

Complete Guide: QGIS to QField Workflow - Field Data Collection and Synchronization

Version-specific field workflow guide for **QGIS 3.44.11 LTR (Solothurn)**, QField, QFieldSync, and QFieldCloud — pinned as of **2026-06-25**.

Version Scope and Interface Note: This guide is intentionally written for **QGIS 3.44.11 LTR**. Exact icon placement and some plugin screen details may vary slightly by operating system or QFieldSync/QField version, but the documented menu paths are aligned to QGIS 3.44.11 and current QField/QFieldCloud documentation. Where a sub-panel name may vary slightly, this guide states so explicitly.

Table of Contents

1. Introduction
2. Part 1: Setting Up Your QGIS 3.44.11 Project
3. Part 2: Preparing the Project for QField
4. Part 3: Uploading to QFieldCloud
5. Part 4: Field Data Collection with QField
6. Part 5: Synchronizing Data Back to QGIS 3.44.11
7. Best Practices and Tips for QGIS 3.44.11
8. Appendices

1. Introduction

1.1 Overview of QGIS and QField Integration

QGIS is a mature open-source geographic information system used to create, edit, style, validate, analyze, and publish spatial data. QField is a mobile application built specifically to bring QGIS projects into the field on Android and iOS devices. Together, they provide a practical office-to-field workflow: the project is designed and quality-checked in QGIS, then deployed to QField for field collection, field verification, and mobile editing.

QFieldCloud extends this workflow by providing cloud-based project distribution and synchronization. A typical sequence is straightforward: prepare the project in QGIS, configure it for mobile use with QFieldSync, upload it to QFieldCloud, collect or update data in QField, and then synchronize those changes back to QGIS for review, reporting, and further GIS work.

This workflow is well suited to inspection programs, asset inventories, environmental surveys, utility mapping, land administration, community mapping, and any operational GIS project where structured field data must return to the desktop quickly and consistently.

1.2 Benefits of Field Data Collection with QField

- One project design can be reused from desktop to field.

- Offline collection is supported once the project is downloaded to the device.
- Field teams can capture points, lines, polygons, attributes, and attachments in one interface.
- Structured forms reduce typing and improve consistency.
- Defaults, constraints, and controlled lists improve data quality at the moment of capture.
- Synchronization through QFieldCloud reduces duplicate entry and manual file handling.
- QGIS 3.44.11 provides improved form configuration, symbology cleanup, labeling control, and database tools that are directly useful for field projects.

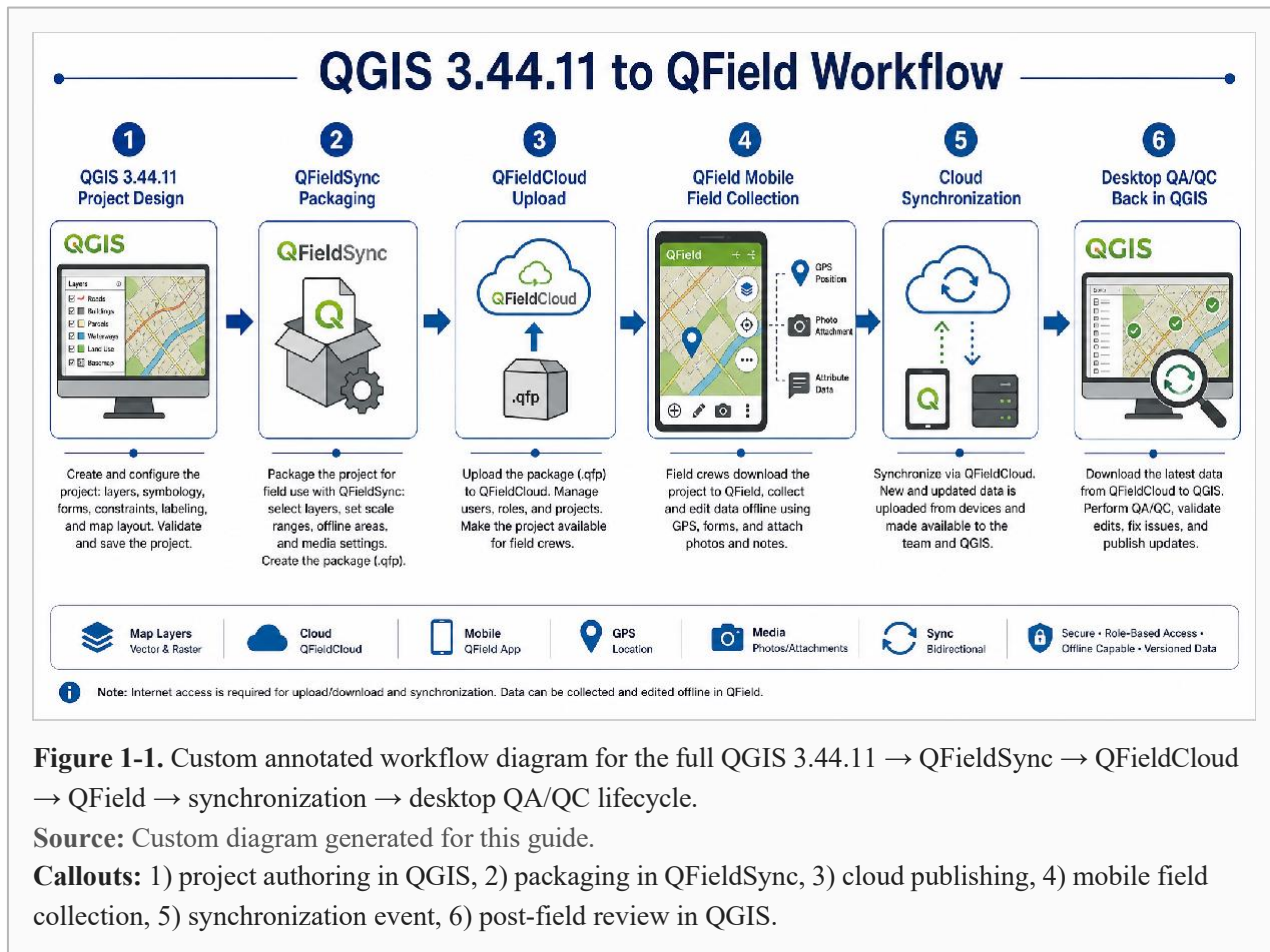


Figure 1-1. Custom annotated workflow diagram for the full QGIS 3.44.11 → QFieldSync → QFieldCloud → QField → synchronization → desktop QA/QC lifecycle.

Source: Custom diagram generated for this guide.

Callouts: 1) project authoring in QGIS, 2) packaging in QFieldSync, 3) cloud publishing, 4) mobile field collection, 5) synchronization event, 6) post-field review in QGIS.

1.3 Prerequisites and Requirements

Before building a field-ready project, confirm that the required software, accounts, data formats, and storage conventions are in place. The following checklist is tailored to QGIS 3.44.11.

Requirement	Recommended Specification	Purpose
QGIS Desktop	QGIS 3.44.11 LTR (Solothurn)	Project design, validation, packaging review, synchronization review
QFieldSync plugin	Current release compatible with QGIS 3.44.11; plugin repository shows recent v4.22.x builds compatible with QGIS 3.40.0–4.99.0	Prepare, package, and synchronize QGIS projects for QField and QFieldCloud
QFieldCloud account	Valid user account or organization access	Project hosting, sharing, and synchronization
QField mobile app	Installed on Android or iOS	Field collection, offline editing, and sync

Requirement	Recommended Specification	Purpose
Mobile device	GPS-enabled phone or tablet with adequate battery and storage	Field mapping, forms, GPS, and attachments
Project storage format	GeoPackage preferred for editable vector layers	Portable, robust offline editing and easier packaging
Project file format	.qgz preferred	Default QGIS project format; supports auxiliary data packaging
Internet access	Required for upload, download, sign-in, and synchronization; not required for actual offline field collection after download	Cloud connectivity
CRS planning	Appropriate projected CRS for local work, or carefully chosen alternative if required by workflow	Measurement accuracy, map consistency, and GPS interpretation

1.4 Version Scope and Compatibility

Component	Version / Scope	Compatibility Position Used in This Guide
QGIS Desktop	3.44.11 LTR (Solothurn)	This guide is pinned to QGIS 3.44.11 as of 2026-06-25
QFieldSync	Recent releases such as v4.22.x	Plugin repository lists compatibility with QGIS 3.40.0–4.99.0, so QGIS 3.44.11 is compatible
QFieldCloud authentication	Current documented workflow	Supports username/password and OpenID Connect when configured
QField project settings	Current documented workflow	Accessible in Project > Project Properties... > QField or via the QFieldSync settings icon
Project storage	QGIS 3.44 behavior	.qgz is the preferred default; projects can also be saved in PostgreSQL, GeoPackage, or Oracle
Form design behavior	QGIS 3.44 and current QField docs	Layer forms configured in Layer Properties > Attributes Form remain the basis of QField forms

Important Note: If you work in a mixed environment with multiple QGIS branches, keep the field project authoring environment standardized. This guide assumes that project preparation and synchronization review occur in **QGIS 3.44.11**, not in an older LTR or a QGIS 4.x build.

1.5 What Changed in QGIS 3.44.11 That Matters for Field Projects

QGIS 3.44 is the final feature release of the QGIS 3.x series, and several improvements are directly relevant to QField project preparation:

- **Attributes Form improvements:** the drag-and-drop form designer includes a search box and can switch between field names and aliases in the available-widgets and form-layout panels.
- **Merge Policies:** field-level merge behavior can now be configured in **Attributes Form > Policies**, which is useful for later office cleanup and controlled editing.
- **Symbology cleanup:** categorized renderers can delete unused categories, useful when preparing reduced field datasets.
- **Labeling control:** new settings for margin around labels and prevention of duplicate labels help reduce clutter on small screens.
- **Project scale calculation:** a project-level scale calculation method now exists and affects scale-dependent visibility and rendering logic.
- **Database/browser improvements:** better PostgreSQL project management, schema restriction, field rename, moving tables between schemas, and stored SQL queries make enterprise field projects easier to manage.
- **CRS functions:** the new `crs_from_text` and `crs_to_authid` expression functions help when building more advanced CRS-aware expressions.
- **GPS API exposure:** GPS tools are exposed to plugins and Python through `QgsAppGpsTools`, important for advanced integrations, even if most ordinary QField users will not use this directly.
- **Processing improvements:** enhancements such as native Fill Sinks Wang & Liu, raster calculator creation options, batch temporary outputs, Raster rank, and modeler refinements help with data preparation and post-field processing.

