**Order Fulfillment Overview**

For implementation of Order Fulfillment, we would like to use Fulfillment Factory Module which is using Item Queue to link item and order’s information. That would be better to sort orders and update stock based on actual available products.

**Database structure**

Here is basic structure for Item Queue

|  |  |
| --- | --- |
| **Attribute** | **Note** |
| itemqueue\_id | Primary Key. auto incremental |
| order\_item\_id | `item\_id` from HP sales\_flat\_order\_item |
| order \_id | `entity\_id` from HP sales\_flat\_order |
| order\_Increment\_id | `increment\_id` from HP sales\_flat\_order |
| store\_id | `store\_id` from core\_store |
| product\_id | `product\_id` from sales\_flat\_order\_item |
| sku | `sku` from sales\_flat\_order\_item |
| name | `name` from sales\_flat\_order\_item |
| qty\_ordered | mapping from sales\_flat\_order\_item |
| qty\_backordered | mapping from sales\_flat\_order\_item |
| qty\_canceled | mapping from sales\_flat\_order\_item |
| qty\_invoiced | mapping from sales\_flat\_order\_item |
| qty\_refunded | mapping from sales\_flat\_order\_item |
| qty\_shipped | mapping from sales\_flat\_order\_item |
| fulfill\_count | available number of products to fulfill |
| status | status of item queue.  STATUS\_PENDING = 1; // start from pending  STATUS\_PARTIAL = 2; // when item is partially filled  STATUS\_READY = 3; //when item is completely filled  STATUS\_PROCESSING = 4; // when all items in the related order are filled and order is being processed. For item in 'processing' should be 'locked', i.e. do not modify while it's still in processing  STATUS\_SUSPENDED = 5; // special flag for admin to suspend an item, this is a placeholder function for now  STATUS\_CLOSED = 6; // when order is closed  STATUS\_CANCELLED = 7; // when order has been cancelled |
| created\_at | Timestamp for creating data |
| Update\_at | Timestamp for updating data |

**Item Queue Functions**

Here are basic functions for Fulfillment Factory:

|  |  |
| --- | --- |
| **Save Item Queue** | |
| **Function Name** | saveFromOrder |
| **Scenario** | Every time when order is created or modified, we should save related Item Queues. |
| **Trigger** | Triggered by ‘order\_split\_after*’* event. |
| **Related code** | *Harapartners\_Fulfillmentfactory\_Model\_Observer*  *-> saveItemQueue* |
| **Note** | * Should consider error handling when order is not saved successfully. |

|  |  |
| --- | --- |
| **Cancel Item Queue** | |
| **Function Name** | cancelItemqueueByOrder |
| **Scenario** | Every time when order is cancelled, we should cancel related Item Queues. For now, we just mark these Item Queue objects’ statuses as ‘Cancelled’. |
| **Trigger** | Triggered by ‘*order\_cancel\_after’ event* |
| **Related code** | *Harapartners\_Fulfillmentfactory\_Model\_Observer*  *-> cancelItemQueue* |
| **Note** | * Should consider error handling when order is not cancelled successfully. |

|  |  |
| --- | --- |
| **Show Item Queue** | |
| **Function Name** | Show Item Queue |
| **Scenario** | Show Item Queue objects in Grid. You can manipulate these objects, such as search, modify, batch cancel etc. |
| **Trigger** | When user go to **Sales -> Fulfillment** factory at Admin Panel |
| **Related code** | *Harapartners\_Fulfillmentfactory\_Adminhtml\_ItemqueueController* |
| **Note** | * May need to implement manual Stock Update button in the future. |

|  |  |
| --- | --- |
| **Batch Cancel Item Queue** | |
| **Function Name** | batchCancel |
| **Scenario** | User can manually batch cancel many item queues. |
| **Trigger** | When user select item queues in Admin grid and click “Batch Cancel” Action. |
| **Related code** | *Harapartners\_Fulfillmentfactory\_Model\_Service\_Itemqueue*  *-> batchCancel* |
| **Note** | * Right now the cancellation logic is cancelling old order and creating an new order with updated items. * The cancellation logic will be changed in the future |

|  |  |
| --- | --- |
| **Stock Update** | |
| **Function Name** | stockUpdate |
| **Scenario** | When Totsy has inventory information from warehouse, they can update order items’ fulfillment detail. Order Item Queue objects by created date and start filling with available products count. |
| **Trigger** | Should be triggered after fetching inventory information from Dotcom |
| **Related code** | *Harapartners\_Fulfillmentfactory\_Model\_Service\_Itemqueue*  *-> stockUpdate* |
| **Note** | * Only consider item queues which are ‘Pending’ and ‘Partial’. * Earliest item queue should be filled first. * Could partial fill items * **Question**: should we keep products’ stock info after Stock Update? * **Question**: should we add manual stock update function? |

|  |  |
| --- | --- |
| **Update Order Fulfillment Status** | |
| **Function Name** | updateOrderFulfillStatus |
| **Scenario** | After system has updated inventory information for magento store, we can update orders’ statuses to determine whether this order could be fulfilled or not. |
| **Trigger** | Should be triggered after stock update function |
| **Related code** | *Harapartners\_Fulfillmentfactory\_Model\_Service\_Itemqueue*  *-> updateOrderFulfillStatus* |
| **Note** | * If All items in an order are ready, this order could be fulfill. * If only parts of items in an order are ready, it would trigger split this order to 2 orders, one could be fulfilled and the other one is still pending. |

**Dotcom API Functions**

Here are basic functions for Dotcom API:

|  |  |
| --- | --- |
| **Submit Products To Dotcom** | |
| **Function Name** | submitProductsToDotcom |
| **Scenario** | After products’ creation in Magento, we should sync products’ information to Dotcom. |
| **Trigger** | Every time when new products are created. |
| **Related code** | *Harapartners\_Fulfillmentfactory\_Model\_Service\_Dotcom*  *-> submitProductsToDotcom* |
| **Note** | * Schedule to run every day after launch. * Only consider ‘simple’ product for now. |

|  |  |
| --- | --- |
| **Update Inventory** | |
| **Function Name** | updateInventory |
| **Scenario** | Fetch latest inventory detail from Dotcom and update current stock in Magento system. |
| **Trigger** | When system needs to update stock detail, scheduled task or manually. |
| **Related code** | *Harapartners\_Fulfillmentfactory\_Model\_Service\_Dotcom*  *-> updateInventory* |
| **Note** | * Schedule to run every day after launch. * **Question**: should we submit products’ stock to Dotcom info after Stock Update? |

|  |  |
| --- | --- |
| **Submit orders to fulfill** | |
| **Function Name** | submitOrdersToFulfill |
| **Scenario** | Submit ‘Ready to ship’ orders to Dotcom for fulfillment. |
| **Trigger** | When orders need to be fulfilled. |
| **Related code** | *Harapartners\_Fulfillmentfactory\_Model\_Service\_Dotcom*  *-> submitOrdersToFulfill* |
| **Note** | * Schedule to run every day after launch. **TBD** * The orders’ status is ‘READY’. **TBD** |

|  |  |
| --- | --- |
| **Update Shipment** | |
| **Function Name** | updateShipment |
| **Scenario** | Fetch Dotcom shipment information and update to Magento system. |
| **Trigger** | Scheduled task |
| **Related code** | *Harapartners\_Fulfillmentfactory\_Model\_Service\_Dotcom*  *-> updateShipment* |
| **Note** | * Schedule to run every day after launch. **TBD** * Needs to specify query ‘from date’ and ‘to date’. |