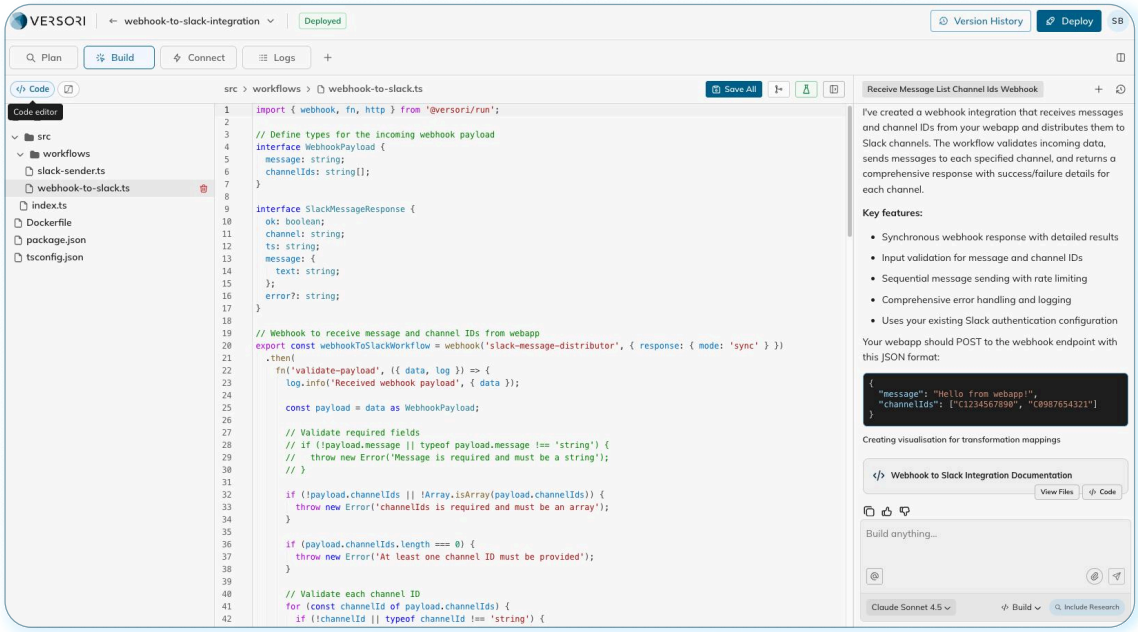
VERSORI

Versori.com

Welcome to Your

Prompting *Guide*



VERSORI

Hey Versori Users 👋

Use this template to define the requirements for an integration workflow. Replace the placeholder values with the details relevant to your project.

Providing structured information helps the Versori Plan Agent understand the systems involved, how data should flow between them, and what outcome the integration should achieve.

Minimum Information Required:

- The systems involved
- The primary data entity (e.g. orders, products, customers)
- A high-level description of the integration workflow

This gives the agent enough context to understand which systems it should interact with and what outcome the integration should achieve.

You can also include additional detail if needed, such as specific endpoints, authentication requirements, or data transformation rules.

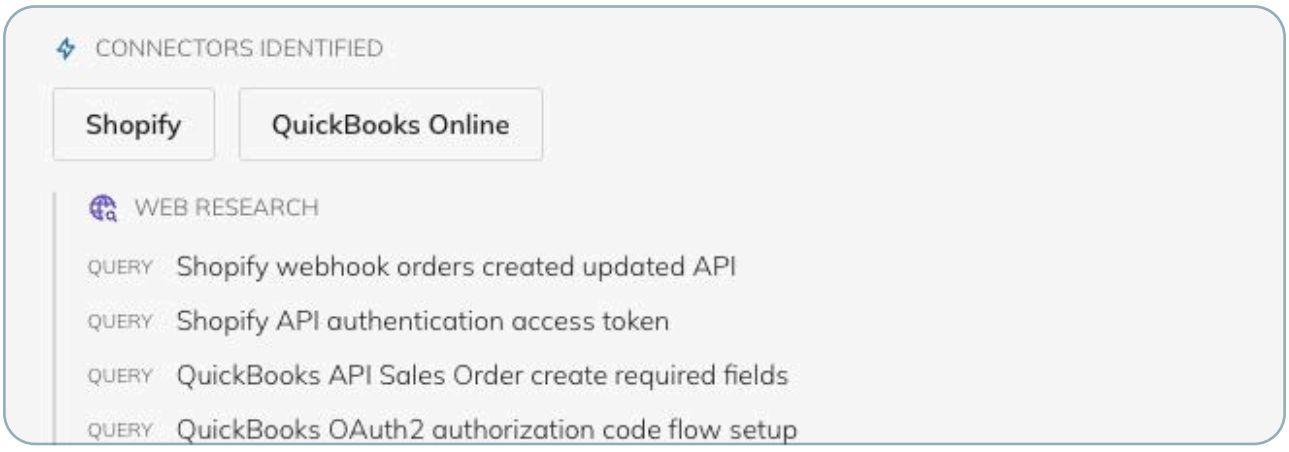
Example

Systems

- Shopify — https://shopify.dev/docs/api
- NetSuite — https://docs.oracle.com/en/cloud/saas/netsuite/

High-level workflow

When a new order is created in Shopify, the integration should retrieve the order details, including customer information, line items, pricing, and shipping data.
This data should then be transformed to match the NetSuite Sales Order schema and sent to NetSuite to create a corresponding Sales Order record.



