Integrating With Coupa

Best Practices Approach

Coupa Integration Best Practices

- Process Driven Integration
- When to use Coupa REST APIs?
- When to use Coupa Flat File Integrations?
- How to parse Coupa REST APIs?
- How to interact with Coupa SFTP?
- How to parse Coupa Flat Files?

Coupa Integration Best Practices

- How to handle Integration Errors?
- Integration confirmation approaches
- Scheduling Data Retrieval from Coupa
- Orchestrating Coupa Services
- Notifying Remote Services
- Single Sign-On Integration
- Integration Test Strategy
- Performance Considerations
- Legacy Data Migrations

Process Driven Integration

- The ‘Right & Light’ Integration Approach
    - Business Process determines what Integrations are needed
    - Implement only the right integrations that are required and determined by the amount of data to be managed and the frequency of data change
    - Adopt the right integration method to support the business process
        - In the Procurement domain not every integration needs to be real time
    - Avoid business logic in integration layer
    - Recognize and avoid over-engineering

When to use / not use Coupa REST APIs?

- Use the Coupa REST API when:
  - You need to update data in Coupa near-real time
  - You have a need to orchestrate different service calls across multiple systems
  - You want sophisticated error handling
  - You need to make frequent API calls with small numbers of records (API should not be used for infrequent batch integrations for example)
- Do not use Coupa REST API when
  - Bulk import/exporting data
  - There is no business need to have near-real time data integration (e.g. nightly update of users from ERP to Coupa)
  - You do not have a middleware tool to simplify and monitor API-based integrations

When to use Coupa Flat File Integrations?

- You need an easy audit trail of data moving between systems
- You want a clearly defined hand-off between internal and external systems
- You want to simplify integrations as much as possible
- Bulk and/or batch integration routines
- Your IT team is more comfortable parsing data and doing data transformations with CSV flat files as opposed to using XML data and making REST API calls

How to parse Coupa REST APIs?

- Plan for change and assume our XML structures can change due to configuration and release upgrades
- Design mapping based on fields required for the specific integration and ignore irrelevant fields
- Build integrations that are configurable and loosely coupled
- Plan for paging as Coupa will only return a max of 50 records per API GET call
- Coupa will return any errors encountered from API calls but it is the responsibility of the calling routine to handle those errors. Coupa does not make API errors visible in the application or send any notifications.

How to interact with Coupa SFTP?

- Use public key authentication
- Post inbound files to Coupa only when ready to load as they will be processed within a few minutes of being received
- Orchestrate dependencies by properly spacing inbound files to Coupa
- Get outbound files (from Coupa) and then delete once successfully processed. Coupa archives all outbound files in each customer instance so there is no need to archive on the SFTP server.

How to parse Coupa Flat Files?

- Parse by column name, not position. This can be achieved by dynamically determining the position of each column at the start of every run or leveraging middleware parsers that handle this for you.
- Make integrations configurable (field names, formats, mappings) to easily adopt future changes, based on Coupa releases and configuration options
- Plan for special characters in text fields (commas, double-quotes, carriage returns)
- Coupa's flat files will always use the UTF-8 character set so configure your integration routines accordingly

How to handle Integration Errors?

- Assess the most common errors occurring when integration with your internal ERP system and determine how better design can avoid them
- For errors that might be unavoidable, assess who should be responsible for resolving
- Ensure users responsible for resolving errors are trained on how to monitor for required actions through custom views in Coupa - do not rely solely on email notifications
- Ensure error handling procedures are in place at every hand-off point between Coupa and your internal systems
- Understand that data errors on Coupa’s File Status page are the responsibility of the customer to monitor, manage, and resolve

Integration Confirmation Approaches

- When sending transactions from Coupa to an ERP system, it is best practice for a confirmation to be returned from the ERP system to Coupa indicating success/failure for end user visibility
    - Allows end users to take appropriate action and also helps to simplify integration logic when it comes to determine whether a transaction needs to be created or changed in the ERP system
- Confirmations from the ERP back to Coupa must be posted via API, there is no standard flat-file available
- The following approaches are available to make this data visible in Coupa:
    - Integration History/Integration Success/Integration Errors - an audit trail of integration attempts and results
    - Custom Fields - add custom text fields at the header level of transactions to store a reference number if successful or any error messages if not successful

When to use Coupa Integration History?