Tip: For most operational field projects, the most important practical gains in QGIS 3.44.11 are not flashy features but better form design, cleaner mobile symbology, stronger data organization, and more predictable project settings.

2. Part 1: Setting Up Your QGIS 3.44.11 Project

2.1 Install QGIS 3.44.11 and Required Plugins

1. Download and install **QGIS 3.44.11 LTR** from the official QGIS download page for your operating system.
2. Start QGIS and allow it to initialize its user profile and default settings.
3. Open **Plugins > Manage and Install Plugins**.
4. Search for and install:
 - **QFieldSync** — required for QField and QFieldCloud workflows.
 - **QuickMapServices** — optional, useful for convenient basemap access.
5. Restart QGIS if prompted after plugin installation.
6. Confirm that QFieldSync appears in the interface after restart.

 Representative web-sourced screenshot of the QGIS Plugin Manager showing QFieldSync

Figure 2-2. Representative web-sourced screenshot of the QGIS plugin manager with the QFieldSync listing visible.

Source: [GIS StackExchange](#).

Annotation focus: 1) plugin search field, 2) QFieldSync result entry, 3) install/update action, 4) enabled state after installation.

2.2 Create and Save a New Project

1. Select **Project > New**.
2. Immediately save the project into a dedicated working folder.
3. Prefer a structure such as:
 - **/project** or root project folder
 - **/data** for GeoPackage and local reference layers
 - **/attachments** or project subfolders used by attachment expressions
 - **/exports** for delivered outputs
4. Save the project as a **.qgz** file, which is the normal default in QGIS 3.44.11.
5. Use a clear name such as **asset_survey_2026.qgz**.

QGIS 3.44.11 supports saving projects to PostgreSQL, GeoPackage, or Oracle as well, but for typical QField workflows a local **.qgz** project file with portable data sources is the most straightforward starting point.

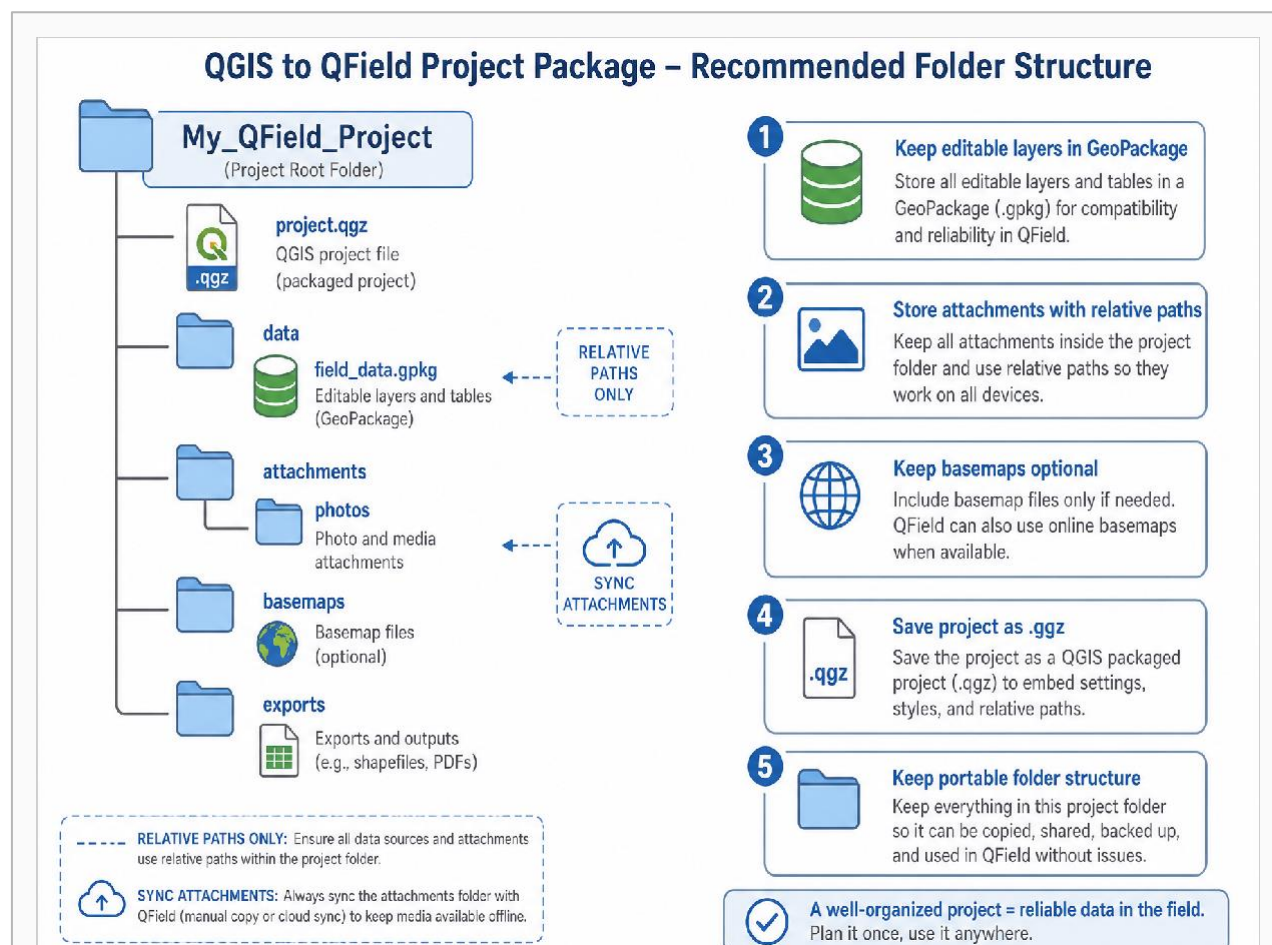


Figure 2-1. Custom best-practice folder structure for a portable QGIS/QField project package.

Source: Custom diagram generated for this guide.

Callouts: 1) save the project as a **.qgz** file, 2) keep editable layers in GeoPackage, 3) store attachments in project subfolders, 4) keep basemaps optional, 5) preserve relative paths for packaging and synchronization.

Warning: Do not build a QField project from unsaved memory layers. If a layer has not been written to disk, it may not package properly for offline use.

2.3 Configure Project Path and Storage Behavior

1. Open **Project > Project Properties...** and review relevant project settings.
2. Where practical, use **relative paths** for project portability.
3. Keep all field-editable layers and required attachments within the project folder structure.
4. If you maintain application-wide defaults, also review **Settings > Options > CRS and Transforms** and related options as needed.

2.4 Add Base Layers

Base layers help field staff orient themselves, but they should be chosen carefully. Some online services are useful for desktop planning but are not suitable for offline packaging.

1. Open the **Browser** panel.
2. Expand **XYZ Tiles**.
3. Drag an existing OpenStreetMap connection into the map canvas if available.
4. To add a new XYZ connection:
 - a. Right-click **XYZ Tiles** and choose **New Connection**.
 - b. Enter a clear name such as **OpenStreetMap**.
 - c. Paste a valid tile URL from your approved provider.
 - d. Save and test the layer.
5. Keep reference layers below editable layers in the layer order.
6. Only retain basemaps in the final mobile project if they are useful and supported by your packaging strategy.

 Representative web-sourced screenshot showing the QGIS Browser panel and XYZ Tiles workflow

Figure 2-3. Representative web-sourced screenshot of the Browser panel workflow used to add XYZ Tiles basemaps in QGIS.

Source: [Silver Spring Energy Consulting Ltd.](#)

Annotation focus: 1) Browser panel, 2) XYZ Tiles branch, 3) context menu or add-connection action, 4) basemap entry added to the project.

Important Note: Basemap licensing, offline caching rights, and mobile performance should be checked before deployment. A basemap that is convenient on the desktop may not be appropriate for a field device.

2.5 Create Editable Vector Layers

In QGIS 3.44.11, GeoPackage remains the recommended storage format for typical QField editing workflows because it is portable, supports multiple layers, and behaves better than ad hoc

collections of shapefiles.

Geometry Type	Typical Field Use Cases	Examples
Point	Discrete locations or observations	Hydrants, trees, poles, valves, sample points, inspection points
Line	Linear assets or routes	Pipelines, roads, fence lines, streams, walking traces
Polygon	Areas or boundaries	Parcels, habitat zones, damaged areas, land-use units, work zones

1. Select **Layer > Create Layer > New GeoPackage Layer**.
2. Create a GeoPackage such as **field_data.gpkg** in the project data folder.
3. Create your first editable layer, for example **survey_points**.
4. Define the geometry type and a proper primary key.
5. Repeat for line and polygon layers if your workflow requires them.
6. Add the new layers to the project and save the project again.

New GeoPackage Layer

Database

/path/to/new/geopackage.gpkg

...

Table name

Clinics

Geometry type

Point

☐ Include Z dimension
☐ Include M values

EPSG:4326 - WGS 84

New Field

Name

Capacity

Type

Whole Number (integer)

Maximum length

Add to Fields List

Fields List

Name	Type	Length
Name	text	80

Remove Field

Advanced Options

Help

Cancel

OK

Figure 2-4. Representative web-sourced screenshot of the New GeoPackage Layer dialog used to create editable field layers.

Source: [School of Data GIS Curriculum](#)

Annotation focus: 1) GeoPackage file path, 2) layer name, 3) geometry type, 4) primary key, 5) CRS selection before creation.

Tip: Be cautious with Shapefile and GeoJSON for editing-heavy mobile workflows. If you must use experimental data sources without a proper provider-side primary key, ensure a lowercase `fid` field is available for QField workflows that require unique feature identification.

