Be sure to back up your records on Flex using the data export feature in data management.

Similarly, to convert a remote object to Flex, disable the 'This is a remote object' option. After the switch, the app users will not have access to their data on the remote

object. If you have the remote object's data in an MS Excel file, you use the import data feature from the management page to transfer the records to Flex. You will also need to update the user permissions from the profiles settings as required.

## Using remote objects for building apps

Referring to remote objects in the app actions is identical to using Flex objects. The remote objects appear in the object picker list when you choose Flex as the data source. You will see a new option, 'Use child object's APIs to access child records,' on the information tab for remote objects. Enable this if you need the app to make a separate API call using the child object's API for fetching child records.

| Using remote objects in apps |
| --- |
| 1. Option to allow the app to access child records using the child object API. |

# Enhancements

## Inline editing in list actions

Enable inline editing in list actions to allow users on Web client to quickly add, edit, and delete records directly in the list without opening the associated action. You can now also hide duplicate records in the list. For instance, supervisors can review and update the status of completed work orders directly on the work order list.

### Enabling in-line editing

Follow these steps to enable in-line editing in a list action:

1. Expand the app menu for your app.
2. Click the list action you need to update.
3. Enable the record editing, addition, and deletion options to allow the web client users to manage their records.
4. Enable the hide duplicate records option to show only unique records in the list to the users.
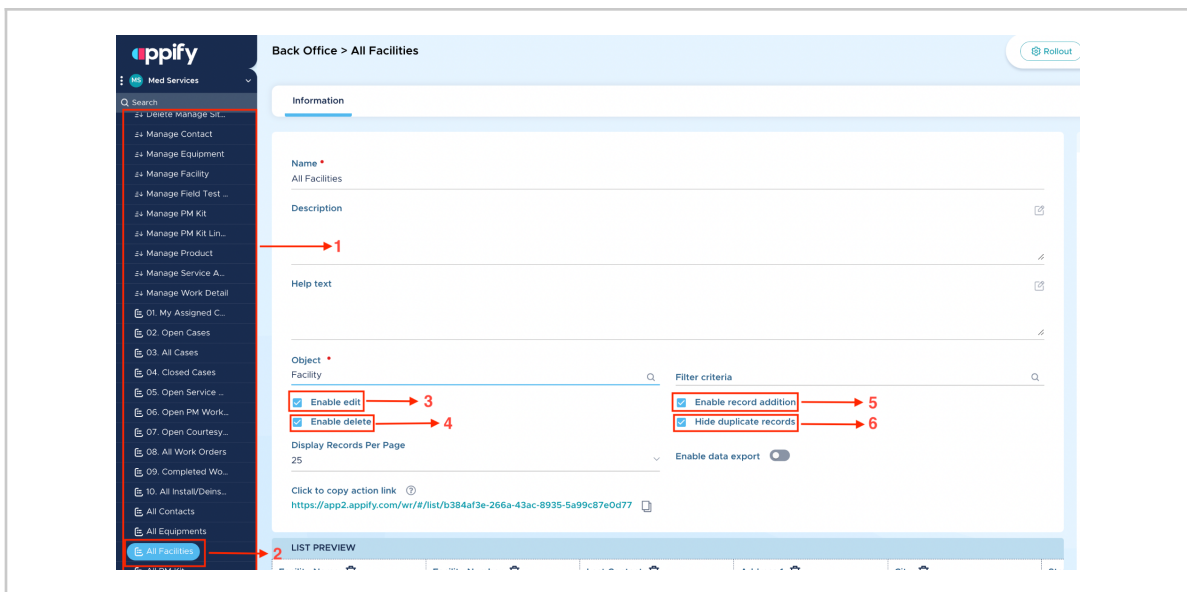
| Enabling inline editing | |
|---|---|
| 1. App menu | 4. Enable record deletion |
| 2. List action | 5. Enable record addition |
| 3. Enable record editing | 6. Hide duplicate record |

When you enable inline editing for lists, your web client users will be able to do the following on the list action page:

- Use the new 'Add record' button at the top of the list action page to insert an empty row for a new record.
- Use the bin icon on the row to delete the record.
- Click on any field in the list of records to directly edit it in place.

---

## Multi-select reference fields

Allow your web client users to quickly search for records from a child section's reference field, select multiple records at once from the search results, and add them directly to the child section. The reference values in the newly added child rows are populated with the selected values from the search result. The users can update the other fields as needed. This feature is available in all child and grandchild sections of the create, edit, and fork actions.

Enabling the multi-select option

Follow these steps to enable the multi-select option for reference fields in child sections:

1. Go to the child or grandchild section for the create, edit, or fork action
2. Click on the reference field in the section. The properties panel for the field appears on the right.
3. Enable the "Create multiple rows using this field" option in the field info section.

Notes for app users

When you enable the multi-selection option for the child section's reference fields, your web client users will see a checkbox in the reference fields lookup panel to select multiple matches and add them to the child section.

---

# Export records from lists

Your Web client users can now easily export a set of records from a list to an MS Excel worksheet. For instance, field executives can export a list of work orders they closed during the week, month, or quarter. They can then generate charts and reports using this data.

Enabling the export records option in list actions

You can enable the export option for a list action using the Enable data export option. You can customize the output file name for exported data using the User and Date literals in the filename.

Enabling export data option for list

| | |
|---|---|
| 1. Enable data export option | 1. Output filename |

## Notes for app users

When you enable the export data option for a list action, your web client users will see a new 'Export records' button at the top of the list page. Using column filters, users can export a subset of records from the list. For example, they can filter and export critical

priority work orders only. Clicking 'Export records' will save the records from the list in the MS Excel file format on users' local storage.



| Exporting a list on Web client | |
| --- | --- |
| 1. Export records button | 2. Filter option |

Exported file contents

The exported file consists of the summary and data worksheets. The Summary worksheet contains information about the file and its contents. The data worksheet contains the records exported from the list, with the field names in the header row.

| | |
|---|---|
|  |  |
| Summary worksheet | Data worksheet |
| 1. Summary tab<br>2. Export details | 3. Data tab<br>4. Fields in the header row |