Mage TITANS

MCR 2017

IGOR MINIAILO

Magento 2 Architect at Magento

@iminyaylo

Architecture and workflow of the multi-source inventory (MSI) project

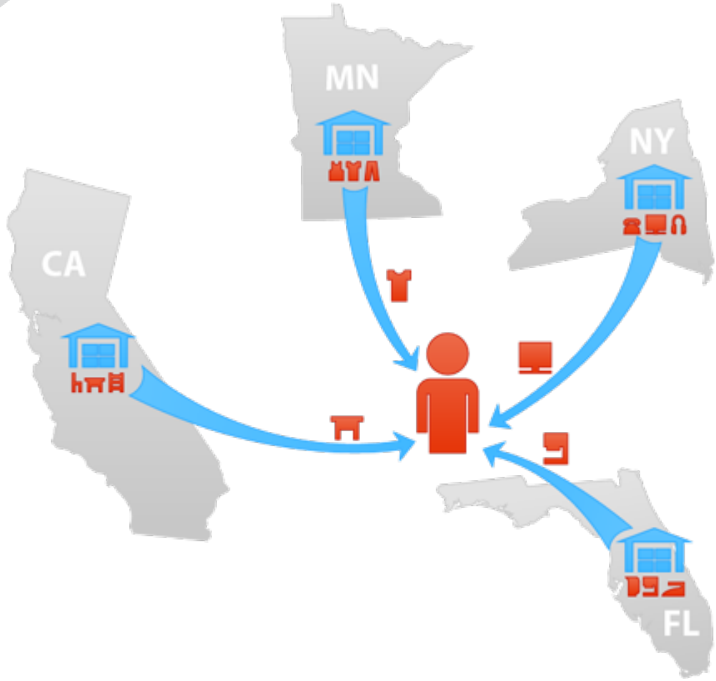
UK.MAGETITANS.COM

#MageTitansMCR @MageTitans

Mage CONF '16
FOCUS ON TECH
SPEAKER
Mage CONF '16
FOCUS ON TECH

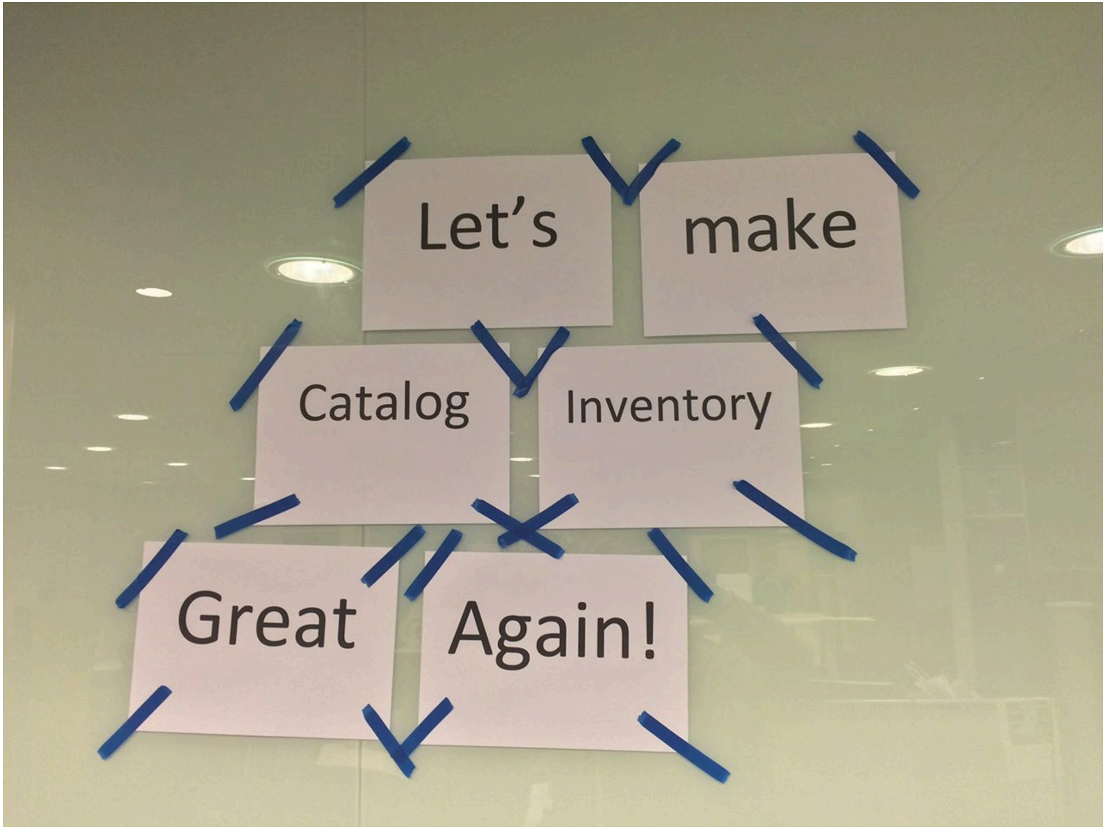
Multi-Source Inventory (MSI)