2.6 Design Attribute Forms in Layer Properties > Attributes Form

Open the **Layer Properties** dialog by right-clicking the layer and selecting **Properties**, or by using the **Layer** menu when the layer is selected. In QGIS 3.44.11, the vector layer properties dialog includes tabs such as Source, Symbology, Labels, Fields, Attributes Form, Actions, Rendering, Temporal, Metadata, Digitizing, and others.

For mobile data collection, the **Attributes Form** tab is one of the most important parts of the project. Use the drag-and-drop designer to create a clean form layout for field users.

1. Open **Layer Properties > Attributes Form**.
2. Switch to the drag-and-drop designer if you need structured forms with groups or tabs.
3. Arrange fields into a logical order such as Identification, Observation, Condition, and Attachments.
4. Use aliases to present friendlier field labels on the mobile device.
5. Hide technical fields that field staff should not edit directly.
6. Use tabs if you intend to enable QField Wizard Mode later.

QGIS 3.44.11 introduces several improvements that are directly helpful here:

- A **search box** in the drag-and-drop designer helps filter large field lists.
- You can switch between **field names and field aliases** in the form designer panels.
- **Merge Policies** can be configured per field in the Policies section, improving later merge workflows.

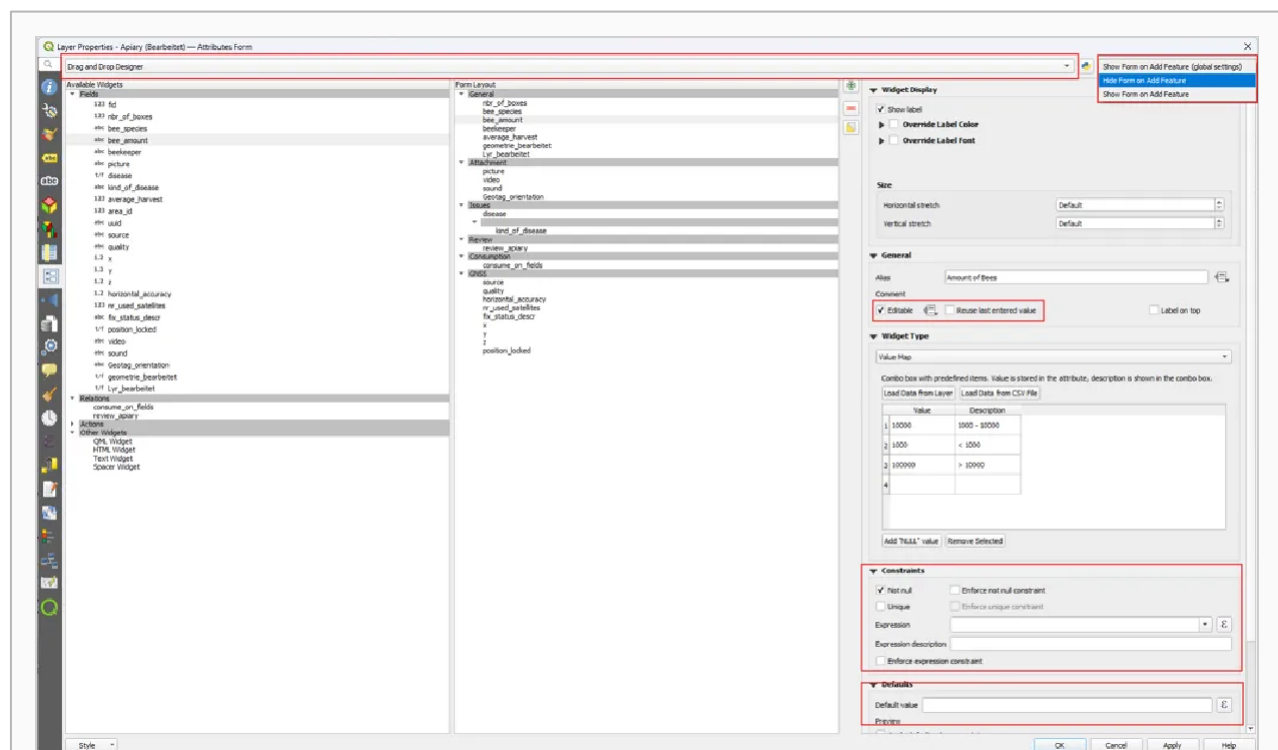


Figure 2-5. Official QField documentation screenshot of the QGIS attribute-form designer used to build mobile-friendly forms.

Source: [QField Ecosystem Documentation](#)

Annotation focus: 1) drag-and-drop form tree, 2) field list, 3) designer canvas, 4) widget properties, 5) tabs/groups needed for Wizard Mode.

2.7 Recommended Field and Widget Design

Field Purpose	Suggested Data Type	Suggested Widget	QField Support Note
Unique identifier	Text or integer	Text Edit or UUID Generator	UUID Generator is supported; avoid update-on-edit defaults for key fields
Status or category	Text	Value Map	Supported; may appear as combobox or radio-style options depending on configuration
Lookup from related table	Text or integer key	Value Relation or Relation Reference	Supported; use display expressions carefully
Yes/No field	Boolean or integer-coded logical	Checkbox	Supported
Measurement score	Integer or decimal	Range	Supported as editable spinbox and slider
Date of visit	Date or DateTime	Date / Time	Supported
Free notes	Text	Text Edit	Supported
Photo, audio, video, document	Text path field	Attachment	Supported; requires relative paths
Instructional content	Container / presentation	HTML, Text, QML, Spacer	Container-type widgets are supported

The current QField-documented supported widgets include: Attachment, Color, Date/Time, Checkbox, Hidden, Range, Relation Editor, Relation Reference, Text Edit, Value Map, Value Relation, UUID Generator, plus container widgets such as QML, HTML, Text, and Spacer.

2.8 Set Constraints, Defaults, and Editable Fields

1. In **Layer Properties > Attributes Form** and related field configuration areas, set required fields where appropriate.
2. Use expressions for **default values** such as current date or project/operator values.
3. Set fields as **editable** or read-only based on operational needs.
4. Configure **constraints** for mandatory values, valid ranges, and business rules.
5. Add user-friendly constraint descriptions so field staff understand what failed and why.
6. Use the option to remember last values only when it supports the workflow; otherwise, rely on deliberate defaults and controlled lists.

Tip: Good field forms minimize typing. If a field can be a default, lookup, checkbox, or constrained list, it usually should be.

2.9 Configure Symbology for Mobile Readability

1. Open **Layer Properties > Symbology**.
2. Use simple, high-contrast symbols that remain readable on a small screen.
3. For categorized layers, remove categories that are not needed in the field.
4. If using line layers, make widths readable without overpowering the map.
5. For polygons, use light fills and clear outlines.
6. Use scale-dependent visibility where necessary to reduce clutter.

In QGIS 3.44.11, categorized symbology includes an option to **delete unused categories**. This is particularly useful when a field project uses only a subset of a larger corporate classification scheme.

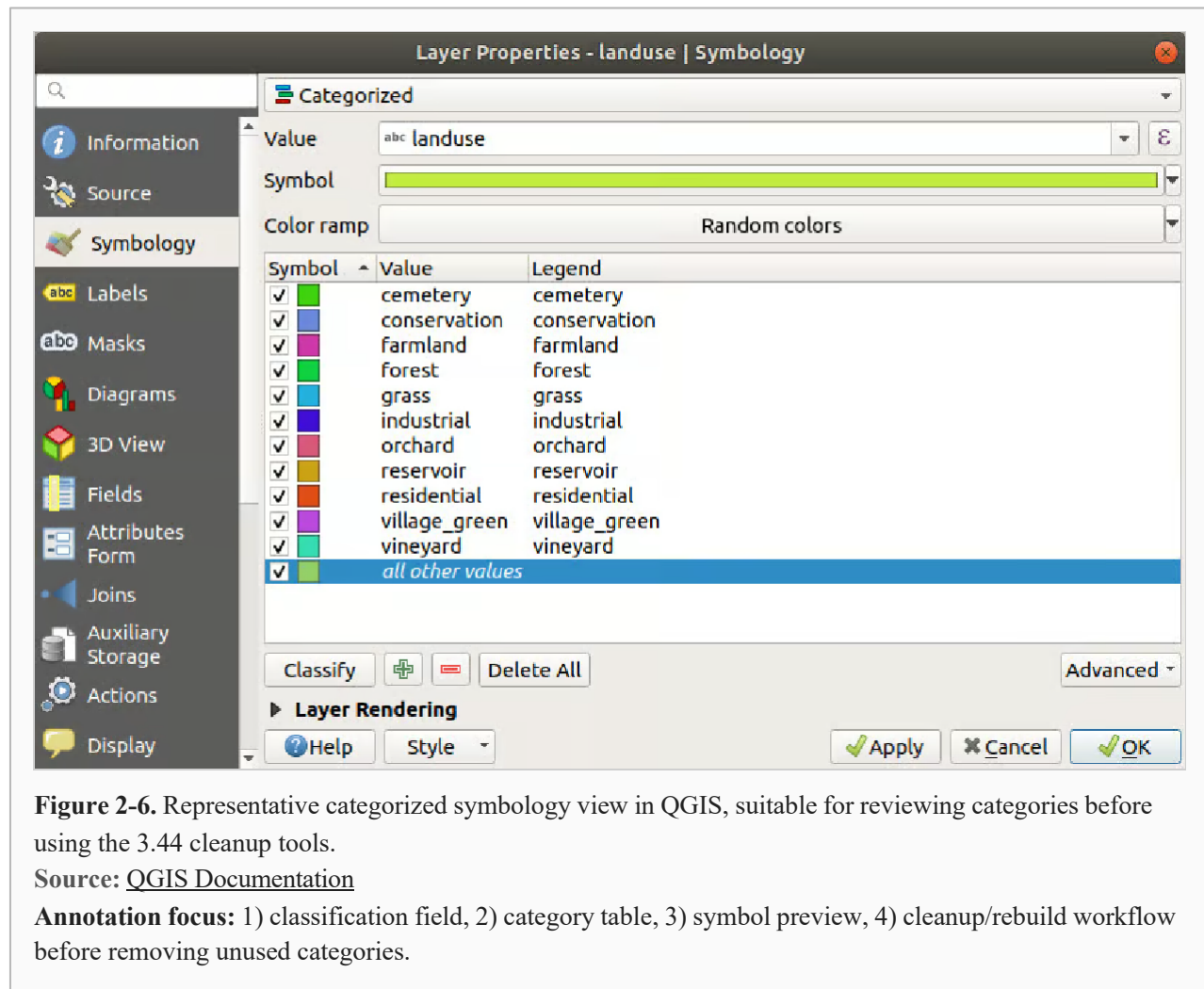


Figure 2-6. Representative categorized symbology view in QGIS, suitable for reviewing categories before using the 3.44 cleanup tools.

