iMobile

Formata Tracker

Product Description and Functional Specification

Document ID	IM.TRACKER.INT.01
Document version	1.0
Date of issue	2026-10-06
Applies to	Formata Tracker 1.0
Language	English
Publisher	iMobile

© 2026 iMobile. https://new-imobile.com

Contents

1. General information . . . 3
1.1. Name and purpose . . . 3
1.2. Field of use . . . 3
1.3. Software class . . . 3
2. Functions . . . 3
2.1. Projects and issues . . . 3
2.2. Rich text . . . 4
2.3. Workflows . . . 4
2.4. Steps by work type . . . 4
2.5. Query string and filter templates . . . 4
2.6. Sprints, backlog and board . . . 5
2.7. Labels . . . 5
2.8. Time and estimates . . . 5
2.9. Reports . . . 5
2.10. Dashboards . . . 6
2.11. Wiki . . . 6
2.12. Access control . . . 6
2.13. Organisational structure . . . 6
2.14. Integration with Gitea . . . 6
2.15. Notifications . . . 6
3. Architecture and components . . . 6
4. Interfaces and data exchange . . . 7
5. Editions and limitations . . . 7
6. Environment requirements . . . 7

© 2026 iMobile

About this document

This document describes the purpose, functions, components and interfaces of Formata Tracker version 1.0, an application for the Formata platform used to track issues, plan work, keep a knowledge base and report on the work of teams.

It is written for the customer, the manager of the implementation project and the administrator.

1. General information

1.1. Name and purpose

Full name: Formata Tracker. The program records issues and manages their life cycle through configurable workflows, plans work in sprints, records time spent, keeps a knowledge base (wiki), and builds reports and dashboards by team, department and project.

1.2. Field of use

Software development units, support services, project offices and other units that organise their work as issues with assignees, due dates and statuses. The program runs on the customer's server; users work with it in a web browser.

1.3. Software class

An application for the Formata server platform: the data structure, logic, forms, queries, scripts, menu and translation dictionary of the program are stored in a PostgreSQL database and executed by the Formata server. The program belongs to the class of issue and project tracking systems (issue trackers) with a built-in knowledge base.

2. Functions

2.1. Projects and issues

- Projects have a key made of Latin letters and digits (for example `PLT`); issues are numbered within their project (`PLT-12`).
- Issue types: Epic, Story, Task, Error (code `bug`), Sub-task; the administrator can add further types.
- Issue hierarchy of any depth: sub-tasks, navigation to the parent issue and its ancestors (a trail of links above the card).
- Issue fields: title, description, type, priority (five levels), assignee, reporter, watchers, sprint, team, labels, estimate in points and in hours, remaining time, start and due dates, resolution.
- Issue links: blocks, linked to, duplicates, copy; a link is shown in both issues.
- Comments, attachments, links to issues, wiki pages and projects.
- Issue change history: every field (old and new value), status transitions, labels, sprint, estimates and dates, comments (added, edited, deleted), links, attachments, logged time. For the title, the description and comments the full text is kept, and versions are compared word by word.
- The Mentioned in block: issues, comments and wiki pages that contain a link to the issue.

© 2026 iMobile

2.2. Rich text

Issue descriptions, comments, wiki pages and dashboard notes are entered in a rich-text editor: headings, text styles, inline code and code blocks, quotes, colour, alignment, lists and checklists, tables, links, horizontal rules, images (pasted from the clipboard or dragged in) that enlarge on click. Before the HTML is saved, the server cleans it against an allow-list of tags and attributes.

2.3. Workflows

- A workflow is a set of statuses and transitions. A status has a category (to do, in progress, done), a colour, a description, an initial-status flag, a waiting flag (time spent in the status counts as idle time), an issue limit (WIP) and a work type.
- A transition defines the status an issue is moved from (or any status) and the status it is moved to, the minimum project role and the "assignee only" flag.
- Transition conditions: a mandatory comment, an assignee, an estimate, logged time, closed sub-tasks, no open blocking issues, a chosen resolution, completed steps of the current work type.
- Automatic actions on a transition: assign the issue to the user who makes the transition, to the reporter or to the project lead, or remove the assignee; set the default resolution; record the start date on the first transition to a status of the "in progress" category.
- Resolutions: Done, Won't be done, Duplicate, Cannot reproduce, Insufficient data.
- A diagram of the workflow on screen, a workflow check (unreachable statuses, dead ends, no status of the "done" category), and the time an issue has spent in each status.
- Four workflows are supplied: *Software development (IT)* (nine statuses from backlog to release, with blocking and rejection), *Development*, *Simple* and *Support*. A workflow is assigned to a project or to an issue type; a workflow without transitions is free: an issue can be moved to any status.

