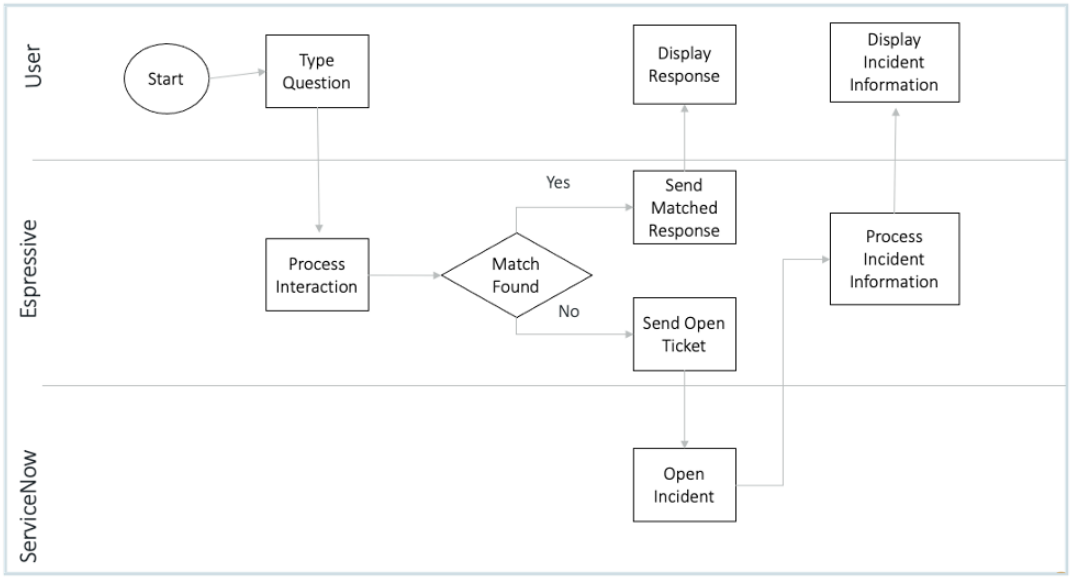
Espressive Barista and ServiceNow Integration Design Document

Table of Contents

Overview

Espressive Barista works with ServiceNow right out of the box and integrates with your existing portal and service catalog. Barista requires no change to how your help desk uses ServiceNow today and will automatically leverage your catalog items, record producers, workflows, variables, and service requests. Integration is bi-directional to an incident, service request, change request, and support connect. When Barista opens a ticket on behalf of an employee, Barista uses machine learning (ML) to learn from your historical ServiceNow data provided through a CSV file and predicts fields such as assignment group, category, and business service.

Process Flow – User



Update Set Install

The most important configurations made on ServiceNow within the update set install includes the following sections in the Espressive Integration.

Integration Settings

All outbound credentials required by the Espressive tenant for access to external systems, such as ServiceNow, are strong, encrypted, and stored in the database. These credentials are encrypted using a Data Encryption Key (DEK) that is unique to the tenant. For more information, see our Data Security chapter.

The user credentials to connect with Barista are:

- Esp.api.username
- Esp.api.password
- Esp.base_url (tenant.espressive.com)Integration Settings

Script Includes

Several libraries, used by various functions required for Espressive and ServiceNow integration, are available as Script Includes. These are used by the Processors, Scripted REST APIs, UI Actions, Business Rules, etc., and are provided by Espressive in its ServiceNow update set. Some of the essential libraries are as follows:

Name of Library	Description
SyncDataScript	Primarily contains utility functions used by Business Rules that need to sync up various objects from ServiceNow instance to the Espressive tenant, such as incidents, service requests, comments, attachments, etc.
EspLogging	Contains Espressive specific logging utilities for debugging purposes.
EspImpersonator	Defines functions that help the API user to impersonate Espressive tenant end-user when syncing is needed for objects from the Espressive tenant to the ServiceNow instance. For example, an Espressive tenant user comment on a task channel on the Barista Application needs to be synced at the corresponding incident at ServiceNow instance, and the comment has to be attributed to the actual tenant user and not the API invoking user (esp.integration).
EspIntegrationHub	Checks the integration health between Espressive and ServiceNow.
EspRecordHelper	Stores the various calls to get a record, a list, and a count.
EspTaskHelper	Retrieves a list of tasks for use with Barista. The tables Barista accesses are Task, Incident, and Request.