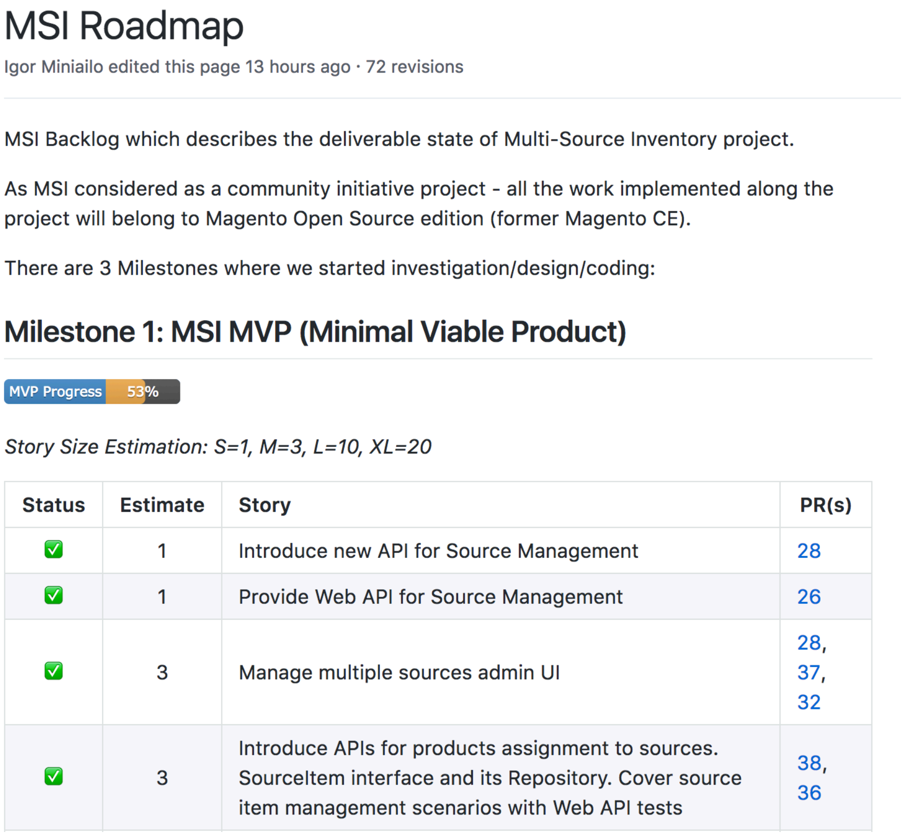
MSI is designed to enable stock management in multiple locations so that merchants can properly reflect their physical warehouses in Magento system without external extensions or customization



Split the inventory between the sources within one Magento installation

Replace current CatalogInventory with MSI





MSI Roadmap

Igor Miniailo edited this page 13 hours ago · 72 revisions

MSI Backlog which describes the deliverable state of Multi-Source Inventory project.

As MSI considered as a community initiative project - all the work implemented along the project will belong to Magento Open Source edition (former Magento CE).

There are 3 Milestones where we started investigation/design/coding:

Milestone 1: MSI MVP (Minimal Viable Product)

MVP Progress 53%

Story Size Estimation: S=1, M=3, L=10, XL=20

Status	Estimate	Story	PR(s)
✅	1	Introduce new API for Source Management	28
✅	1	Provide Web API for Source Management	26
✅	3	Manage multiple sources admin UI	28, 37, 32
✅	3	Introduce APIs for products assignment to sources. SourceItem interface and its Repository. Cover source item management scenarios with Web API tests	38, 36

https://github.com/magento-engcom/msi/wiki/MSI-Roadmap

MVP MSI Business Values for Merchants

- Management of all physical locations (Sources) and products stored there (SourceItems)
  - As a merchant, I want Magento to provide an ability to manage all physical Sources where the products are stored. To get an understanding of the Stock level on each of them.