2.4. Steps by work type

- A workflow status is given a work type: Development, Code review, Testing, Acceptance or a custom one.
- In the issue card, versions of steps are kept for each work type (the version heading is a build number or a title), each version with its own checks. The "to-dos" view uses a checkbox; the "test cases" view marks a step as passed or failed, with a note.
- Each version shows an overall status: a progress bar, a colour-coded completion percentage and the total "N of M done".
- Standard step sets are taken from a wiki space: list items on the pages become steps, or a whole page becomes a single step with a link to it.
- Step permissions are set on the work type: who writes steps and who checks them off (a project role no lower than the one specified).

2.5. Query string and filter templates

- An issue filter is a query string: conditions of the form "field operator value", the connectives `AND`, `OR`, `NOT`, brackets and sorting with `ORDER BY`. Words without an operator are searched for in issue titles and descriptions.
- Fields of the query language: project, type, status, status category, priority, assignee, reporter, watcher, sprint, label, team, department, resolution, issue key, parent issue, text, title, the dates of creation, update and resolution, the due and start dates, estimates in points and hours, remaining time, logged time (26 fields in total).

© 2026 iMobile

- Dictionary values are accepted by code, key, login or name in any of the interface languages; the functions `me()`, `openSprints()`, `futureSprints()`, `closedSprints()`, `myTeams()`, `today()`, `startOfWeek()`, `endOfWeek()`, `startOfMonth()`, `endOfMonth()`, `startOfYear()`, `endOfYear()` are available; dates are written as `2026-11-30` or `30.11.2026`, or as offsets from the current date (`-7d`, `+2w`, `-1m`).
- Hints while typing: fields, the operators allowed for a field, field values, connectives.
- Visual builder: condition groups "all conditions (AND)" and "any condition (OR)", negation of a group, conditions "field — operator — values", sorting; the query string is built automatically.
- Syntax errors are shown while the string is typed, value errors after the query is run; the message gives the character position.
- Filter templates: personal and shared (visible to all users). Twelve templates are supplied (*My open issues*, *Overdue*, *In open sprints*, *Done in the last 7 days* and others). A template is changed and deleted by its author; the supplied templates by the tracker administrator.
- The query string is used in the issue list, in the Logged time, Issue completion and Completion by team and department reports, and on dashboards (for the whole dashboard and for an individual panel).

2.6. Sprints, backlog and board

- Project backlog, moving selected issues into a sprint and back, sprint goal and dates.
- Starting and closing a sprint; when a sprint is closed, unfinished issues are moved to a chosen sprint or to the backlog.
- Sprint progress, burndown chart, team velocity over recent sprints.
- A board by the statuses of the project's workflow: moving a card to another column performs a transition according to the workflow rules; a card shows the priority colour, labels, key, assignee, due date and estimate; a column over its WIP limit is highlighted.

2.7. Labels

Label groups, including "one of group" groups (a new label replaces the previous one). Rules: "requires" (the required label is added automatically), "excludes" (the labels cannot be used together), parent label (a child label can be set only together with it). Global labels and project labels. The program tells the user what was changed by the rules.

2.8. Time and estimates

Estimates in points and in hours, remaining time in hours. Time is logged with a date and a comment; input formats are `1h 30m`, `1:30`, `1.5`, `90m`, `1d` (a day equals the employee's daily working hours). The remaining time decreases automatically or is set by hand. Due dates are shown with traffic-light colours.

2.9. Reports

- Logged time: totals, by employee, by day, all entries for a period or a sprint; export to CSV.
- Issue completion: totals, by assignee, overdue issues, issues completed late.
- Completion by team and department: weekly charts of completed work in estimate points.
- Report filters: period, project, sprint, team, department, employee and query string.