Any data exposed to the Espressive integration user (esp.integration) via the Access Control List (ACL) or the roles is potentially readable by Barista. To create a restriction, refer to Access Control inside the System Security configurations. The script includes don't require any additional roles or privileges other than the ones provided for the overall ServiceNow integration.

Business Rules

We use ServiceNow business rules to automatically synchronize relevant fields with the Espressive tenant as well as change values in the ServiceNow form fields when the specified conditions are met. In the following table, the term "ESP" is used to refer to Espressive tenant application.

Note: Espressive Business Rules and UI Actions will appear in the Toggles UI Page after installing Espressive update sets.

UI Action	Table	Description
1. Add variables and sets with category	Default Workflow Map [u_default_workflow_map]	When there is an update on the workflow at ServiceNow, it will also reflect in Barista.
2. Push to ESP when user is deleted 3. Push to ESP when user is changed	User [sys_user]	When a user is deleted or changed, it will also be reflected in Barista.
4. Copy attachment from req to all items 5. Push req attachment to ESP 6. Push to ESP on creation 7. Copy req item attachment to parent.items	Attachment [sys_attachment]	When an attachment is added to a request, it will also reflect in Barista.
8. Push to ESP when HR case is created 9. Assignment rule sync for Espressive 10. Push to ESP when HR case is changed	HR Case [sn_hr_core_case]	When an HR case is created, changed, or assigned, it will also reflect in Barista.
11. Push to ESP when request item complete 12. Copy req item comment to other items 13. Copy req item comment to parent req	Requested Item [sc_req_item]	When an item request is completed, or a comment is added, it will also be reflected in Barista.
14. Push to ESP when req is updated (comment) 15. Push to ESP when request is complete 16. Push to ESP when req is created on ServiceNow 17. Copy comment to all items	Request [sc_request]	When a request is created, updated, or completed, it will also reflect in Barista.
18. Push to ESP when incident is created 19. Assignment rule sync for Espressive 20. Update incident post deflection from ESP 21. Push to ESP when incident is changed	Incident [incident]	When an incident is created, changed, deflected, or assigned, it will also reflect in Barista.
22. Push to ESP when chat queue is changed	Chat Queue Entry [chat_queue_entry]	Allows bilateral communication between the chat queue of ServiceNow and Barista
23. Push to ESP when a change request is changed	Change Request [change_request]	When a Change Request changes, it will also reflect in Barista.

UI Actions

The following UI actions are configurations to add elements on some of the ServiceNow tables to push the required information to Barista:

UI Action	Table	Description
1. Teach Barista 2. Mark as outage	Incident [incident]	This enables the 'Teach Barista' function, setting a response/path for common cases This helps mark an incident as an outage
3. Publish as Barista FAQ	Knowledge [kb_knowledge]	Allows a published ServiceNow knowledge article to be used as a Barista FAQ.
4. Import Service Request to ESP 5. Import Product to ESP 6. Import Super Product to ESP	Catalog Item [sc_cat_item]	These actions help sync template definitions of Service Requests, Products, and Super Products to the Espressive tenant so that Service Requests or Catalog Requests can be created based off them on behalf of an end user.

Processors

Processors provide a customizable URL endpoint that can execute arbitrary server-side JavaScript code and produce output. We use these utility methods for Espressive to add common tasks on the ServiceNow side. Following table lists the processors added:

Processor Name	Path
Add Comment	esp_add_comment
Add Live Comment	esp_add_live_comment
Create Connect Support	esp_create_connect_support
Get Espressive Version	esp_get_version
Get Reference Records	esp_search
Get Reference Records for Choices	esp_search_sys_choice
Get Req Items Info	esp_get_req_items_info
Get Username from ServiceNow Instance	esp_get_snow_username
Import Group Member	esp_import_group_member
Nudge	esp_nudge
Publish Attachment	esp_publish_attachment
Remove ESP	esp_remove