Source: [QGIS Documentation](#)

Annotation focus: 1) classification field, 2) category table, 3) symbol preview, 4) cleanup/rebuild workflow before removing unused categories.

2.10 Configure Labeling for Small Screens

1. Open **Layer Properties > Labels**.
2. Label only what a field worker actually needs.
3. Prefer concise labels and scale-dependent label visibility.
4. Test the label density at realistic mobile scales.

QGIS 3.44 introduces two especially relevant labeling controls:

- **Margin around labels**, which helps separate labels spatially.
- **Prevent duplicate labels**, which reduces repeated labels across layers.

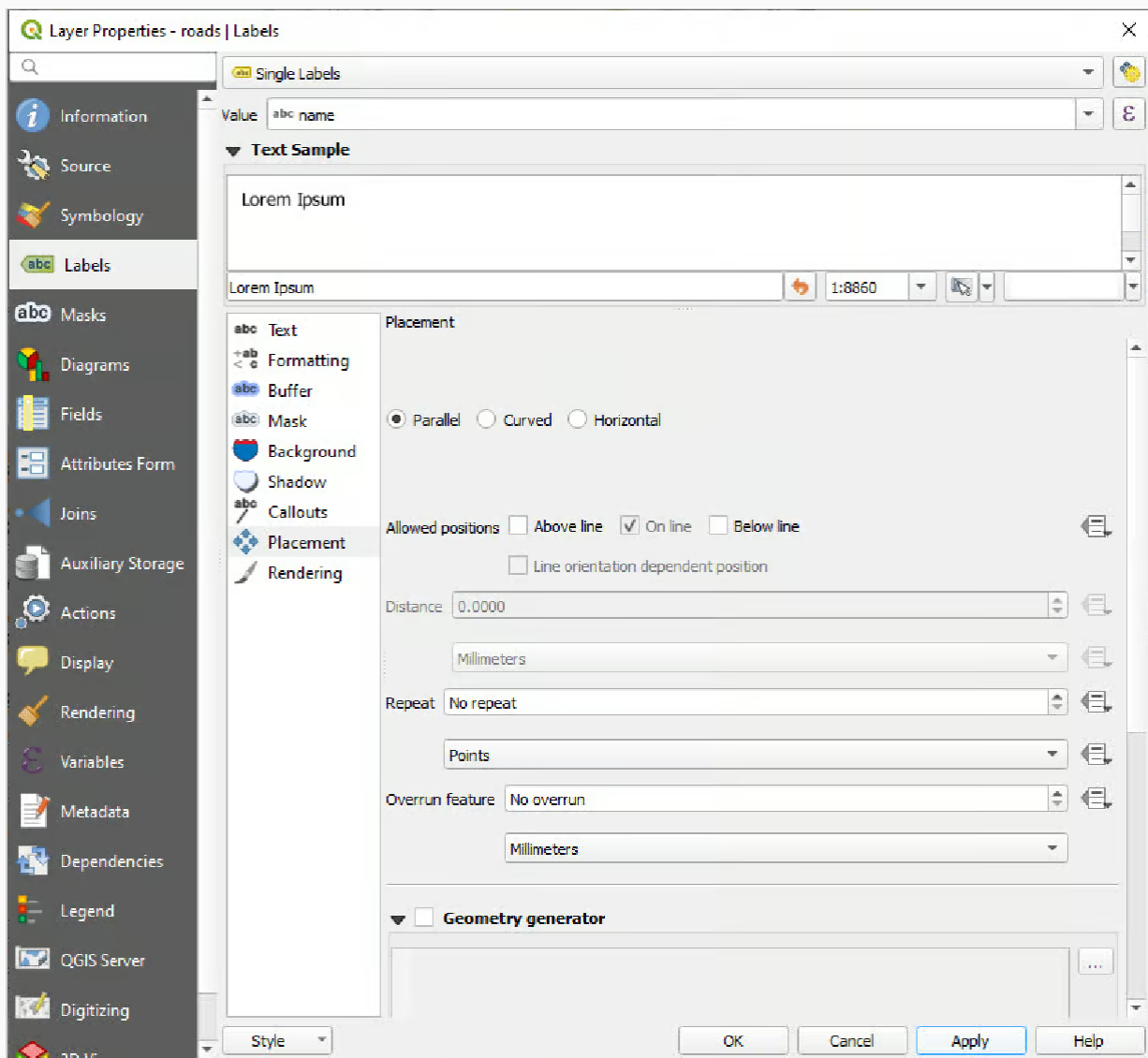


Figure 2-7. Representative QGIS labeling panel used to review label density and readability for mobile screens.

Source: [QGIS Documentation](#)

Annotation focus: 1) labeling enablement, 2) text formatting controls, 3) placement options, 4) collision/readability settings to reduce clutter in QField.

3. Part 2: Preparing the Project for QField

3.1 Confirm QFieldSync in QGIS 3.44.11

1. Open **Plugins > Manage and Install Plugins** if QFieldSync is not already installed.
2. Verify that the plugin is enabled in QGIS 3.44.11.
3. Confirm that QFieldSync tools and settings are available.

Recent QFieldSync releases such as v4.22.x are listed in the plugin repository as compatible with QGIS 3.40.0 to 4.99.0, so **QGIS 3.44.11 is within the supported range**.

3.2 Open QField Project Settings

1. Select **Project > Project Properties....**
2. Open the **QField** section.
3. Alternatively, use the QFieldSync settings icon if you prefer the plugin shortcut.

Current QField documentation identifies these as the main locations for QField-specific project settings. Depending on plugin version and layout, some labels may appear slightly differently, but the project-level QField settings are the correct place to start.

*Visual note: the current **Project Properties** > **QField** layout is shown in Figure 3-1 below; use that panel as the central location for packaging, attachment, and Wizard Mode settings.*

3.3 Review Packaging Actions and Offline Readiness

1. Review every layer and decide whether it should be editable, copied for reference, or excluded.
2. Ensure editable layers are stored in portable, offline-friendly sources such as GeoPackage.
3. Remove or replace data sources that are unsuitable for offline field editing.
4. Check that file paths are portable and relative where possible.
5. Verify that the project reopens correctly on the same machine from the project folder structure alone.

Important Note: QGIS 3.44.11 supports many source types, but a source being valid in desktop QGIS does not automatically mean it is appropriate for a mobile offline workflow. For field editing, keep the project conservative and portable.

3.4 Configure Mobile-Friendly Form Behavior

1. Reduce the form to essential fields for the field task.
2. Use group boxes or tabs for longer forms.
3. Hide non-essential or technical fields from field users.
4. Prefer Value Map and Value Relation widgets over free-text entry when appropriate.
5. Use display expressions in layers that will be searched through relation-based widgets.

3.5 Enable Wizard Mode Where Appropriate

QField supports a step-by-step **Wizard Mode** for feature forms. This is enabled in **Project > Project Properties... > QField**.

1. First organize your form into multiple root tabs using the drag-and-drop designer in QGIS.
2. Open **Project > Project Properties... > QField**.
3. Enable the setting for QField feature-form wizard mode.
4. Save the project and synchronize it again after enabling the setting.

Wizard Mode only works when the form has been organized into tabs. In the field, it changes the experience from a multi-tab desktop-like form into a step-by-step mobile workflow.

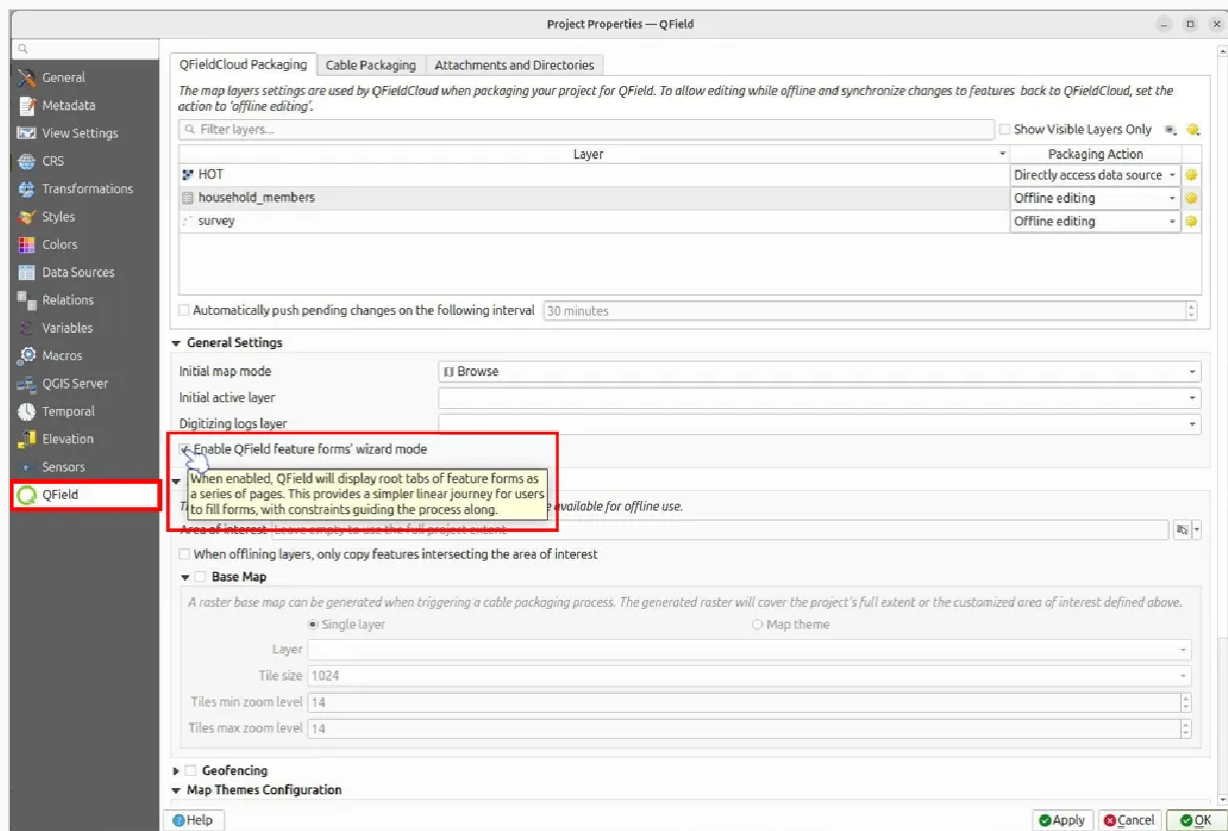


Figure 3-1. Official QField documentation screenshot of **Project Properties > QField** with the feature-form Wizard Mode option enabled.

