nuvei Payment Middleware

Integrator Quickstart Guide

Examine each workflow part below to see the required steps, review process diagrams, and jump directly to the relevant operations in the API Explorer.

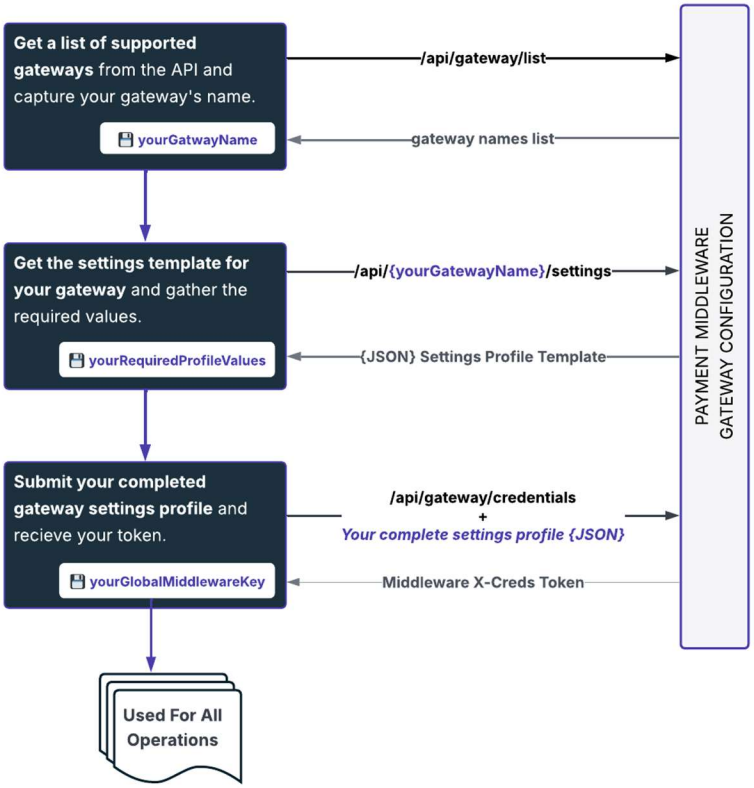
> 💡 **Pro Tip:**
>
> When using the live **API Explorer**, selecting a gateway from the dropdown will guide you through the required steps to generate a live x-credentials token. This token will then be automatically saved in Explorer Settings allowing you to test operations and view code examples tailored to your use case.

Part 1: Settings Profile & Middleware Token

Your journey begins by exchanging gateway credentials for a middleware token. This token (x-credentials) authenticates all subsequent requests for that specific gateway.

1. Call GET /api/gateway/list to find the exact system name for your gateway (e.g., "PayaConnect").
2. Fetch the settings template with GET /api/{gateway_name}/settings. This shows the exact fields required to build your settings profile in the next step.
3. Submit your completed settings profile to POST /api/gateway/credentials.
4. Securely store the returned x-credentials token. **Note:** When you run this operation in the API Explorer, the token is automatically saved as a variable for use in subsequent test calls.

Workflow Diagram:



Note: This is the fundamental first step: exchange your gateway settings for a middleware token that you will use globally.

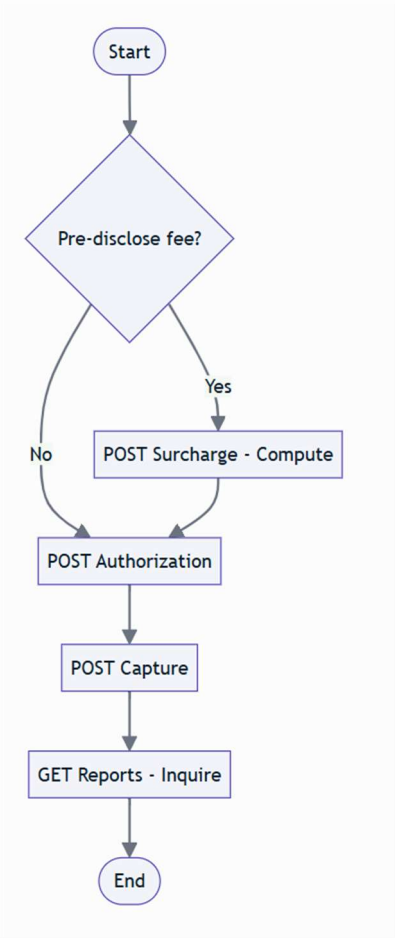
Part 2: CC Payment with Surcharge (InterPayments)

This workflow demonstrates a common credit card transaction that includes an automatically calculated surcharge via InterPayments.

1. Obtain your x-credentials token for your gateway (PayaConnect in this example), ensuring the InterPayments fields are included in your settings profile.
2. (Optional) Pre-calculate the fee for customer disclosure by using POST /api/surcharge/compute.
3. Perform an Authorization with the subtotalAmount set and surchargeAmount set to null. The middleware will compute and add the surcharge.
4. Finalize the transaction by calling Capture on the successful authorization.