- Improve Checkout Performance (even for Single Stock merchants)
  - As a merchant, I want to eliminate excessive database locks and Inventory consistency validation during checkout process which lead to performance degradation when there is a high checkout concurrency rate. So that, Reservation mechanism introduced which supports Asynchronous Inventory Deduction.

MVP MSI Business Values for Merchants

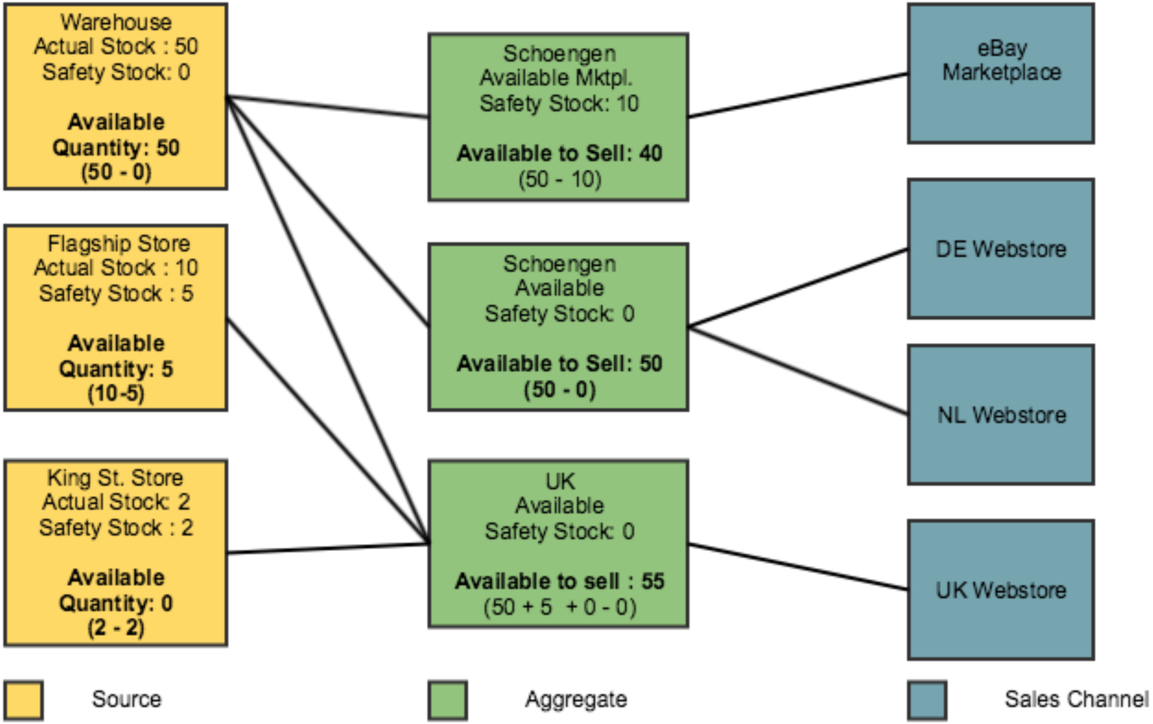
- Import/Export Stock data
  - As a merchant, I want new Import/Export process for Inventory to be introduced to speed up integration with an external ERP system.

- Drop Shipping support
  - As a merchant, I want to have an ability to setup external Source to be used for Drop shipping.

- Stock Control / Low Stock Reports
  - As a merchant, I want Magento to support Stock Control on all the physical locations (Sources) and notify me when products on some location are running out.

- No additional complication for Single Stock Merchants
  - As a Merchant who is satisfied with Magento Single Stock, I want no added complexity for UI and Stock management. The user experience for Single Stock usage should be preserved.

Base Concept



Warehouse
Actual Stock : 50
Safety Stock: 0

Available Quantity: 50 (50 - 0)

Flagship Store
Actual Stock : 10
Safety Stock : 5

Available Quantity: 5 (10-5)

King St. Store
Actual Stock: 2
Safety Stock : 2

Available Quantity: 0 (2 - 2)

Schoengen
Available Mktpl.
Safety Stock: 10

Available to Sell: 40
(50 - 10)

Schoengen
Available
Safety Stock: 0

Available to Sell: 50 (50 - 0)

UK
Available
Safety Stock: 0

Available to sell : 55
(50 + 5 + 0 - 0)

eBay Marketplace

DE Webstore

NL Webstore

UK Webstore

🟨 Source

🟩 Aggregate

🟦 Sales Channel

Checkout Action Points/Goals

- Unload the process of placing an order as much as possible
- Remove unnecessary locks (get rid of race condition between concurrent transactions)
- Improve the scalability of the checkout (shipping algorithms)