Source: [QField Ecosystem Documentation](#)

Annotation focus: 1) QField project settings panel, 2) Wizard Mode checkbox, 3) save/apply workflow after changing the setting, 4) packaging behavior that depends on tabbed forms.

3.6 Configure Value Map and Value Relation Behavior

1. For short controlled vocabularies, use **Value Map**.
2. For maintained lookup tables, use **Value Relation** or **Relation Reference**.
3. Set clear display values so field users see meaningful labels rather than internal keys.
4. For large lookup tables, enable completer/search behavior where supported.
5. Group values if that improves usability, especially for species, defect types, or asset classes.

QField-specific layer properties may also allow you to influence when values appear as buttons versus a longer list. Exact labels can vary slightly by plugin version, so test the behavior on an actual device before deployment.

3.7 Configure Attachments Correctly

Attachment handling is one of the most common setup errors in field projects. QField requires **relative paths** for attachment workflows.

1. Create a text field intended to store the attachment path if you do not already have one.
2. Set its widget type to **Attachment**.
3. Configure the project so attachment paths are stored as relative paths.
4. Use QField/QFieldSync attachment settings if you need naming expressions or folder control.
5. Test photo capture, save the feature, and verify that the stored path is relative.

6. Verify that the attachment subdirectory is included in the synchronization workflow.

Current QField documentation states that attachments are stored in project subdirectories and that the attachment subdirectory must be included on synchronization.

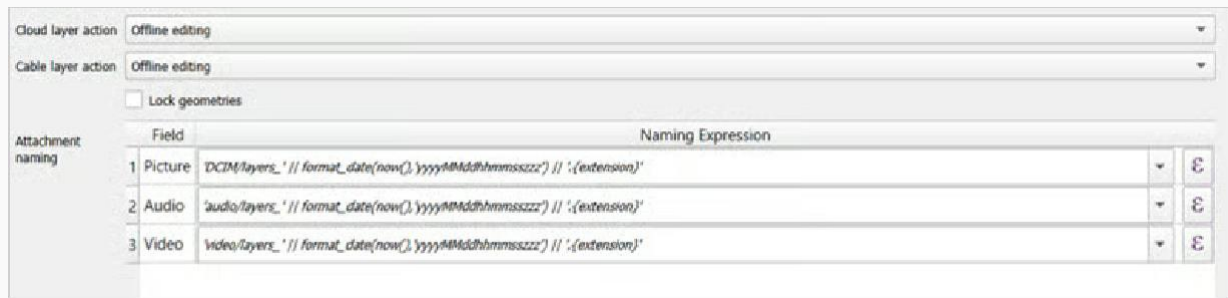


Figure 3-2. Official QField documentation screenshot of attachment path configuration for relative media storage.

Source: [QField Ecosystem Documentation](#)

Annotation focus: 1) attachment settings area, 2) folder/path expression, 3) relative-path behavior, 4) synchronization requirement for media subdirectories.

Warning: If attachments are stored with absolute paths, they may work on the authoring computer but fail on the mobile device or after synchronization to another machine.

3.8 Use Safe Expression Practices for QField

1. Use **layer names** rather than layer IDs in expressions intended for QField workflows.
2. Review default values, constraint expressions, and relation expressions for portability.
3. Test any QFieldCloud or positioning variables used in expressions before field deployment.

This is important because QFieldSync conversion can change layer IDs. Layer names are therefore the safer reference for QField-oriented expressions.

3.9 Configure CRS and Transformations in QGIS 3.44.11

1. Open **Project > Project Properties... > CRS** and confirm the project CRS.
2. Review **Project > Project Properties... > Transformations** if transformation control is important for your workflow.
3. Also review **Settings > Options > CRS and Transforms** for application defaults.
4. If you define custom CRS values, prefer **WKT**; PROJ strings are legacy and not recommended.
5. Use a projected CRS for most local measurement-heavy workflows.
6. Test coordinate behavior with the same layers you will send to the field.

CRS Choice	When It Fits	QGIS 3.44.11 Guidance
Local projected CRS	Engineering, cadastre, utilities, local environmental survey	Usually best for accurate distance and area work in field projects

CRS Choice	When It Fits	QGIS 3.44.11 Guidance
EPSG:4326 (WGS 84)	Simple global workflows, interoperability, some GNSS-centric workflows	Usable, but not usually preferred for precise local measurement work
EPSG:3857 (Web Mercator)	Display alignment with web basemaps	Acceptable for display convenience, not ideal for precise measurement
Custom CRS	Special institutional or engineering requirements	Define with WKT where needed; test carefully before deployment

QGIS 3.44 also introduces the expression functions `crs_from_text` and `crs_to_authid`, which may be useful in advanced expression-based logic.

3.10 Optimize Rendering for Mobile Performance

1. Remove layers that are not needed in the field.
2. Use scale-dependent visibility for dense reference layers.
3. Reduce excessive labeling.
4. Simplify symbology where possible.
5. Use QGIS vector simplification thoughtfully for heavy layers.
6. Test the project on an actual mobile device, not only on a desktop monitor.

In QGIS 3.44.11, application-level rendering options include vector simplification and other rendering controls that can materially improve field performance when used appropriately.

4. Part 3: Uploading to QFieldCloud

4.1 Create a QFieldCloud Account

1. Open the QFieldCloud web application.
2. Create an account with your email address and password, or use your organization's configured sign-in method if applicable.
3. Verify the account if required.
4. Review your dashboard and confirm whether you are working in a personal account or an organization context.

4.2 Understand Authentication Options

Current QFieldCloud documentation states that authentication can use:

- **Username and password**, or
- **OpenID Connect (OIDC)** when configured by the service or organization.

In QField and QFieldSync, a first attempt to store credentials in QGIS may trigger the **master authentication password** prompt. This is normal QGIS authentication system behavior and should be planned for during desktop setup.

Important Note: If you are deploying QGIS on shared machines, document who controls the QGIS authentication database and master password. This is especially important in team environments.

4.3 Log In from QFieldSync in QGIS 3.44.11

1. Open the QFieldSync interface or cloud connection tools in QGIS.
2. Choose the sign-in option for QFieldCloud.
3. Authenticate with username/password or OIDC if your environment supports it.
4. Complete any authentication prompts, including the QGIS master authentication password if asked.
5. Confirm that your project list or account context loads correctly.

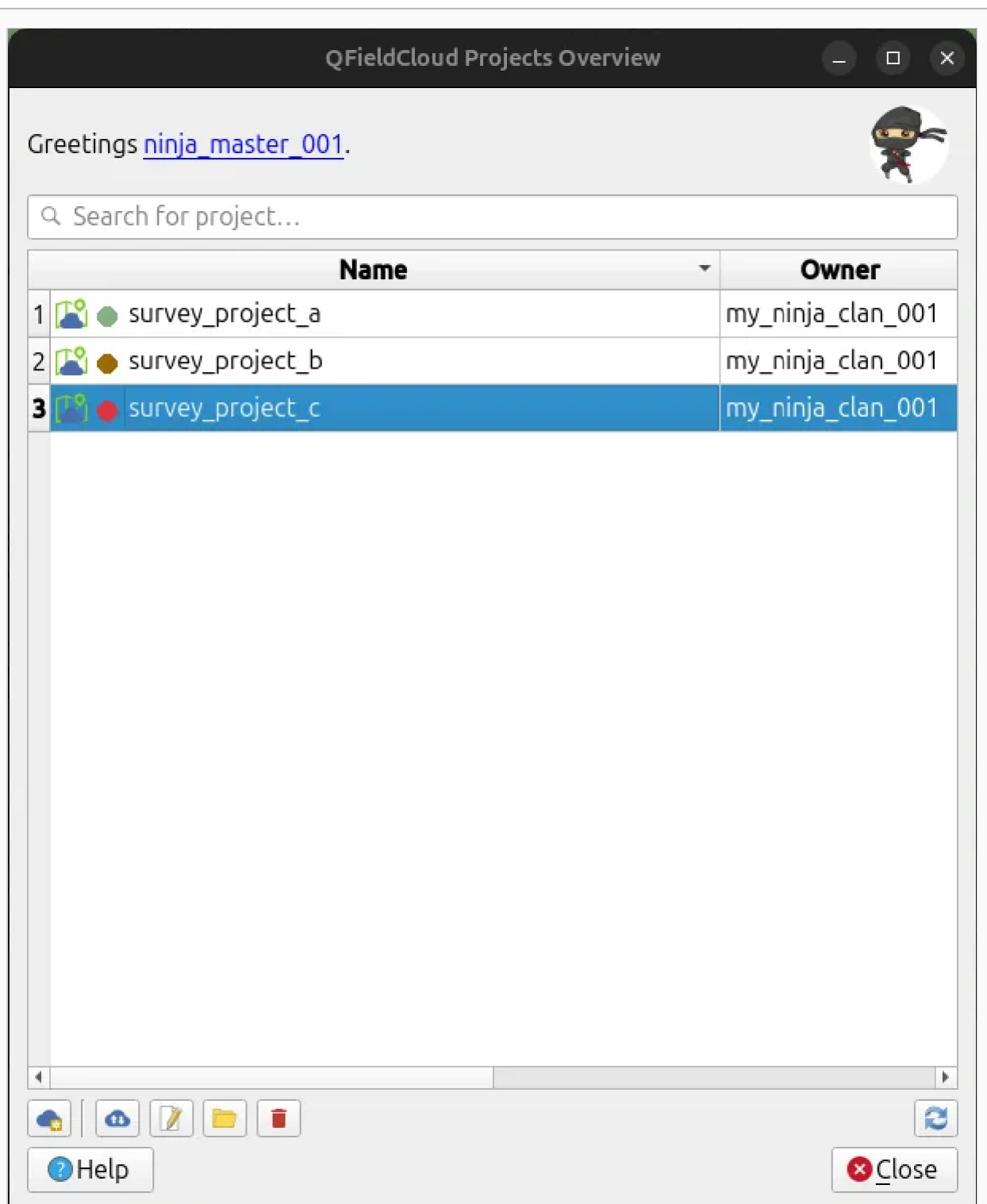


Figure 4-1. Official QFieldCloud documentation screenshot of the QFieldSync cloud workspace after sign-in.

