

AMG Coffee Management System

Coffee Management System for Odoo

ODOO







17.0

LICENSE

LGPL-3

A comprehensive Odoo module to manage the end-to-end coffee supply chain, from raw bean procurement to finished product sales. This module introduces a highly specialized workflow tailored for the coffee industry, integrating deeply with core Odoo applications.

Key Features

-  **Specialized Procurement Workflow:** A unique, multi-step process for handling coffee arrivals, from quality to weight to final stock reception, capturing far more detail than the standard purchase workflow.
 -  **Dynamic Product Generation:** Automatically creates new, distinctly graded `product.product` variants based on quality control results. This ensures your inventory is always tracked by its true, evaluated grade, not a generic name.
 -  **Direct Manufacturing Control:** Bypasses Odoo's standard MTO route automation for a more explicit, one-click creation of both Delivery and Manufacturing Orders directly from a `coffee.contract`.
 -  **Intelligent Component Checking:** Proactively checks for raw material shortages before confirming manufacturing orders and creates warning activities for planners to prevent production delays.
 -  **Advanced Reporting Suite:** Includes a full suite of wizard-based PDF reports for key metrics and on-screen analytics (Pivot & Graph views) for all major operations.
 -  **Granular Data Classification:** Leverages custom models for `Coffee Type`, `Origin`, `ECX Grade`, `Zones`, and `Woredas` to build a highly structured and detailed database of your coffee products.
-

workflows Workflow Deep Dive

This module introduces two primary, interconnected workflows that are highly customized for the coffee industry.

1. Procurement: From Arrival to Graded Stock

This flow replaces the standard RFQ -> Receipt process with a more granular set of models and steps.

1. **`coffee.arrival`:** The process begins. A user creates an arrival record, selecting an `ecx.coffee` master record which represents the expected coffee classification.
2. **`coffee.quality.evaluation`:** A quality check is performed. Upon saving the evaluation, the system calculates a final **AMG Grade**.
3. **Dynamic Product Creation:** The system then constructs a product name based on the Origin, Type, and final AMG Grade (e.g., `Raw-Guji-Washed-G1-Coffee`).
 - If a `product.product` with this name already exists, it's linked to the arrival.

- If not, a **new product.product is created automatically**. This is a key feature that ensures inventory is tracked by its true, evaluated grade.
- 4. **coffee.weight.history**: The batch weight is recorded, and a permanent log is created in `coffee.weight.history.log`.
- 5. **coffee.stock.receiving**: A Goods Receiving Note (GRN) is created. When confirmed, a **stock.picking** is created and validated, moving the specific, graded product into a warehouse location.

2. Sales & Manufacturing: The Direct Control Approach

This flow uses the custom `coffee.contract` model, which gives the user explicit control over the creation of downstream documents, rather than relying on Odoo's background MTO automation.

1. **coffee.contract**: A user creates a sales contract for a customer, adding finished goods to the `coffee.contract.line`.
2. **action_confirm_contract**: This button executes a powerful custom logic:
 - **Creates a Delivery Order (stock.picking)** for all lines on the contract.
 - Simultaneously, for each coffee product line that has a valid Bill of Materials (BoM), it **manually creates a Manufacturing Order (mrp.production)**.
 - It performs a **component availability check**. If raw materials are short, the MO is created in a "Draft" state, and an `mail.activity` is scheduled to warn the user. Otherwise, the MO is confirmed.
 - The contract state is moved to **Confirmed**.

Setup & Configuration

Dependencies

Ensure the following Odoo applications are installed (as listed in `__manifest__.py`):

- `stock`
- `mrp`
- `purchase`
- `sale`
- `product`
- `contacts`
- `account`
- `mail`

Configuration Steps

A one-time configuration of master data is required for the module to function correctly.

1. **Geographical Data**: Navigate to **Coffee Management > Configuration** and populate the master data for:
 - `Zones/Province`
 - `Woredas/District`
2. **Coffee Classification Data**: In the same configuration menu, define your company's master data for:

- Coffee Types
- Coffee Origins
- ECX Coffee Grades

3. **ECX Product Master:** In **Configuration > ECX Coffee Products**, create the base combinations of Origin, Type, and Grade. These are used as templates for the dynamic product creation during the arrival process.
4. **Finished Goods & BoMs:** For any coffee product you intend to manufacture and sell (e.g., "Roasted Guji G1 Coffee"), you must manually create it as a **Storable Product** and define a Bill of Materials (BoM) for it. The raw material components used in the BoM should be the products that are dynamically created by the arrival workflow.

Module Structure

The module is organized into the following key components:

- **models/:** Contains all Python logic for the custom models.
 - **coffee_arrival.py:** Manages incoming raw coffee deliveries.
 - **coffee_quality.py:** Logic for the quality evaluation process and dynamic product creation.
 - **coffee_weight.py:** Handles weight recording and logging.
 - **coffee_stock_receiving.py:** Manages the Goods Receiving Note (GRN) process.
 - **coffee_contract.py:** The core of the sales workflow, with logic for creating deliveries and MOs.
 - **coffee_classification.py, coffee_location.py, ecx_coffee.py:** Define the master data models.
 - **report_coffee.py:** Contains the logic for the wizard-based PDF reports.
- **views/:** Contains all XML definitions for the user interface.
 - **coffee_arrival_views.xml, coffee_contract_views.xml, etc.:** Define the primary form and tree views for each model.
 - **report_views.xml:** Defines the wizard for the date-range reports.
 - **report_analytics_views.xml:** Defines the pivot and graph views for on-screen analysis.
 - **menuitems.xml:** Defines the application's menu structure.
- **reports/:** Contains the QWeb templates and definitions for printable PDF reports.
 - **coffee_reports.xml:** Defines the **ir.actions.report** records.
 - **coffee_report_templates.xml:** Defines the visual HTML layout for each PDF report.
- **security/:** Defines the access rights.
 - **ir.model.access.csv:** Contains the model-level security rules.
- **data/:**
 - **coffee_data.xml:** Defines **ir.sequence** records for auto-numbering documents.
- **static/description/:**
 - **index.html:** The main description page for the Odoo App Store.

- `icon.png`: The module's icon.
-

License

This module is licensed under the LGPL-3 License.