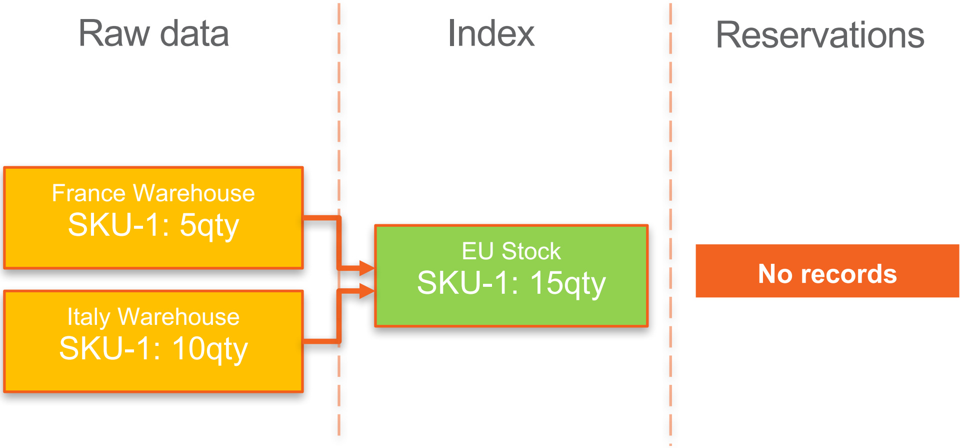
Reservation mechanism

Reservation - the entity is used when the order is placed and we need to reserve some product quantity in stock.

Reservations are append only operations and help us to prevent blocking operations and race conditions at the time of checkout.

https://github.com/magento-engcom/msi/wiki/Reservations

Order Placement – Step 1

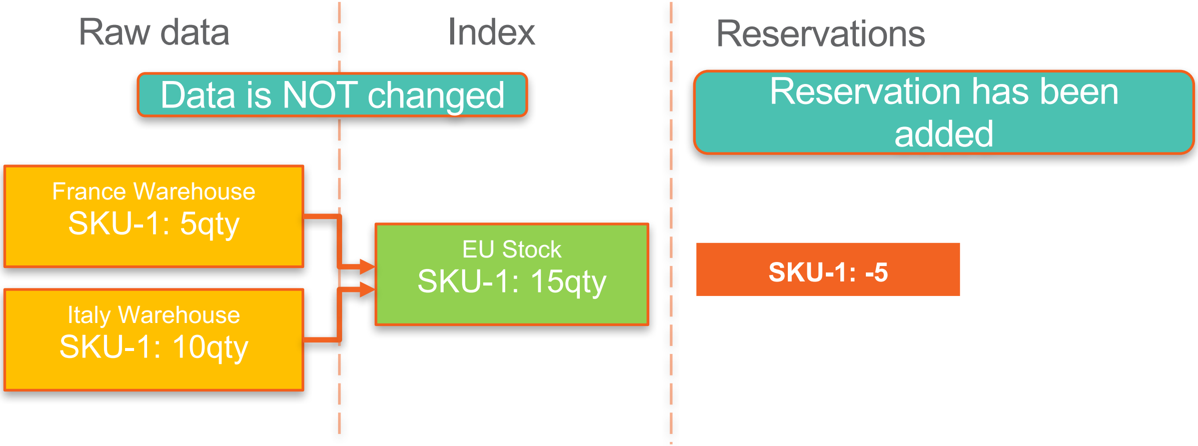


Available Qty: 15qty (data from index, empty reservations)