Scripted REST APIs

The following table shows the Scripted REST API endpoints that Espressive application uses for various operations on an integrated ServiceNow instance:

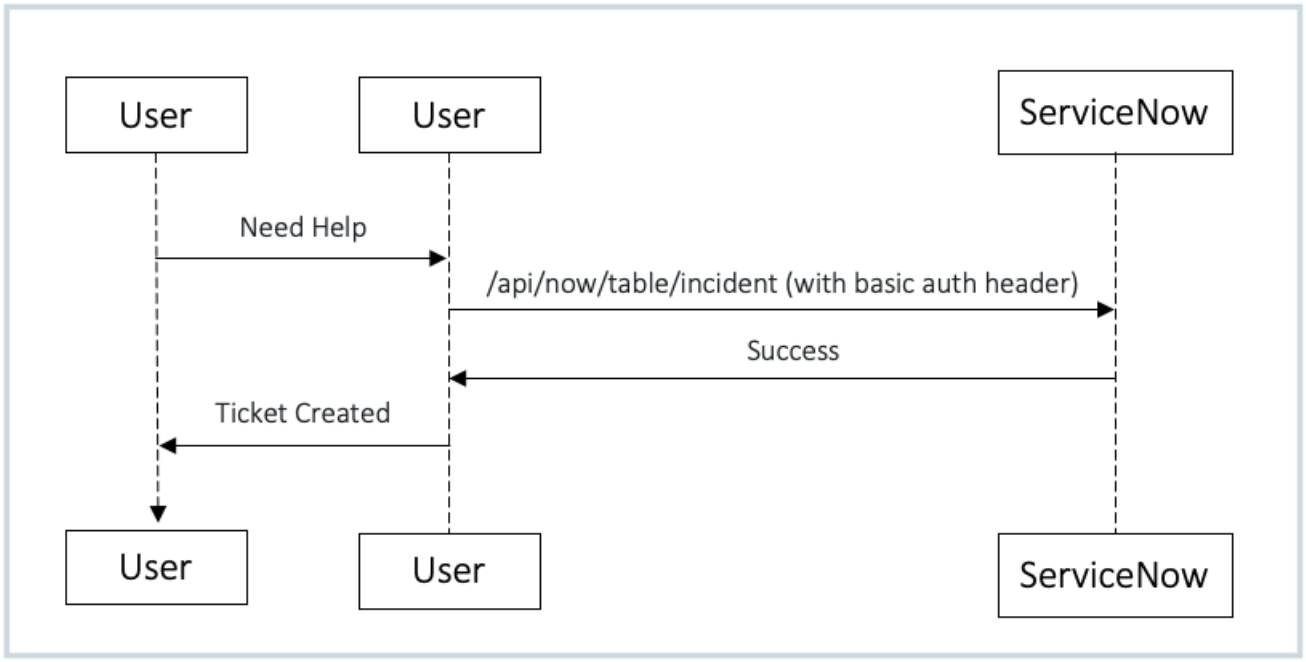
Scripted REST API Name	API ID	Description
Espressive CMDB Integration	esp_cmdb_demo	Contains an endpoint to get (business services) cmdb_ci_services info and their outage status (planned, degraded, etc.)
Espressive Generic Record Fetcher	esprecordfetch	Collection of endpoints to count, fetch, list records from a table in a generic way.
Espressive HR Integration	esp_hr	Endpoints that operate on sn_hr_core table to add comments or nudge an HR case.
Espressive Knowledge Search	espkbsearch	Endpoint that allows to search knowledge base by keywords.
Espressive Task Integration	esptaskdemo	Collection of endpoints that can fetch a given task (Incident, Service Request, Request Item) or get its status or even to set some fields on the task.

Note: In addition to these custom Scripted REST APIs, Espressive application uses ServiceNow's own API endpoints such as the following for various operations, including Espressive to ServiceNow synchronizations and to operate on any ServiceNow table in general that is permissible for the Espressive Integration user.

