Inventory

Purpose: Inventory is the FOLIO app where bibliographic information from a variety sources can be presented in a uniform, abstracted form for management of the collection regardless of the format or content rules used to describe a resource. It is a sort of staff discovery layer, useful for search and identification. As such, Inventory represents bibliographic, holdings, and item data and integrates with apps like Order, Check in, Check out, and Request. Inventory exists primarily to manage collections (physical and/or virtual). Some types of records for library resources may also be managed in other FOLIO apps or outside of FOLIO.

Inventory comprises four record types: Instance, Holdings, Item, and Container. (Container is not yet implemented as of December 2019)

Instance records may be derived from full bibliographic records (in MARC or other formats) and are intended to provide enough information for library staff to identify & select records in order to perform work on associated holdings and items. Instance records can also be in the native FOLIO format when full bibliographic description is not required.

Holdings records provide information needed, such as location, call number, and volumes owned, for staff to locate and manage library holdings. Holdings records may describe library holdings that are physical, electronic, or other formats. Holdings records are created and edited in Inventory.

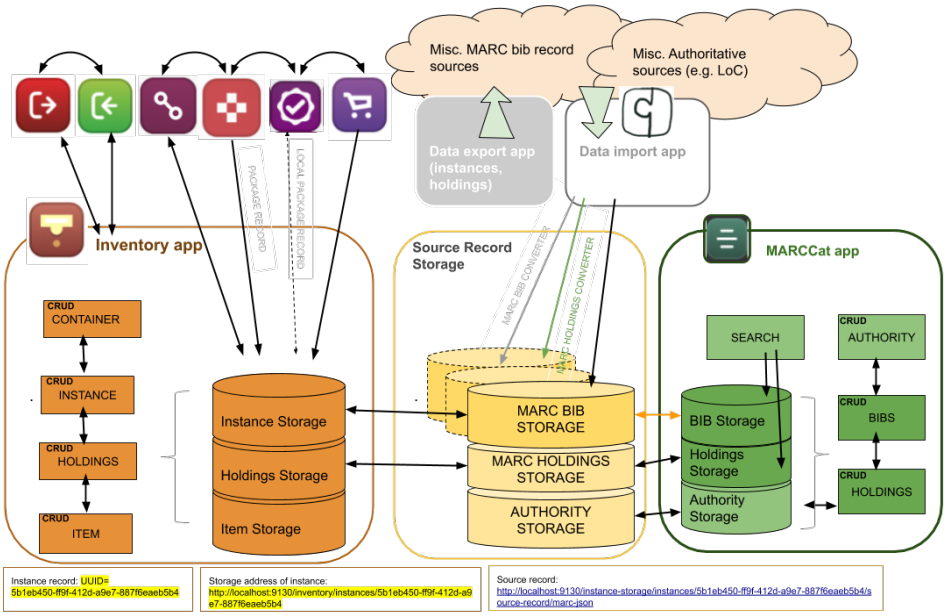
An Item record provides the information needed to identify and track a single item or piece, such as barcode, availability, and material type. Item records are created and edited in Inventory.

Container records (not yet implemented as of 3/27/2020) will hold multiple Instance, Holdings, and/or Item records and serve as virtual containers, a way to represent an additional level of hierarchy and associate data that may not have bibliographic associations (such as membership plans or standing orders). Containers help staff manage materials that may be acquired, paid for, or managed in one way but described in another.

While Inventory stores bibliographic data, it is not a cataloging module. Inventory is essentially a hub, bridging the gap between other functions (such as circulation or acquisitions) and catalog data. Holdings and items data for patron-facing discovery will come from Inventory, but bibliographic data for discovery will usually come from a richer source record (such as a MARC record).

Areas of Focus

Inventory Storage MARCCat diagram courtesy of Charlotte Whitt



Note 1: Arrows indicate direction of data flow
Note 2: Yellow boxes indicate separate storage modules. GENERIC STORAGE is the source record, and for V1 it is MARC Bib Storage; post V1 we'll have BIBFRAME storage etc.
Note 3: Inventory holds the storage layer (MOD-INVSTOR primarily backend functionality.)
Note 4: Two frontend features: Import and Export apps.
Note 5: The exchange of records between MARCCat Storage and GENERIC BIB STORAGE will go via REST-APIs e.g. to connect MARCcat app to the bib storage.
Note 6: Dotted lines - are features to be developed post-V1. E.g. The connection between container in Inventory and ERM (package) will probably take some time, at the moment the subgroup isn't considering it as a v1 ERM feature.
Note 7: 3 scenarios:

a) Some instances lives ONLY in Inventory - no sync is needed
b) Some instances are created by MARC, with MARC as source of truth (batch load, MARCcat)
c) Some instances starts without MARC; e.g. brief order record, fast add record, which later will be upgraded to a completed bib record updated in MARCcat. Identification is done on UUID.

Note 8: UUIDs across the apps:
* Instance record: UUID= 5b1eb450-ff9f-412d-a9e7-887f6eaeb5b4 [Inventory, and brief order record, fast-add record (To be confirmed)]
* Storage address of instance: http://localhost:9130/inventory/instances/5b1eb450-ff9f-412d-a9e7-887f6eaeb5b4
* Address of its source record: http://localhost:9130/instance-storage/instances/5b1eb450-ff9f-412d-a9e7-887f6eaeb5b4/source-record/marc-json
* Address of the MARC bib version: http://...marccat ….5b1eb450-ff9f-412d-a9e7-887f6eaeb5b4/source-record/marc-json (TBD)
* Address of the BIBFRAME bib version: http://...bibframe ….5b1eb450-ff9f-412d-a9e7-887f6eaeb5b4/source-record/marc-json (TBD)