Order Placement – Step 2

Action: Customer buys 5 products on frontend

Order Placement – Step 3



Raw data

Index

Reservations

Data is NOT changed

Reservation has been added

France Warehouse
SKU-1: 5qty

Italy Warehouse
SKU-1: 10qty

EU Stock
SKU-1: 15qty

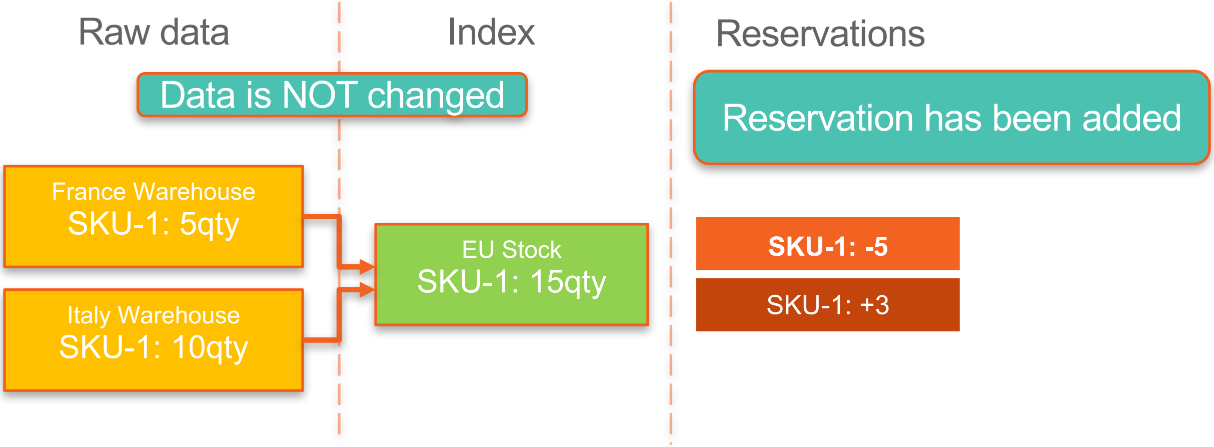
SKU-1: -5

Available Qty: 10qty (data from index 15, apply all reservations -5)

Order Placement – Step 4

Action: Admin makes a re-order 3 products out of 5 returned, and new order consists of 2 products

Order Placement – Step 5

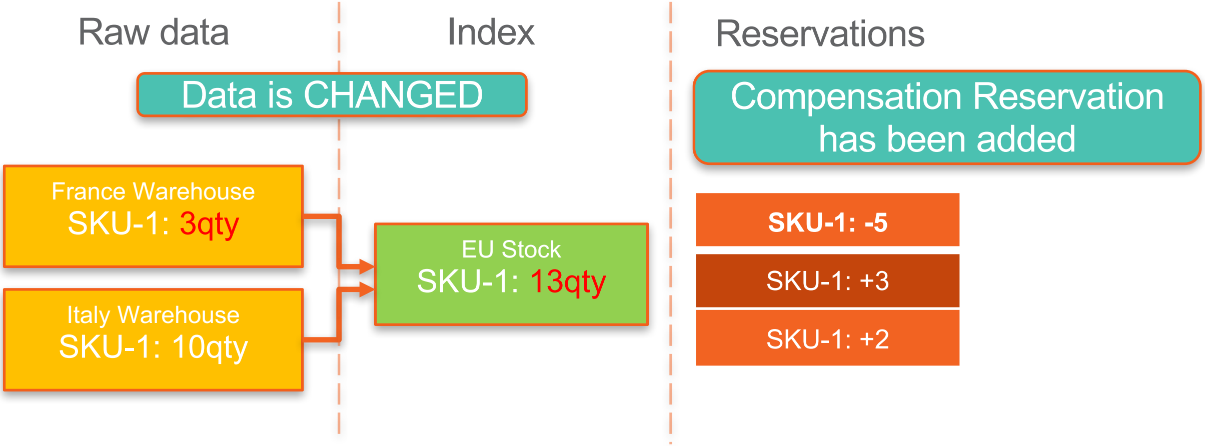


Available Qty: 13qty (data from index 15, apply reservations -5+3)

Order Placement – Step 6

Action: Admin completes order. Re-index was run.

