Summary  
InvenTree

Vaishnavi Agnihotri

# Table of Contents

1. **About**
2. **Key Features**
   1. **Parts**
   2. **Suppliers**
   3. **Instant Stock Knowledge**
   4. **BOM Management**
   5. **Build Parts**
   6. **Reports**
   7. **Extend and Customize**

## What is InvenTree?

We have huge sets of data around us. If there could be someone who could effectively manage and sort the data and present it in a very organized manner, it would be Great! Here InvenTree comes to rescue.

It is an open source Inventory management system. It is lightweight and ensures that the stock tracking history is maintained. It is a Python based application which could be accessed through the web browser or could be integrated into applications through API. The installation is very simple and easy.

## Features:

## Part

It is the most basic and fundamental unit of the InvenTree ecosystem. It is a sort of example or a simple model of the stock items in the inventory. It is arranged in a hierarchy so that we could organize and filter it.

Categories:

It displays a list of all parts and also the sub-categories that come into a category. It could be arranged as per user. It displays Part Name, description, category, and stock level. These parts could be easily filtered out using various parameters.

Options:

We get a set of options with Parts. Some of them are discussed below:

a. Virtual: It does not exist physically but still should be tracked by the system.

b. Template: It is used to create a base-variant, and contains information that could be used by variants. It dictates a special configuration and determines options that come with part, like color, specs, harnesses. Templates provide certain options like having unique serial numbers to parts linked with template-variant relationships, including stocks of all variants in a template, and local grouping.

We can enable the Template option and create variants.

c. Assembly: Can be created or built from other components. The Bill of Materials defines the list of components required to make an assembly. The BOM contains some line items, which have various properties.

d. Trackability: assigned batch numbers which uniquely identify a particular stock item. For faster input there are several ways to define the wanted serial numbers. The rules could be manipulated by using whitespaces or commas separating them.

e. Salable and Purchasable: if a part could be purchased from external suppliers, it is called as purchasable and if a part could be sold to customers, it is called salable.

f. Active: if a part is marked inactive, it is not available for some actions. By default, all parts are marked as active.

The parts could be imported by staff members on the part list view, part settings or admin page for parts.

## Suppliers:

A Supplier is a vendor of parts and raw materials. We can add, edit or delete the suppliers or the parts provided by a particular supplier. We can also check the availability of a supplier part by updating information manually or by using it in conjunction with the InvenTree plugin system.

## Stock Location:

Physical real world location where the stock items could be stored. Each location could contain multiple sub-locations or stocks.

Stock item is a physical quantity of the Part at a specific location. The stock item provides various information like its Part, location, quantity, supplier, last updated and status.

The stock tracking option ensures a complete history of the stock item is maintained.

## Bill of Materials

It is a list of materials required to make an assembly and allocate inventory. It comprises a number of BOM line items, and all of them have certain properties. If we are using template feature, we can use the inheritance feature, by which a line item could automatically be included in the BOM of every variant of a particular template or variant.

We can add the BOM item or a substitute for the same manually. We can also upload files and match the BOM file columns with InvenTree part fields. After creating/uploading, we also need to validate it.

## Build Parts

Create a new stock. It uses the BOM to allocate stock items to the assembly. We can have a table or a calendar view for displaying the build orders.

The build parameters are available for each build order, which consists of reference, description, part, quantity etc.

Build Output creates a new stock instance of a specified quantity.

Stock allocation: it refers that the stock item will be removed from the database once build order is completed. Inventree provides an allocation interface which attempts to keep user interactions to a minimum. There are tracked and untracked stocks based on their persistence and presence in the database. We can do stack allocation manually, edit, unallocate, or allocate automatically using auto allocate. There are different options for tracked and untracked stocks.

The detail view for a single build order provides multiple display tabs, like build details, allocate stock, build outputs, child builds, attachments, and notes.

If incomplete build outputs remain, the build order will not be able to be completed. All build outputs must be completed first.We can also cancel the order or consider it overdue if it remains incomplete after the due date.

## Reports

InvenTree allows users to develop reporting templates. They are generated from HTML/ CSS and are used to format data. It utilizes the WeazyPrint pdf generation engine.

Each report template is provided with a set of context variables which can be used when rendering the template. We can also do conditional rendering as we use the Django framework which allows this.

A number of global reporting options are available. We can enable reporting feature, set page size of reports, set up the debug mode before generating the pdf, upload templates, upload asset files, include external template files, and report types.

## Extend and customize

InvenTree provides a powerful REST API for interacting with inventory data on the server. For this, users must be authenticated, and should have username, password and token authentication.

Each user has an authentication token which could be used across multiple sessions. We can request a token via the API itself by a GET request.

Users can only perform REST API actions which align with their assigned role permissions.

If an API action outside of the user's role(s) is attempted, the server will respond with a 403 permission error message.