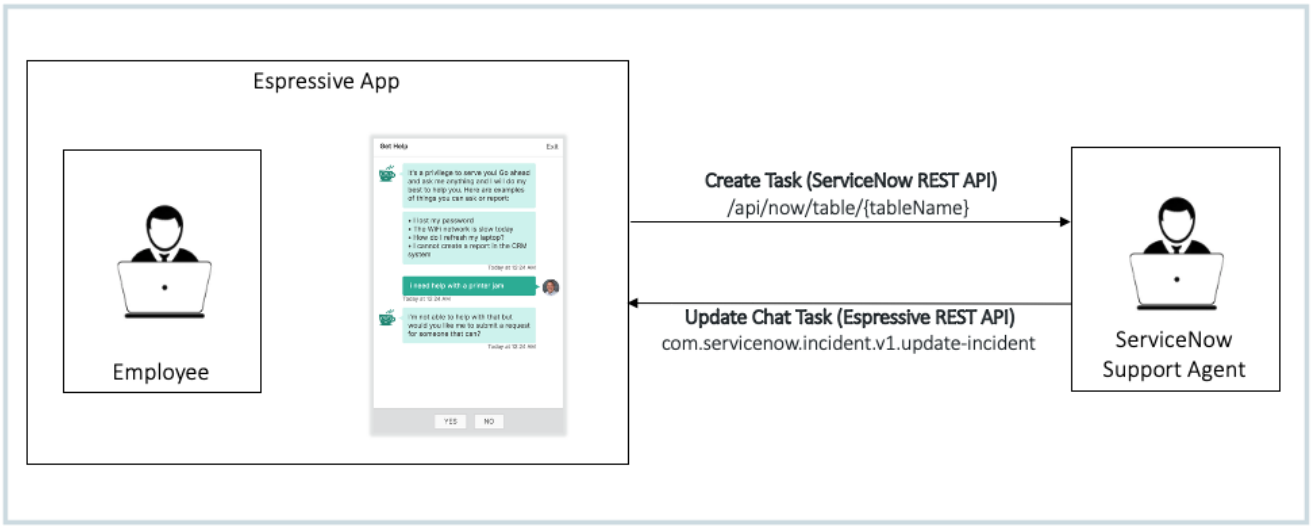
- /api/now/v1/table/sys_user
- /api/now/v1/table/incident
- /api/now/v1/attachment/
- /api/now/table/cmn_location

ServiceNow Updating Espressive API Sequence

Espressive Creating ServiceNow Incident API Sequence



Barista AI – Route to Agent



Data Security

As previously mentioned, all outbound credentials are encrypted using a Data Encryption Key (DEK) that is unique to the tenant. AES 256 cipher with GCM block cipher mode is used for these credentials' encryption. They are only decrypted for use by the application when needed.

The DEK itself is stored encrypted in the Espressive tenant database. The DEK is encrypted by a Master Key (Key Encryption Key) using AES 256 and GCM mode in AWS Key Management Service. Also, the entire database instance used for the Espressive tenant application data is itself AES 256 encrypted.

- In transit: HTTPS TLS 1.2
- At rest: Encrypted using AES256

Authentication

- For Users: SSO
- For Espressive to ServiceNow: Local ServiceNow user credentials configured on Espressive.
- For ServiceNow to Espressive: Local Espressive user credentials configured on ServiceNow.

Authorization

- Admin defined Roles and Permissions.
- Few out of box role defined, e.g. ADMIN, FAQ ADMIN, ANNOUNCER, etc.
- Permission model based on role/department/location.

