API Requirements & User Stories

JOOR Integration Standard

We pride ourselves on top level technology standards. Because of this, we have an integration standard that all technology partners and brands must adhere to in order to be considered integrated with JOOR. The integration data flows are listed below:

- Orders
- Styles
- Customers
- Linesheets
- Images
- Inventory

The rest of this document consists of requirement details that can be used as a checklist as you build out an integration for each data flow. This is to ensure that you are able to integrate to our standard. We are happy to help you along the way.

API Requirements & User Stories Checklist

Data Flow	Type	Requirement	Description	User Story	Completed (Y/N)
General	Technical	Tech Point of Contact	Provide singular technical point of contact for integration related work, issues, and troubleshooting.	JOOR needs to work with a single technical contact that will be the ERP's project lead, liaising between JOOR and the ERP's engineering team.	
General	Functional	Provide JOOR access to ERP test environment	ERP to create a JOOR log-in for integrations@joor.com to access ERP test environment.	Enables the JOOR Integrations team to become familiar with ERP functions and understand how to troubleshoot between both systems.	
General	Functional	Provide documentation on the JOOR integration	ERP to provide JOOR with integration documentation on how to operate the integration functionalities within ERP.	JOOR needs to be able to advise brand clients on how to use the integration within the ERP.	
General	Technical	Sandbox Access	Provide the ability to test the integrations using JOOR's Sandbox Account.	All integrations should be tested on JOOR's Sandbox environment to avoid bad data in/interrupting usage of JOOR production.	
General	Technical	Provide Header/Body Call Details	Send JOOR the header (including endpoint URI) and body (XML/JSON) details of specific API requests to help troubleshoot issues.	When an integration fails, JOOR needs the API call details in order to fully troubleshoot.	
General	Functional	ERP Interface	Provide interface with integration settings on the ERP accessible by the brand.	I want to be able to control basic integration settings from within the ERP, without having to reach out to JOOR or the ERP to make changes. I.e. I want to be able to enter the API token credentials, provided by JOOR, into my ERP.	
General	Functional	ERP Interface	ERP to provide shipping methods, payment terms, warehouse name(s), and price type(s) to the JOOR integrations manager at the start of the brand integration project.	JOOR integration requires pre set-up of shipping methods, payment terms, warehouse name(s), and price type(s) before we can begin testing the brand's data flow from ERP to JOOR.	

General	Functional	API Token on ERP UI for Clients	Clients need to be able to insert their JOOR provided API token for both sandbox and production environments on ERP.	I want to be able to insert the API token on my ERP to connect to JOOR.	
General	Functional	Display API Errors for all integrations	Process and display REST API responses to users to allow for troubleshooting errors.	I want to see errors instantly so I can troubleshoot integration failures/bad data on my own.	
General	Functional	ERP Training	Provide training to brand on ERP integration process with JOOR.	I want to be familiar with how to use the integration on my ERP, and to be able to troubleshoot issues by referencing training documentation.	
General	Functional	Runs Automatically and Manually	Trigger all integrations automatically via a set schedule and manually on demand.	I want my integrations to run automatically, but also have the ability to run them whenever I need to.	
General	Functional	Split JOOR Accounts	Store multiple API tokens, if account is split to multiple on JOOR. Usually per division or company.	I want to utilize multiple accounts on JOOR, so that I can use separate JOOR accounts to manage my brands. This way, I can have different lines that may sell different styles to different customers.	
Orders	Technical	Validate Import before Marking as Exported	Import orders from JOOR, perform a valid import to ERP, and mark the order as “Exported” on JOOR using sync=1 parameter.	I want to validate the import into my ERP prior to marking the order as exported on JOOR, so the order won’t show as successfully exported on JOOR when it has failed to import.	
Orders	Functional	Store JOOR ID	Store JOOR’s Order ID for reference on the ERP.	I want to be able to quickly see which order on JOOR corresponds with the one on my ERP.	
Orders	Functional	Import Updates	Import the same order more than once to overwrite data on ERP.	I want to make changes to approved orders on JOOR and export those changes to my system.	
Orders	Functional	Import all JOOR Data	Import all style, customer, and order data from JOOR.	I want to receive all order information that I put on JOOR, so I don’t have to manually enter any fields.	