5. Use Reports - Inquire to retrieve the final transaction status and confirm the persisted surchargeAmount.

Workflow Diagram:



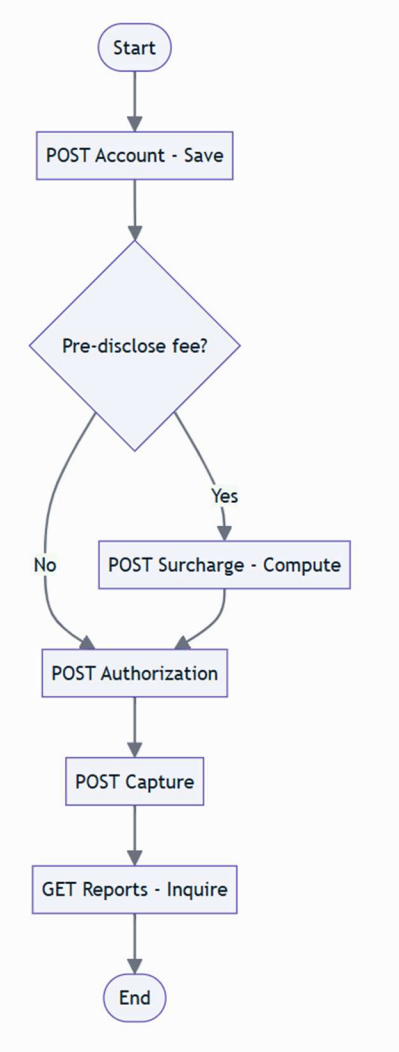
Note: To trigger surcharge calculation, send subtotalAmount and set surchargeAmount: null in your Authorization request.

Part 3: Token-on-File CC Server-to-Server Auth/Capture

This workflow shows how to save a customer's payment method and then use the resulting token for subsequent server-to-server transactions without requiring UI.

1. Create a vaulted record using Account - Save. Ensure you include the postal/ZIP code for accurate surcharge calculations.
2. (Optional) Pre-compute a surcharge using Surcharge - Compute, passing the profileId and accountId.

3. Initiate a transaction using the token with the Authorization endpoint.
4. Capture the successful authorization.
5. Confirm final status and amounts by calling Reports - Inquire.



Note: The postal/ZIP code stored on the profile impacts surcharge outcomes for token-based payments.

Part 4: APM (Bank Transfer) via Hosted IFrame

Alternative Payment Methods (APMs) like SEPA bank transfers are often asynchronous and require a hosted UI and webhook notifications.

1. Launch the hosted UI via POST ProcessTransaction, specifying SEPA as the payment method. The middleware will return a URL to embed in an iframe.
2. The user completes the payment flow within the hosted UI.
3. Your system listens for an asynchronous webhook notification confirming the transaction outcome.
4. Upon receiving the webhook, call GET Reports - Inquire to get the final, persisted transaction details from the middleware.

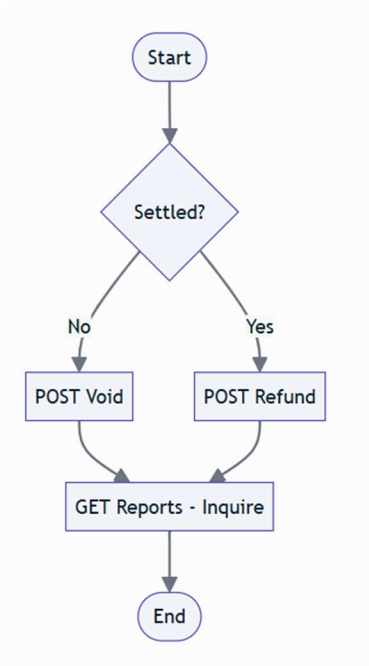
Note: The SEPA flow remains in the same popup. Some other APMs may open new window

---

Part 5: Post-Transaction Void/Refund & Reporting

After a transaction is complete, you may need to reverse it. The correct action depends on whether the transaction has been settled.

1. If the transaction has **not yet settled**, reverse it using the appropriate Void endpoint (e.g., PayaConnect Void or NCP Void).
2. If the transaction **has already settled**, you must issue a Refund (e.g., PayaConnect)
3. In either case, use the Inquire endpoint (e.g., PayaConnect Inquire or NCP Inquire) to confirm the final status and amounts.



Part 6: Understanding API Responses & Error Handling

The middleware uses a two-layer response system. Your code should first check for middleware-level errors before parsing the gateway's response.

Layer 1: Middleware Validation Errors

These errors occur if your request fails initial validation before it can be sent to the gateway.

- **401 Unauthorized:** Your x-credentials token is missing or invalid.
- **422 Unprocessable Entity:** Your request is valid JSON, but a required field is missing or a value has the wrong format. The response body will detail the specific error.
- **400 Bad Request:** Your request body is not well-formed JSON.

Layer 2: Gateway Responses

If your request passes middleware validation, it is sent to the gateway. The middleware passes the gateway's response back to you transparently.

- **200 OK:** The gateway successfully processed the request.
- **402 Payment Required:** The gateway rejected the transaction. The response body will contain the gateway's specific error message (e.g., "Declined," "Invalid Card Number").