Audit Trails and Intrusion Detection

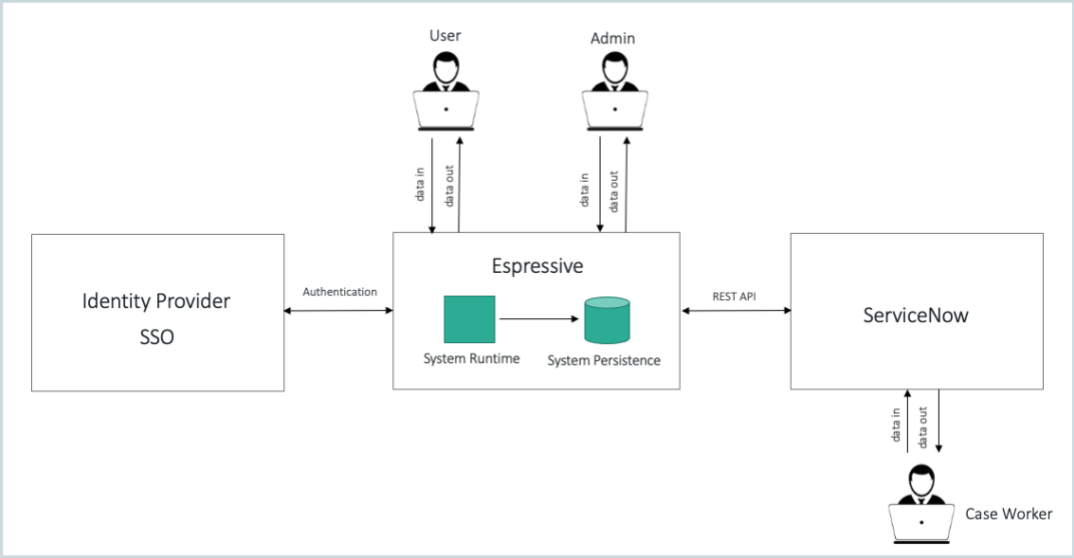
As we have an API tracking mechanism where APIs that add/edit/delete data in our database via HTTP methods such as POST, PUT, PATCH, and DELETE are tracked and recorded in our system.

The tracking API allows a privileged user with the right set of permissions to search for API usages using several filters such as user email, remote address, API method used, query parameter applied, etc. Additionally, this user can search for these records within a period defined by the start and end date.

Data Privacy

Privacy Policy: https://privacy.espressive.com/Privacy-Barista.pdf

System Context



Catalog Item and Service Request Import from ServiceNow Catalog

Barista is capable of ingesting catalog items from ServiceNow. When Barista was created, one of the problems we wanted to solve was to avoid overloading the catalog with products and services, because as consumers, people do not think of going shopping for services.

In Barista we have the concept of “products” and “services requests.” Think of a product as anything you would typically find on Amazon: hardware, accessories, software. Service requests would be a service IT delivers to the user, such as a password reset, creating a new VM, changing permissions on a security badge, etc.

From the “maintain items” (sc_cat_item) section of ServiceNow, you can import items by choosing the item and selecting the hamburger menu. From here, you will have a number of options to import into Barista.

Import Products

Suppose you have a product, such as a laptop, monitor, or keyboard. You can import the product to Barista, and these products will automatically show up in the Storefront Designer (in the Barista Admin menu). Any variables or questions set up for the product will automatically show up in the product on the Barista side.

When a product is published directly from the Storefront Designer, it will create associated items on the ServiceNow side as well. This allows you to simply pick from the Barista Reference Catalog (BRC) and eliminate the complexity of creating catalog items in ServiceNow. For this to work, it is necessary to first import a similar product from ServiceNow with the workflow and category set to the desired state to teach Barista about the ServiceNow configuration for such items. Once this import has been done, you can define the default workflow and category for items in the ServiceNow integration area.

If you wish to use a different workflow for a specific type of device (e.g., the Dell monitor does not use the standard monitor workflow), this can be updated in the ServiceNow catalog item.

Import Service Requests

Service Request items can be imported from ServiceNow (sc_cat_item) into Barista. Barista then learns the request's language and exposes the service to your employees. Barista understands all of the questions and variables associated with every service request and will collect the necessary information before automatically submitting requests on behalf of employees in ServiceNow.

Unlike a product displayed in the Barista shop, when import of a Service Request item is done from ServiceNow into Barista, an intent is created in the "Service Request" Barista application for use by the AI engine.

Knowledgebase Ingestion

Barista is capable of automatically identifying knowledge base (KB) articles within ServiceNow. Our AI-based KB Ingestion feature uses Machine Learning to quickly decipher all of your knowledge articles in bulk and leverage your existing content to increase ticket deflection. With KB ingestion, when Barista does not have a response to a topic, or the employee states the response is not helpful, Barista can display any KBs related to the topic.

As your content team makes changes to knowledge articles in ServiceNow, those changes are automatically re-ingested once approved so that Barista results are always reflective of the current knowledge base without requiring any additional task from the Espressive Admin.