Source: [QField Ecosystem Documentation](#)

Annotation focus: 1) signed-in user context, 2) project list, 3) create/refresh cloud controls, 4) workspace state confirming a successful connection.

4.4 Package and Upload the Project

1. Save the QGIS project.
2. Review QField settings, form design, attachments, and layer actions one last time.
3. Use QFieldSync to package or synchronize the project.
4. Choose whether you are creating a new cloud project or updating an existing one.
5. Wait for the upload to complete before closing QGIS.

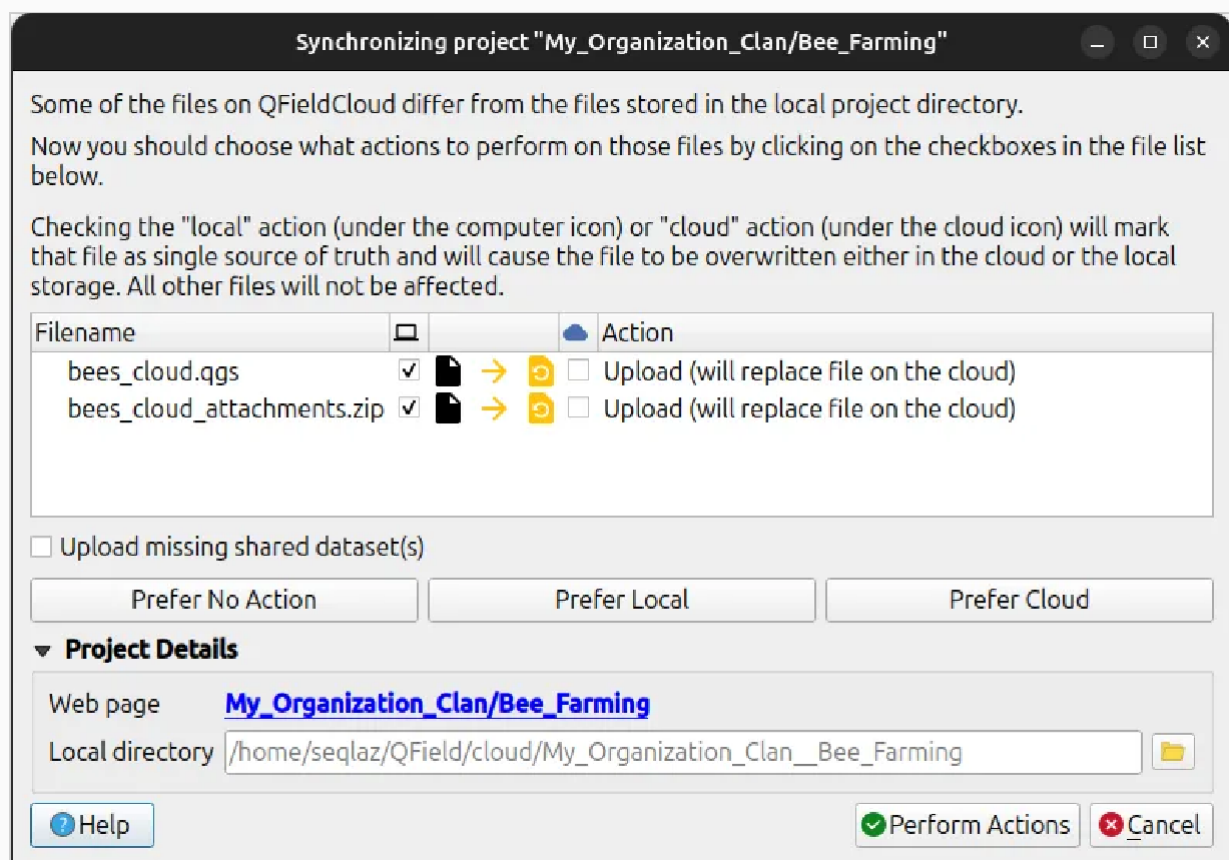


Figure 4-2. Official QFieldCloud documentation screenshot of the synchronize dialog used when publishing or refreshing a cloud project.

Source: [QField Ecosystem Documentation](#)

Annotation focus: 1) local-versus-cloud choice, 2) replace or amend behavior, 3) synchronization action button, 4) consequence of publishing the local package.

4.5 Understand Cloud and Local Sync Direction

One of the most important operational points is understanding whether you are pushing local changes to the cloud, pulling cloud changes to the local project, or performing a full synchronization. Use clear team procedures so project managers and field users do not accidentally overwrite expected changes.

- **Local to cloud:** publish or replace the cloud project with your current local project state.
- **Cloud to local:** refresh the local project from the cloud state.
- **Synchronization:** exchange changes according to the workflow and project state.

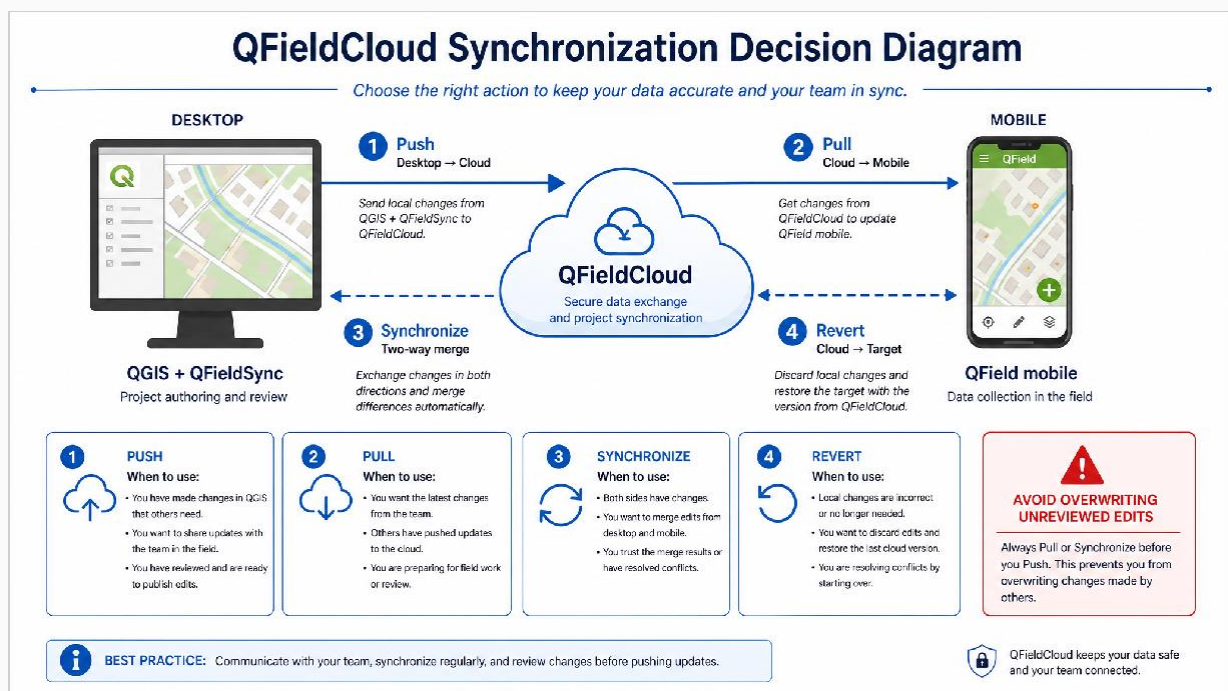


Figure 4-3. Custom synchronization decision diagram for QGIS, QFieldCloud, and QField mobile.

Source: Custom diagram generated for this guide.

Callouts: 1) **Push** sends local edits upward, 2) **Pull** refreshes the desktop or device from cloud state, 3) **Synchronize** exchanges changes in both directions, 4) **Revert** discards unneeded local edits.

4.6 Review the Cloud Dashboard

1. Open the project in the QFieldCloud web dashboard.
2. Confirm that the project appears under the correct owner or organization.
3. Review uploaded files and project status.
4. Check that the project is ready for field users before inviting them to download it.

4.7 Set Permissions and Sharing

1. Add users or team members to the project as needed.
2. Assign appropriate permissions such as administrator, editor, or viewer according to your governance model.
3. Confirm which users are expected to upload edits and which are read-only.
4. Document the synchronization schedule for the field team.

Tip: Use concise project names and descriptions in QFieldCloud so field staff can identify the correct project quickly on a mobile device.

5. Part 4: Field Data Collection with QField

5.1 Install QField on the Mobile Device

1. Install QField from the appropriate Android or iOS app store.
2. Open the app and grant permissions for location, camera, and file access as required.
3. Charge the device fully before deployment.

4. If external GNSS hardware is part of the workflow, pair and test it in advance.

5.2 Sign In to QFieldCloud

1. Open QField.
2. Select the cloud sign-in option.
3. Authenticate with username/password or your configured OIDC provider if available.
4. Wait for your project list to load.

Visual note: mobile sign-in varies slightly by platform and app version; the authenticated cloud context is represented in Figure 4-1 and the mobile collection interface is shown in Figure 5-1 below.