- You want to send notifications to specific users or groups of users through Coupa when transactions fail to post to your ERP system
- You want to trace the history of each transaction as it relates to posting to your ERP system (for failure rate analysis for example)
- You need to post a series of error messages returned from your ERP system in Coupa for a particular transaction (line level error messages for example)
- You want to post confirmation details on Receipts in Coupa (custom field option not available as receipts are not editable once created)

When to use Custom Fields for confirmation?

- You want end users or administrative users to be able to create simple custom views to monitor ERP posting errors
- You don’t need a notification to be sent for every transaction error
- You want to simplify your confirmation process as much as possible
- You want to leverage the confirmation details to drive query logic for other integrations

Scheduling Data Retrieval from Coupa

- Leverage Coupa's delivered "Exported" flag on transactions for easy determination of when a transaction is ready to be pulled from Coupa
- Ensure a process is in place for resetting the Exported flag to No when transactions are modified or voided
- Schedule transaction integrations frequently throughout each day to spread the volume more efficiently
- If you need near-real time integration, schedule your retrieval as often as every 5 minutes (beware of volume spikes which may drive processing time beyond 5 minutes)

Orchestrating Coupa Services

- Use Coupa API when there is a need to synchronously manage the transaction between Coupa and customer systems
- Leverage a middleware layer to orchestrate bi-directional integration between systems by calling both Coupa and customer system APIs to ensure data updates are done consistently across systems and the appropriate confirmations are provided to each system
- Ensure any technical and/or data errors are handled appropriately and provide immediate awareness to the appropriate customer resources

Notifying Remote Services

- Coupa provides the ability to notify a remote REST-based service and pass Requisition/Invoice XML data as part of the transaction approval process in Coupa
- One-way invocation service model to Remote Service (no response processing)
- Remote Service responsible for calling Coupa APIs to update transactions (if needed) and approve/reject the transaction
- Remote Service and its availability is the responsibility of the customer
- If Remote Service is not available, the notification will be retried on the same interval as reminder notifications for user-based approvals

Single Sign-On Integration

- Single Sign-On (SSO) is recommended for all Coupa Production environments and is supported in non-Production environments as well if desired.
- Details on the methods Coupa supports as well as configuration details can be found HERE.

Integration Test Strategy

- Document and execute detailed integration test scenarios based on customer-specific business process and data flows
- Ensure both positive and negative testing scenarios are captured
- Automate as much as possible so the test scenarios are easily repeatable
- Plan to reuse and evolve integration test scenarios to easily assess the impact of future Coupa and customer system updates

Performance Considerations

- Coupa’s platform is a highly scalable, multi-tenant platform and efficiency is of critical importance, especially when it comes to Integrations
  - Bulk data loads must be done through Flat File integration only, APIs are not designed for high-volume data import/export
  - Large flat files should be split to allow multiple files to be processed at the same time
  - Ongoing reference data loads must be incremental for larger customers with high volumes
  - Data volumes must be understood and made know to Coupa very early in the implementation process (usually during the initial kickoff)
  - Plan for performance early in the project for success

Legacy Data Migrations

- Best practice is to make every effort to close out open transactions in your current system as part of any migration to a new P2P system, including Coupa
- For those instances where legacy data migration is required, it is critical that this be planned for from the outset of the project due to the time and complexities involved in merging legacy data modeled for legacy processes into a new process and integration model
- Limit the volume of data that needs to be migrated as much as possible and ensure both your Integration and User testing plans account for testing all actions that may need to be taken on legacy data in Coupa