VERSORI

Integration Prompt Guide

Use this template to define the requirements for an integration workflow. Replace the placeholder values with the details relevant to your project.

Customer / Project

Customer Name: `<Customer Name>`
Project Type: `<POC / Production / Internal>`
Number of Connectors: `<Number of systems involved>`

Integration Overview

Provide a short description of the integration and what it is intended to achieve.
Example structure:

- Primary Data Entity: `<Orders / Products / Customers / Inventory>`
- Source System: `<System where the data originates>`
- Destination System: `<System receiving the data>`
- Direction of Data Flow: `Unidirectional / Bidirectional`
- Source of Truth: `<Which system owns the canonical data>`

Example:
This integration builds a workflow that pushes product data from Shopify to NetSuite, with Shopify acting as the source of truth.

Systems

List the systems involved in the integration and how they will be accessed.

System 1
System Name: `<System Name>`
Access Method: `<API / Connector / SDK>`
Documentation URL: `<URL if available>`
If documentation is not publicly available, provide the API request details below.
Example API Request

```
curl --location '<API Endpoint>' \
--header '<Header Key>: <Header Value>' \
--header '<Header Key>: <Header Value>'
```

Authentication

Describe authentication requirements if applicable.
Example:

- No authentication required
- API key
- OAuth
- Custom headers

Expected Data Volume

Provide an estimate of the number of records processed by the integration.
Examples:

- 50 orders per day
- 5,000 products
- 1 million records per sync

This helps determine batching, pagination and processing strategies

Workflow Trigger

Describe how the integration should start.
Examples:

- Cron schedule
- Webhook event
- Manual trigger

Pagination (If Applicable)

Describe how pagination works for the source API.
Pagination Parameter: <Parameter Name>
Pagination Logic: Explain how the total number of pages should be calculated.
Example fields returned in the API response:

```
<response.path.totalCount>
<response.path.pageSize>
```

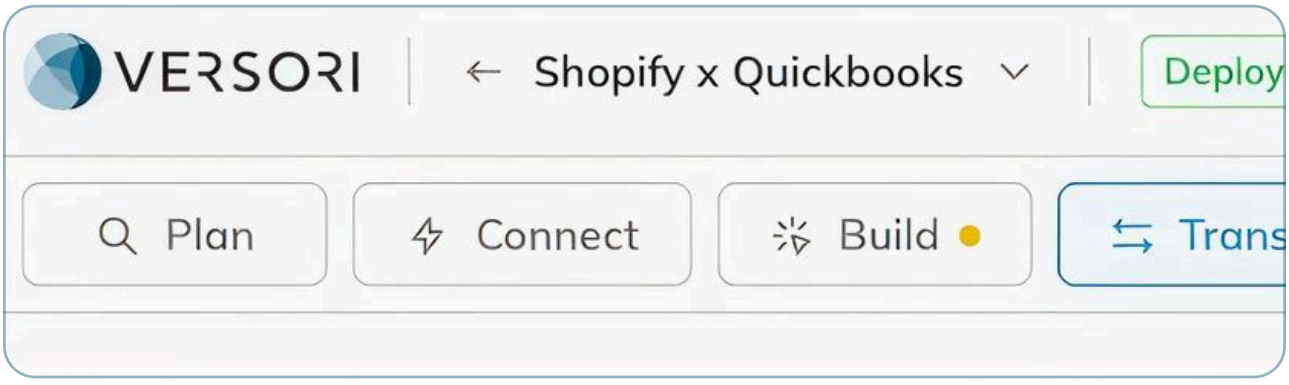
Example Response:

```
{"metadata": {
    "totalCount": 75,
    "pageSize": 60,
    "pageNumberParameter": "p"}
```

The integration should iterate through pages using the pagination parameter until all records have been retrieved.

Data Extraction

Describe how the integration should extract records from the source system.
Example:
Products should be read from the API response, filtered using the provided criteria, and processed sequentially for transformation and delivery to the destination system.



VERSORI

Business Logic

Describe the step-by-step logic the integration should perform.
Example structure:

1. Fetch records from `<Source System>` using the appropriate API endpoint and pagination logic
2. Filter the response to include only relevant entities (e.g. `type === "product"`)
3. Map the retrieved data to the `<Destination System>` schema or file format
4. Generate the required payload format (e.g. JSON, CSV, XML)
5. Send or upload the transformed data to `<Destination System>`
6. Monitor or poll the processing status if asynchronous imports are used
7. Log or generate a summary of the import results

Workflow Summary

Provide a concise summary of the workflow logic.
Example:

1. Retrieve records from `<Source System>`
2. Handle pagination if required
3. Extract the relevant entities (e.g. products, orders)
4. Transform the data into the `<Destination System>` schema
5. Send the transformed data to `<Destination System>`

Final Result

Describe the expected outcome of the workflow once it completes successfully.
Example:
All records retrieved from `<Source System>` are created or updated in `<Destination System>.`
The workflow should produce a summary log containing counts of successful records, failed records, and warnings to confirm the results of the import.

User Stories

User stories provide additional business context and help the agent understand why the integration exists and what success looks like.
Use the format:
As a `<role>`, I want `<outcome>` so that `<business value>.`
Example:

- As a business operator, I want data from `<Source System>` automatically synchronised with `<Destination System>` so that our systems stay up to date without manual work.
- As a technical user, I want workflow logs or summaries so that errors can be identified and resolved quickly.
- As a stakeholder, I want the integration initially scoped to a specific dataset or category so that the approach can be validated before scaling.