5.3 Download the Project

1. Select the required project from the project list.
2. Download it completely to the device.
3. Open the project and confirm that forms, layers, and map context appear correctly.
4. If the job is offline, perform this download while still connected.

Visual note: after sign-in, the project list exposes the download control for each cloud project. Verify owner, project name, and readiness state before downloading to the device.

5.4 Understand the QField Interface

Interface layouts vary by device size and app version, but field workers will typically interact with the map canvas, project/layer controls, GPS position tools, identify/edit tools, and feature forms.

- **Map canvas:** navigate, inspect, and digitize.
- **Layer/project controls:** turn layers on and off, access legends, and browse features.
- **GPS tools:** locate current position and support position-based capture.
- **Feature forms:** enter and validate attributes.
- **Sync controls:** review local changes and synchronize when connected.

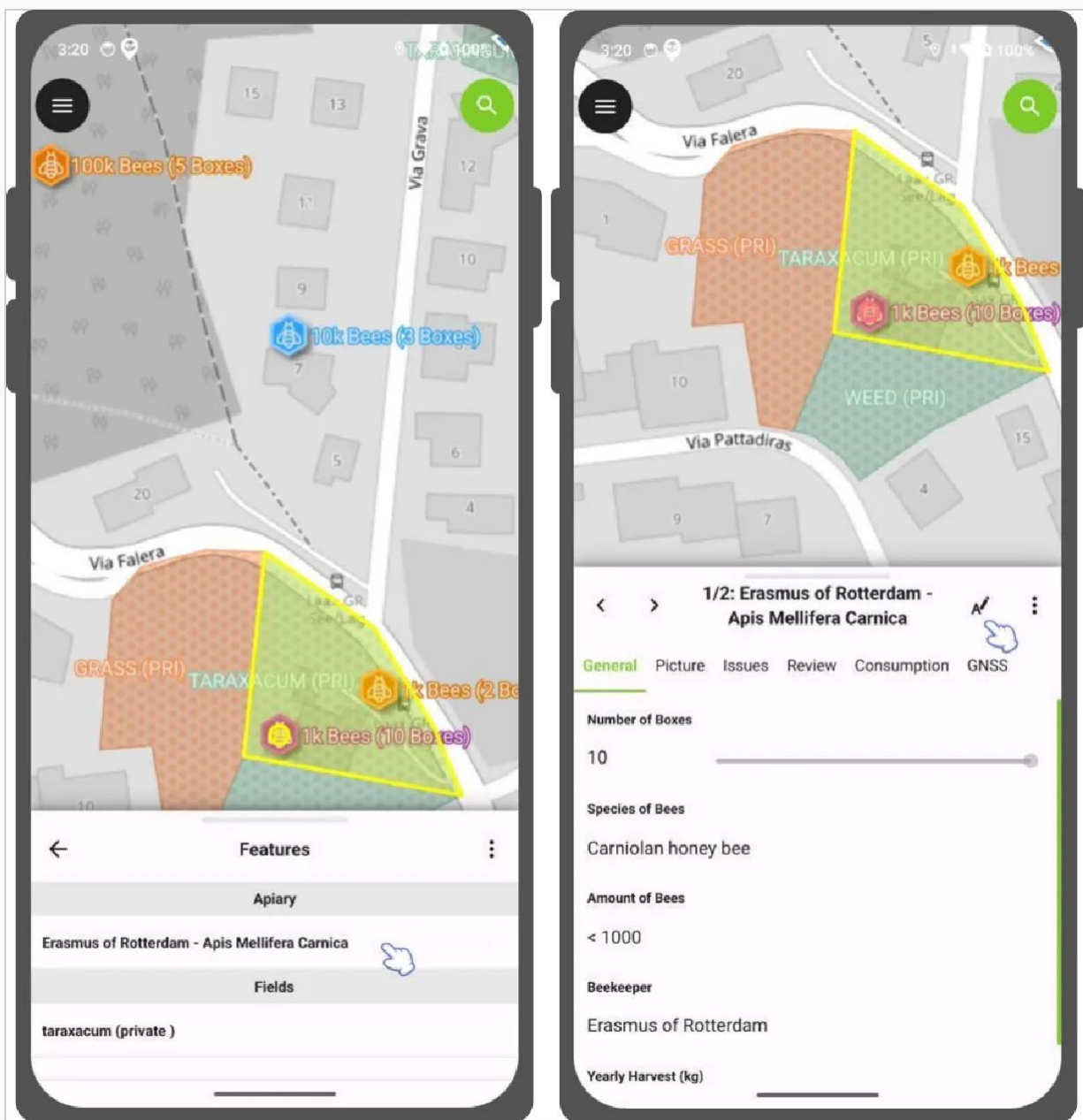


Figure 5-1. Official QField documentation screenshot showing a typical field interface in browse mode with identify and edit tools.

Source: [QField Ecosystem Documentation](#)

Annotation focus: 1) map canvas, 2) feature identify panel, 3) layer or tool controls, 4) entry points to editing and form review.

5.5 Capture GPS Points with Attributes

1. Move to the required location.
2. Allow the device GPS or connected GNSS to stabilize.
3. Select the target point layer.
4. Create a new point feature using the current position or map tap, according to project procedure.
5. Complete the attribute form.
6. Save the feature.

5.6 Draw Lines and Polygons

1. Select the line or polygon layer.
2. Start feature capture.

3. Add vertices in the required sequence.
4. Finish the geometry.
5. Complete the form and save.

5.7 Use Attachments and Photos

1. Open the feature form.
2. Find the Attachment field configured in QGIS.
3. Capture a photo, audio clip, video, sketch, or attach a supported document as required by the workflow.
4. Confirm that the item is stored and visible in the form.
5. Save the feature.

Because the project was designed with relative attachment paths, the link stored in the feature should remain portable between device, cloud, and desktop.

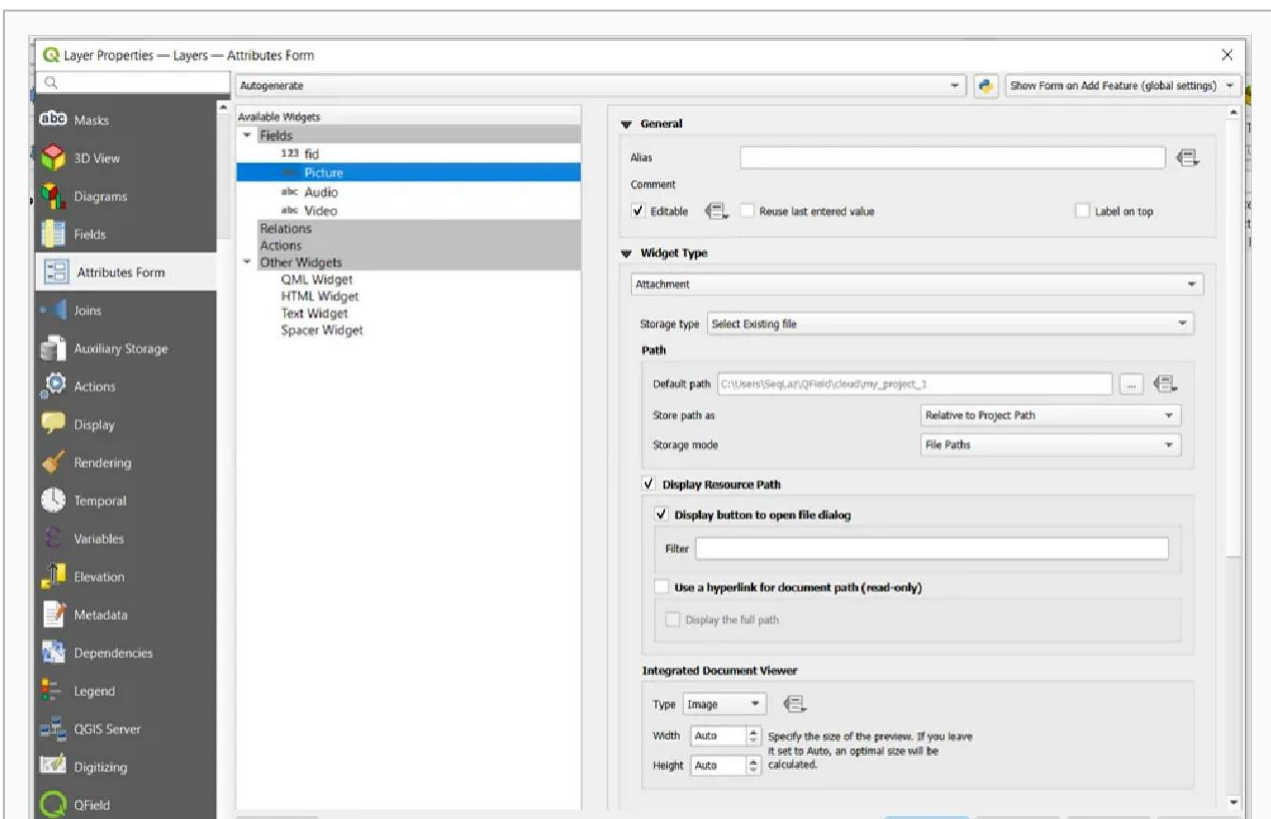


Figure 5-2. Official QField documentation screenshot of the attachment widget used for photo and media capture in the field.

Source: [QField Ecosystem Documentation](#)

Annotation focus: 1) attachment field, 2) media action buttons, 3) stored relative-path media item, 4) save-and-sync expectation for captured files.

5.8 Edit Existing Features and Respect Validation

1. Tap an existing feature or use identify/list tools to open it.
2. Edit attributes or geometry only if permitted by project rules.
3. Observe warnings or errors from constraints and correct any invalid values before saving.
4. Save the updated feature.

If Wizard Mode was enabled, the form may guide the user page by page. Hard constraints can block moving to the next page until invalid values are corrected.

5.9 Work Offline

1. Ensure the project is fully downloaded before leaving connectivity.
2. Collect features as normal while disconnected.
3. Do not manually move or rename project files on the device.
4. Reconnect only when ready to synchronize.

Tip: Before a live campaign, run one complete rehearsal: download the project, create and edit features offline, add a photo, and synchronize back successfully.