© 2026 iMobile

2.10. Dashboards

Any number of dashboards: personal, shared with everyone, and available to individual employees, access groups or teams (to view or to configure). Twenty-seven metrics: numbers, charts, tables, scales, notes. Filters of a dashboard and of an individual panel: project, sprint, team, department, period, "only mine" and query string. A My work dashboard is created automatically for every user.

2.11. Wiki

- Spaces (global or belonging to a project) with a page tree, subpages, page order, attachments.
- Page versions: every save creates a version (author, time, a "what changed" comment); any version can be compared with any other word by word; a version can be restored; another user's edit is protected from being overwritten.
- Export of a page, a branch or a space to HTML (images embedded), Markdown (an archive with images and attachments) or a Word document (.doc); printing to PDF from the browser.

2.12. Access control

- Project roles: viewer (reads), reporter (creates issues and comments), developer (edits issues, makes status transitions, logs time), manager (sprints, settings, access). Access is granted to an employee, an access group or a team.
- Closed projects (access by permission only) and internal projects (visible to all employees).
- Wiki space roles (reader, editor, space administrator) and dashboard permissions.
- Tracker roles: tracker administrator, project lead.
- Permissions are checked on the server, including when attachments are served and when a query string is run.

2.13. Organisational structure

Departments in a tree, teams (membership is changed by the team lead), access groups, employees. User accounts are created by the tracker administrator.

2.14. Integration with Gitea

The program receives a signed webhook (HMAC-SHA256) from the Gitea system and sends nothing to Gitea. Branches, commits and pull requests that have an issue key (for example `PLT-12`) in the branch name, the commit message, or the pull request title or description are shown in the issue card and in the issue history. Workflow rules move an issue on the events "branch created", "commit", "opened pull request", "pull request merged" and "pull request closed without merging", if the transition is allowed and its conditions are met. The smart commits `#time` and `#comment` log time and add a comment on behalf of the employee with the given Gitea login.

2.15. Notifications

Notifications about issue events are shown in the interface (the bell icon) in the language of the user who performed the action. Notifications are not sent by email.

3. Architecture and components

The program consists of:

- the Formata platform server version 1.7 (`crm_server`) and the platform web client;

© 2026 iMobile

- the tracker application: SQL files that create a separate schema in the database (`tracker` by default) with tables, PL/pgSQL functions, named queries, forms, Python scripts, a menu and a translation dictionary;
- modules of the Formata CRM application that the tracker uses as common shell components (user accounts, messages, links to documents, notifications, printing).

All application logic of the tracker (permissions, status transitions, label rules, query-string parsing, metric calculation) is implemented as PostgreSQL functions; form scripts collect the entered data and call a single write function. An action therefore has the same result whether it is performed through the forms or through the server interface.

4. Interfaces and data exchange

- User interface: a web browser; HTTP, with access over HTTPS through the customer's reverse proxy.
- Inbound Gitea webhook: `POST /api/inbound/gitea`, with an HMAC-SHA256 signature in a header; the secret is set in the tracker settings.
- Links to tracker objects inside texts and notifications are addresses of the form `formata://trk_issue/<number>`.
- Exports: CSV (issue list, logged time), HTML, Markdown and Word (wiki pages).
- Interfaces of the Formata platform (integration API, named queries) are described in the platform documentation.

5. Editions and limitations

The program is supplied in a single edition. Limitations of version 1.0:

- notifications are not sent by email;
- there is no built-in import of data from other trackers;
- PDF export is done by printing the HTML export from the browser; Word export produces a .doc file;
- Gitea is the only supported source-control system, and the integration is inbound only;
- team charts count completed work by resolution date: an issue that is reopened is removed from the week in which it was completed;
- demonstration data is supplied in Russian and English.

6. Environment requirements

A server running Ubuntu 24.04 LTS or 22.04 LTS (or a container based on Ubuntu 24.04), PostgreSQL 16 with the `pgcrypto` extension, and the Formata platform server version 1.7. User workstation: a Chromium-based web browser. Detailed requirements and the installation procedure are given in the Installation and Administration Guide.

Document history

Document version	Date	Product version	Change
1.0	2026-10-06	1.0	First issue

© 2026 iMobile