Order Placement – Step 7



Raw data

Index

Reservations

Data is CHANGED

Compensation Reservation has been added

France Warehouse
SKU-1: 3qty

Italy Warehouse
SKU-1: 10qty

EU Stock
SKU-1: 13qty

SKU-1: -5

SKU-1: +3

SKU-1: +2

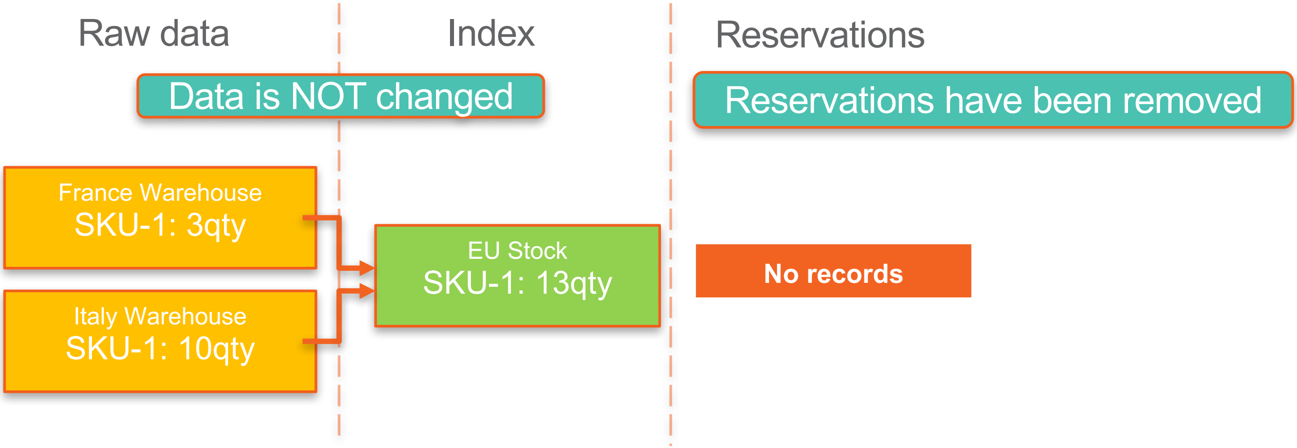
Available Qty: 13qty (data from index 13, apply reservations -5+3+2=0)

Order Placement – Step 8

Action: Reservation cleaning

Looping through these reservations we could find reservations which in sum gives 0 (Zero) and remove them.

Order Placement – Step 9 (like Step 1)



Raw data

Index

Reservations

Data is NOT changed

Reservations have been removed

France Warehouse
SKU-1: 3qty

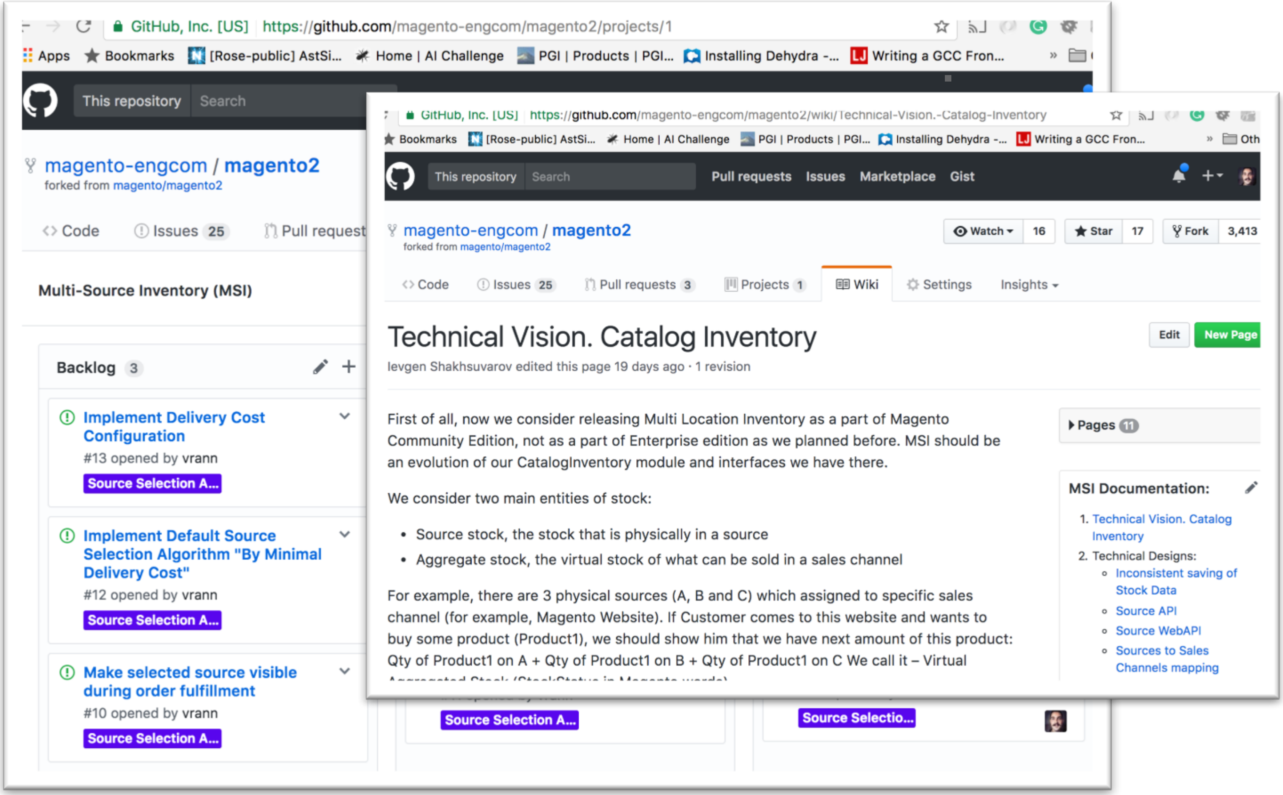
Italy Warehouse
SKU-1: 10qty

EU Stock
SKU-1: 13qty

No records

Available Qty: 13qty (data from index, empty reservations)

Kanban board and Wiki are on the GitHub



Multi-Source Inventory (MSI)

Backlog 3

Implement Delivery Cost Configuration
#13 opened by vrann
Source Selection A...