6. Part 5: Synchronizing Data Back to QGIS 3.44.11

6.1 Understand Mobile Sync Actions

Current QFieldCloud documentation describes three key mobile actions:

- **Revert Changes:** discard local unsynchronized edits on the device.
- **Push:** send local device changes to QFieldCloud.
- **Synchronize:** push local changes and refresh the device with the newest merged cloud project state.

Warning: Revert Changes is destructive for unsynchronized local edits. Field staff should be trained not to use it casually.

6.2 Upload Field Changes from QField

1. Reconnect the mobile device to the internet.
2. Open the project in QField.
3. Review the number of pending local changes.
4. Choose **Push** or **Synchronize** according to project procedure.
5. Wait until the operation completes successfully.

05:09 

  0.00 KB/s    80%



QFieldCloud

Greetings nlnja_mastw_001.



There is/are 1 local change(s)



Synchronize

Synchronize the whole project with all modified features and download the freshly updated project with all the applied changes from QFieldCloud.



Push changes

Save internet bandwidth by only pushing the local features and pictures to the cloud, without updating the whole project.



Revert local changes

Revert all modified features in the local layers. You cannot restore those changes.

Automatically push changes every 30 minute(s)



Last synchronized just now
No changes pushed yet



Figure 6-1. Official QFieldCloud documentation screenshot of the mobile sync actions menu.

Source: [QField Ecosystem Documentation](#)

Annotation focus: 1) pending-change indicator, 2) **Revert Changes**, 3) **Push**, 4) **Synchronize**, 5) choose the action that matches your team procedure.

6.3 Pull the Updated Project Back into QGIS 3.44.11

1. Open the corresponding project in **QGIS 3.44.11**.
2. Sign in to QFieldCloud from QFieldSync if necessary.
3. Use the appropriate synchronize or refresh action to bring cloud changes back to the desktop environment.
4. Wait until data and attachments are refreshed.
5. Save the project after confirming the updated state.

Visual note: desktop refresh after field upload follows the same QFieldSync cloud workflow shown in Figures 4-1 through 4-3. Always confirm that new features and attachments are present before saving the refreshed QGIS project.

6.4 Review Conflicts and Validate the Data

1. Check whether the same feature was edited in more than one place before synchronization.
2. Review any conflict indicators or operational inconsistencies.
3. Validate required fields, coded values, and geometry quality.
4. Confirm that all expected new features are present.
5. Document any manual corrections made after synchronization.

6.5 Review Attachments and Attributes

1. Open a sample of synchronized features in QGIS.
2. Verify that attribute values were preserved correctly.
3. Open photo or file attachments from the form where possible.
4. Confirm that relative attachment links remain valid inside the synchronized project structure.

6.6 Export Data for Delivery or Analysis

1. Right-click the layer and select **Export > Save Features As....**
2. Choose the output format, target CRS, and output location.
3. Export only after reviewing whether the synchronized layer set is final.

Export Format	Best Use	Recommendation in a QGIS 3.44.11 / QField Workflow
GeoPackage	Working storage, archiving, GIS delivery	Preferred for most ongoing GIS work and project handover
Shapefile	Legacy compatibility	Use only when required; field-name and structure limits remain significant
GeoJSON	Web sharing, lightweight interoperability	Useful for exchange, but not the preferred main editable storage for complex field projects
CSV	Reporting, non-GIS review, tabular handoff	Good for attributes and coordinate extracts, not for full GIS editing workflows

QGIS 3.44.11 has broad GDAL/OGR format support, but **GeoPackage** remains the preferred **working format** for most QField-authoring workflows.

7. Best Practices and Tips for QGIS 3.44.11

7.1 Core Recommendations for QGIS 3.44.11 Field Projects

- Prefer **GeoPackage** for editable vector layers.
- Use **relative paths** for project portability and especially for attachments.
- Use **layer names instead of layer IDs** in QField-oriented expressions.
- Use **.qgz** as the primary project file format unless you have a specific database-based reason not to.
- Prefer **WKT** rather than PROJ strings for custom CRS definitions.
- Use scale-dependent visibility and vector simplification for heavy layers.
- Keep attribute-table behavior efficient when working with large datasets.
- Test every field form on an actual mobile device before rollout.

Visual best-practice references: Reuse **Figure 2-1** for portable folder structure, **Figure 3-2** for attachment-path discipline, **Figure 4-3** for sync-action decisions, and **Figure 6-1** for the mobile push/synchronize menu before every deployment.

7.2 Battery and Field Hardware Practices

- Start with a full charge and carry backup power.
- Reduce screen brightness where safe to do so.
- Disable unnecessary radios when not needed.
- Protect the device from heat, moisture, and physical impact.
- Test GPS and camera behavior before leaving the office.

7.3 Backup and Data Protection Practices

- Keep a pre-deployment archive copy of the project.
- Create dated or versioned backup copies before major structural changes.

- Synchronize field edits as soon as reliable connectivity is available.
- Export milestone copies of critical layers during long campaigns.
- Document who is responsible for final synchronization review in QGIS 3.44.11.

7.4 Performance Practices

- Reduce unnecessary labels and decorative cartography.
- Keep form layouts focused and short.
- Use the categorized symbology cleanup tools in QGIS 3.44 where appropriate.
- Consider vector simplification for dense reference data.
- For large attribute tables, configure behavior so QGIS does not always open full tables unnecessarily.

7.5 Security Practices

- Use strong passwords or managed identity provider authentication where supported.
- Protect field devices with screen lock and device encryption where available.
- Remove project access promptly for users who no longer need it.
- Store sensitive location data and attachments according to organizational policy.
- Control who manages the QGIS authentication database and master password on shared desktops.

7.6 Troubleshooting Common Issues

Issue	Likely Cause	Recommended Action
Project opens in QGIS but not properly in QField	Unsupported or non-portable source, missing package element, or broken path	Move editable layers into GeoPackage, use relative paths, repackage with QFieldSync, and retest
Attachments do not appear after sync	Attachment path not relative or attachment subdirectory not included correctly	Reconfigure Attachment widget to use relative paths and verify attachment subfolders
Form values or expressions behave unexpectedly in QField	Expressions depend on layer IDs or unsuitable desktop assumptions	Use layer names instead of IDs and retest all defaults and constraints
Project is slow on the device	Too many layers, dense labels, heavy symbology, or oversized rasters	Simplify rendering, reduce labels, use scale-dependent visibility, and remove nonessential layers
Edits appear missing in QGIS after field sync	Wrong sync action used, incomplete sync, or desktop project not refreshed	Confirm mobile sync success, then refresh/synchronize again from QFieldSync in QGIS 3.44.11
QGIS prompts for a master password unexpectedly	First-time use of stored credentials through the QGIS authentication system	Set and document the master authentication password according to organizational policy

Issue	Likely Cause	Recommended Action
GPS geometry appears offset or measurements seem wrong	Unsuitable CRS choice or misunderstood transformation behavior	Review project CRS, transformation settings, and whether a local projected CRS should be used

7.7 Known Workflow Caveats

- Not every valid QGIS layer source is suitable for offline mobile editing.
- Wizard Mode only works when the form is organized into tabs.
- Attachment workflows depend on relative paths and correct project subfolder handling.
- Expressions using layer IDs may break after QFieldSync conversion.
- Desktop cartography that looks excellent on a large monitor can be too slow or cluttered on a phone.
- Legacy formats such as Shapefile may remain useful for exchange, but they are rarely ideal as the main editable source for field projects.

8. Appendices

8.1 Useful Keyboard Shortcuts in QGIS 3.44.11

The following are common defaults, but **keyboard shortcuts are configurable in Settings > Keyboard Shortcuts**. If your organization has customized shortcuts, those custom mappings take precedence.

Shortcut	Typical Default Function
Ctrl + N	New project
Ctrl + O	Open project
Ctrl + S	Save project
Ctrl + Shift + S	Save project as
Ctrl + Z	Undo
Ctrl + Y	Redo
Spacebar	Temporary pan in many navigation contexts
F3	Open locator/search bar in common configurations

8.2 Recommended Plugins

Plugin	Purpose	Recommendation for This Workflow
QFieldSync	Prepare, package, and synchronize QGIS projects for QField and QFieldCloud	Essential core plugin

Plugin	Purpose	Recommendation for This Workflow
QuickMapServices	Convenient access to common web basemaps	Useful during project design and orientation
DB Manager or database-management tools available in your environment	Inspect GeoPackage and database layers	Helpful for enterprise or QA workflows
Geometry/validation tools available in your QGIS setup	Quality assurance before and after field synchronization	Strongly recommended for production projects

8.3 Additional Resources and Documentation Links

- [QGIS Download Page](#)
- [QGIS User Manual](#)
- [QField Documentation](#)
- [QFieldCloud Getting Started Guide](#)
- [QFieldCloud Authentication Reference](#)
- [QFieldSync Plugin Repository Page](#)
- [QField Attribute Form Setup Guide](#)
- [QGIS 3.44 Visual Changelog](#)

8.4 QGIS 3.44 Features Worth Exploring Later

Feature Area	What to Explore	Why It Matters for Field Programs
Processing	Native Fill Sinks Wang & Liu, Raster rank, raster calculator creation options, batch temporary outputs, modeler toolbox consolidation	Useful for pre-field terrain preparation and post-field analysis workflows
PostgreSQL browser improvements	Stored project management, schema restriction, rename field, move table to schema, stored SQL queries	Improves enterprise GIS and database-backed field project maintenance
GPS and plugin APIs	QgsAppGpsTools exposure in Python/plugin API	Relevant for custom institutional integrations
CRS expressions	<code>crs_from_text</code> and <code>crs_to_authid</code>	Helpful in advanced expression-driven workflows
Form designer improvements	Search box, alias toggle, merge policies	Directly useful when maintaining large field forms

Closing Recommendation

A successful QGIS-to-QField workflow in **QGIS 3.44.11** depends less on the upload step than on disciplined project preparation. If your data storage, forms, constraints, attachments, CRS choices, and mobile symbology are designed carefully in QGIS, then field collection in QField becomes faster, cleaner, and easier to synchronize. For production deployment, always run a pilot cycle with one project, one device, and one test user before rolling the workflow out to the full team.