Orders	Functional	Import Order/Line Discounts	Import Order and Line Item level discounts	I may take order or line item level discounts on JOOR, and I want both to be reflected on the order on my ERP.	
Styles	Functional	Create/ Update/ Delete Styles	Create, update, and delete styles.	I want to fully manage my styles on my ERP, not having to manually add, update, or delete styles on JOOR.	
Styles	Functional	Multiple Price Types and Currencies	Support multiple price types and currencies for wholesale and retail prices.	I want to have multiple currencies on JOOR, with multiple price levels per currency, to reflect pricing that is different for different customers.	
Styles	Functional	Deactivate Sizes	Deactivate sizes	I may cancel production of a certain size for a style, and I need that size to be deactivated on JOOR so it can’t be purchased.	
Styles	Functional	Delete Colors	Delete Colors	I may cancel production of a certain color for a style, and I need that color to be removed from JOOR so it can’t be purchased.	
Styles	Functional	Customizable Style Identifiers	Split styles on JOOR by brand’s desired method (i.e. colors, fabrication, silhouette, etc.).	I want to choose which style characteristics determine a unique style on JOOR.	
Styles	Functional	Style Tags	Create and assign custom style fields (style tags).	I want to use custom tags on JOOR that will allow me to group styles via data that aren’t already sent through the integration.	
Styles	Functional	JOOR Categories	Send JOOR Categories when creating styles.	I want to assign my styles to categories, to provide more visibility and filtering options for both reps and retailers.	
Styles	Functional	Casepacks	Send casepack sizes.	I sell some styles in casepacks/pre-packs, and I want to be able to do so on JOOR.	
Styles/ Customers	Technical	ISO Currency Codes	Use standard ISO 3-character currency codes	JOOR requires the use of ISO currency codes.	
Styles/ Customers	Technical	Send Consistent Price Types	Send the same price type currencies and labels for styles and customers.	JOOR requires that customers and styles are sent with matching price type currencies and labels, so duplicate price types aren’t created.	

Customers	Technical	Bulk Connections Endpoint	Utilize the Bulk Connections endpoint to create and update customers.	JOOR requires the use of the bulk connections endpoint for all customer integrations. Other endpoints are being deprecated.	
Customers	Functional	Create Doors	Create Doors based on shipping addresses.	I want to place an order on JOOR for the same customer with multiple shipping locations - this is achieved via doors on JOOR.	
Customers	Functional	Discounts at Orders or Item	Assign Discounts at the Order or Item level.	On JOOR, I can take discounts on the whole order or per item, and I want to choose how this is defaulted via the integration.	
Customers	Functional	Warehouses	Assign warehouses to customers.	I want customers to be assigned to the correct warehouses so inventory for them will appear accurately on JOOR.	
Customers	Functional	Sales Reps	Assign Sales Reps to customers.	I want the default sales rep for a customer to be reflected on JOOR.	
Linesheets	Technical	Allow Manual Linesheets	Allow users to manually create linesheets on JOOR (using the delivery_overwrite field)	I want to be able to create linesheets (collections) on JOOR based on my own criteria/the styles I want to highlight.	
Linesheets	Functional	Customizable linesheets	Create linesheets on JOOR based on brand's criteria (season, delivery, etc.).	I want to decide what characteristics determine what styles go in a unique linesheet on JOOR.	
Linesheets	Functional	Color Exclusions	Exclude colors from linesheets per season.	I sell only certain colors of a style in each season, and I want to ensure that colors aren't on linesheets when they aren't available in a certain season.	
Linesheets	Functional	Drop from Linesheets	Drop styles from linesheets.	I may decide not to sell a style, and I don't want it to be available on any linesheets.	
Images	Technical	Send Images as public URLs	Send images as non-password protected URLs.	JOOR requires all images to be sent via URLs that are publicly accessible.	

Images	Functional	Send all Images	Send style, color, and swatch images to JOOR as desired by brand.	I want to send all of the images on my ERP, utilizing all three levels of images on JOOR.	
Inventory	Technical	Use Style ID or Identifier	Load inventory using Color, Size, and Style ID or Style Identifier if splitting styles by data other than Style Number.	If a brand has styles split on JOOR by something other than style number/color, inventory needs to be loaded via style ID or identifier.	
Inventory	Technical	Use UPC	Load inventory using UPC value..	As a brand I want to use the UPC value associated with my styles, sizes and colors, to load inventory values.	
Inventory	Functional	Overwrite or Update	Provide option for brand to send all inventory (Overwrite) or updates only (Update).	I want to update inventory information so that I can keep inventory information with the rest of my styles and only update relevant information. I want to overwrite inventory information so that I can keep up-to-date inventory information and mark other styles as made to order (MTO).	
Inventory	Functional	Multiple Warehouses and Dates	Support multiple warehouses and availability dates (Immediate and Future).	I use multiple warehouses and I want both immediate (ATS) and future (WIP) inventory to be shown on JOOR.	
Inventory	Functional	Casepacks	Send inventory for casepack sizes.	I sell styles in casepacks/pre-packs, so I need inventory to be shown per pack.	