Implement Default Source Selection Algorithm "By Minimal Delivery Cost"
#12 opened by vrann
Source Selection A...

Make selected source visible during order fulfillment
#10 opened by vrann
Source Selection A...

Technical Vision. Catalog Inventory

Ievgen Shakhsuvarov edited this page 19 days ago · 1 revision

First of all, now we consider releasing Multi Location Inventory as a part of Magento Community Edition, not as a part of Enterprise edition as we planned before. MSI should be an evolution of our CatalogInventory module and interfaces we have there.

We consider two main entities of stock:

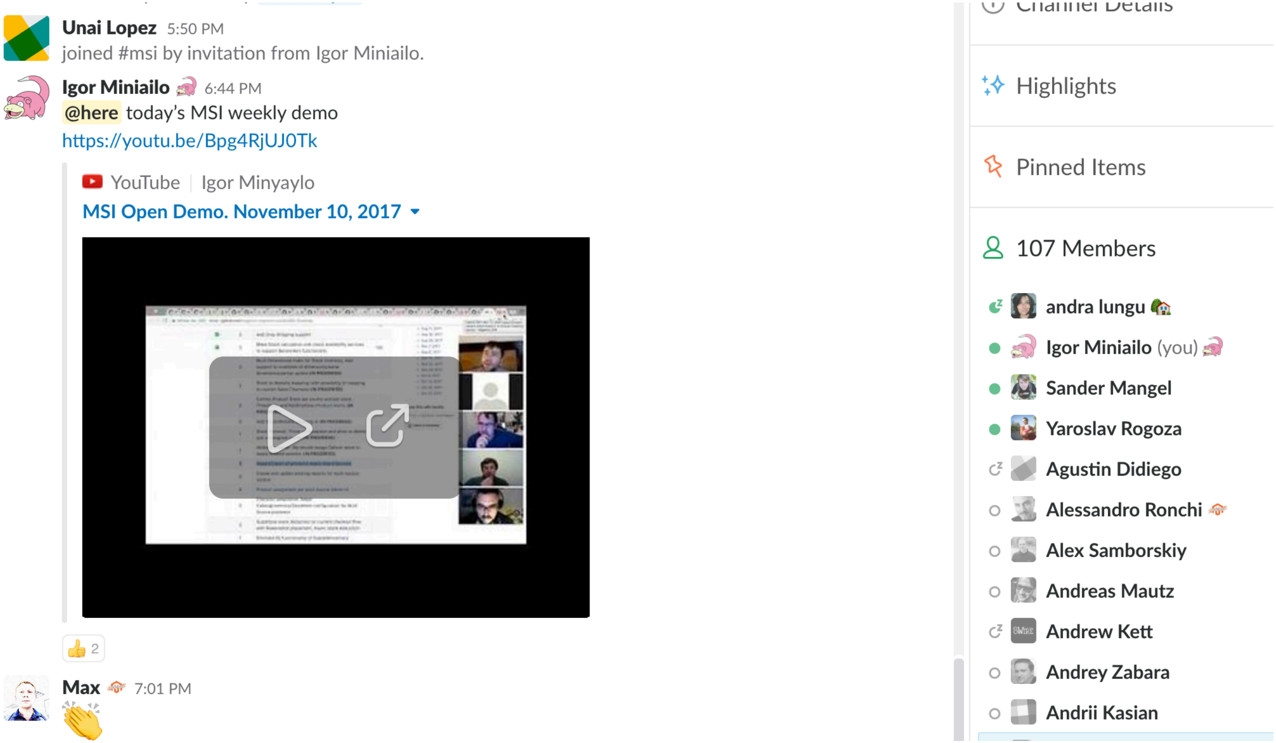
- Source stock, the stock that is physically in a source
- Aggregate stock, the virtual stock of what can be sold in a sales channel

For example, there are 3 physical sources (A, B and C) which assigned to specific sales channel (for example, Magento Website). If Customer comes to this website and wants to buy some product (Product1), we should show him that we have next amount of this product: Qty of Product1 on A + Qty of Product1 on B + Qty of Product1 on C We call it – Virtual Aggregated Stock (StockStatus in Magento words)

MSI Documentation:

1. Technical Vision. Catalog Inventory
2. Technical Designs:
   - Inconsistent saving of Stock Data
   - Source API
   - Source WebAPI
   - Sources to Sales Channels mapping

Slack Channel to Exchange ideas (#MSI)



Channel Details

Unai Lopez 5:50 PM
joined #msi by invitation from Igor Miniailo.

Igor Miniailo 6:44 PM
@here today's MSI weekly demo
https://youtu.be/Bpg4RjUJ0Tk

YouTube | Igor Minyaylo
MSI Open Demo. November 10, 2017

👍 2

Max 7:01 PM
👏

Highlights

Pinned Items

107 Members

- andra lungu
- Igor Miniailo (you)
- Sander Mangel
- Yaroslav Rogoza
- Agustin Didiego
- Alessandro Ronchi
- Alex Samborskiy
- Andreas Mautz
- Andrew Kett
- Andrey Zabara
- Andrii Kasian

Related Links

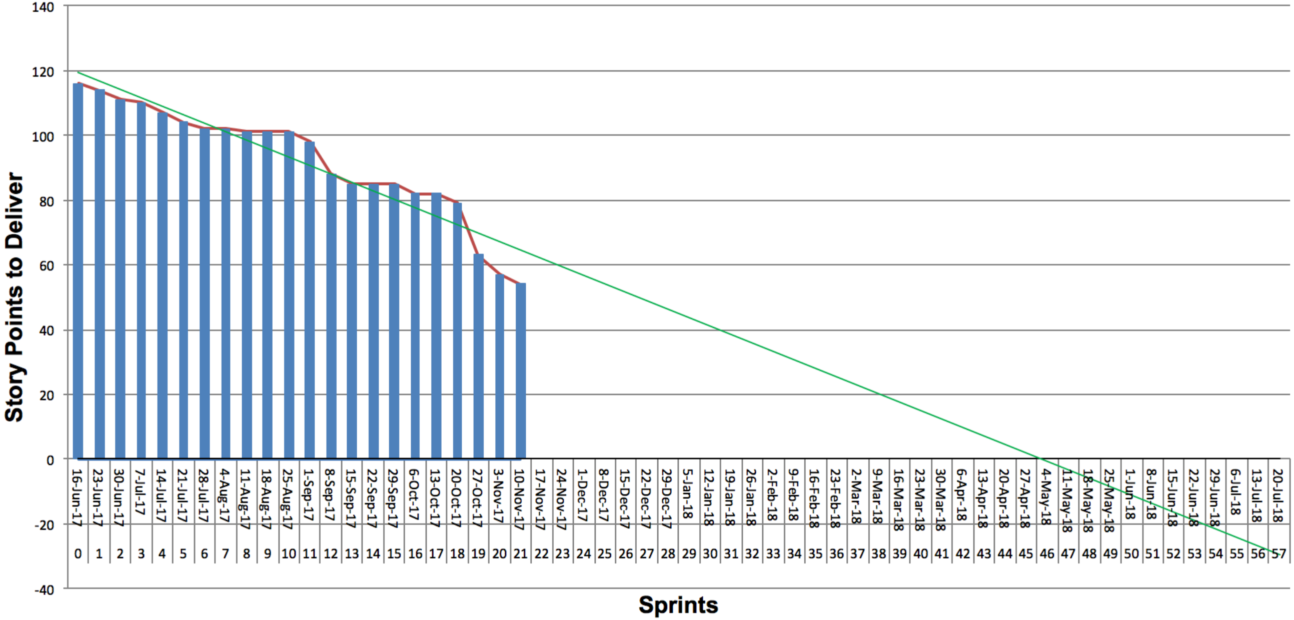
- Documentation
  https://github.com/magento-engcom/msi/wiki

- Roadmap
  https://github.com/magento-engcom/msi/wiki/MSI-Roadmap

- Dashboard
  https://github.com/magento-engcom/msi/projects/1

- Weekly Demos
  https://www.youtube.com/watch?v=ZtBNIE8FbjI&list=PLrQ5FBCRsEbWKK6U_3Awe7X-nG7KY0WPW

MSI Release Burndown



Magento

Thank y’all!

How to join us? Send an email to

engcom@magento.com

@iminyaylo 🐦