

User Scenarios — Origo Cloud Events Arionbanki

Publisher: Origo **Version:** 1.0.0.0 **Submission Date:** 2026-07-07 **Test Environment:** Requires an Icelandic BC sandbox with "Origo Cloud Events Core" installed. The extension connects to Arion Banki's API. See "Test Credentials" section below.

Test Credentials

This extension connects to the Arion Banki banking API:

Service	Purpose	Test Environment
Arion Banki API	Bank statements, claims, payments, accounts, credit cards, foreign payments	Arion Banki sandbox API — requires test API key and client credentials

For Arion Banki: Provide test API credentials in the submission. The Arion Banki sandbox environment accepts test requests without real financial consequences.

Scenario 1: Extension Installation and Setup

Area: Installation & Activation

Setup

1. Start with a clean BC sandbox with IS Core localization installed (Cronus IS company)
2. Install the "Origo Cloud Events Core" extension (dependency)
3. Install the "Origo Cloud Events Arionbanki" extension

Steps

1. Search for "Cloud Events Setup" in the BC search bar
2. Verify the setup page opens without error
3. Navigate to the Arion banki section on the setup page
4. Verify the Arion Banki API credential fields are visible
5. Search for "Help.Arionbanki.Get" — verify it appears in the message type list

Expected Results

- The extension installs without error
- The Cloud Events Setup page shows the Arion banki configuration section
- API credential fields (client ID, secret) are available for configuration
- The Help.Arionbanki.Get message type is registered

Scenario 2: Configure Arion Banki Credentials

Area: Configuration

Setup

1. Complete Scenario 1 (extension installed)
2. Have Arion Banki test API credentials ready

Steps

1. Open the Cloud Events Setup page
2. Navigate to the Arion banki section
3. Click "Enter Arion banki Secret" to open the secret entry modal
4. Enter the test API client ID and secret
5. Save the configuration

Expected Results

- The secret entry modal opens and accepts credentials
 - Credentials are stored securely (not visible in plain text after entry)
 - The setup page confirms the credentials are configured
-

Scenario 3: Help Overview

Area: Core Functionality

Setup

1. Complete Scenario 2 (credentials configured)

Steps

1. POST a Cloud Event message via the Queue API:
 - `type : Help.Arionbanki.Get`
 - `sendContent : {}`
2. Process the task via the Task API
3. Retrieve the response from the Data API

Expected Results

- The response contains a Markdown-formatted help document
 - The document lists all Arion Banki message types grouped by domain (statements, claims, payments, accounts, foreign payments, credit cards)
 - Each message type includes a description and usage example
-

Scenario 4: Bank Statement Retrieval

Area: Bank Statements

Setup

1. Complete Scenario 2 (credentials configured)
2. Ensure at least one bank account exists in the test environment

Steps

1. POST a Cloud Event message via the Queue API:
 - `type : Arionbanki.Statement.Get`
 - `sendContent : {"accountId": "<test-account-id>", "dateFrom": "2026-01-01", "dateTo": "2026-01-31"}`
2. Process the task via the Task API
3. Retrieve the response from the Data API

Expected Results

- The response contains bank statement data in JSON format
 - Statement includes transactions with dates, amounts, and descriptions
 - The "Select Start Date" page modal is available for interactive statement import
 - The "Statement Import Summary" page shows reconciliation results after import
-

Scenario 5: Claim Query

Area: Claims Management

Setup

1. Complete Scenario 2 (credentials configured)

Steps

1. POST a Cloud Event message via the Queue API:
 - `type : Arionbanki.Claim.Query`
 - `sendContent : {"claimantKennitala": "<test-kennitala>", "dateFrom": "2026-01-01", "dateTo": "2026-06-30"}`
2. Process the task via the Task API
3. Retrieve the response from the Data API
4. POST a second message:
 - `type : Arionbanki.Claim.QueryOne`
 - `sendContent : {"claimId": "<claim-id-from-step-3>"}`
5. Process and retrieve

Expected Results

- Claim.Query returns a list of claims matching the filter criteria
 - Claim.QueryOne returns detailed information for a single claim
 - Results include claim status, amounts, due dates, and payer information
-

Scenario 6: Claim Batch Creation and Cancellation

Area: Claims Management

Setup

1. Complete Scenario 2 (credentials configured)
2. Configure a claim account via the "Arion Claim Account" page

Steps

1. POST a Cloud Event message via the Queue API:
 - `type : Arionbanki.Claim.CreateBatch`
 - `sendContent : {"claims": [{"payerKennitala": "<test-kennitala>", "amount": 10000, "dueDate": "2026-08-01", "reference": "TEST-001"}]}`
2. Process the task via the Task API
3. Retrieve the response (batch ID and operation status)
4. POST a follow-up message:
 - `type : Arionbanki.Claim.GetOperationResult`

- `sendContent : {"operationId": "<operation-id-from-step-3>"}`

5. Process and verify the batch was accepted

6. POST a cancellation:

- `type : Arionbanki.Claim.CancelBatch`
- `sendContent : {"batchId": "<batch-id-from-step-3>"}`

7. Process and verify cancellation

Expected Results

- CreateBatch returns a batch ID and async operation ID
 - GetOperationResult confirms the batch was processed successfully
 - CancelBatch cancels the test batch without error
 - Claim accounts are manageable via the Arion Claim Account page
-

Scenario 7: Payment Batch Submission and Results

Area: Payments

Setup

1. Complete Scenario 2 (credentials configured)
2. Ensure test bank accounts are configured

Steps

1. POST a Cloud Event message via the Queue API:
 - `type : Arionbanki.Payment.Batch`
 - `sendContent : {"payments": [{"recipientKennitala": "<test-kennitala>", "recipientAccount": "<test-account>", "amount": 5000, "reference": "PAY-001"}]}`
2. Process the task via the Task API
3. Retrieve the response (batch ID)
4. POST a results query:
 - `type : Arionbanki.Payment.ResultBatch`
 - `sendContent : {"batchId": "<batch-id-from-step-3>"}`
5. Process and retrieve payment results

Expected Results

- Payment.Batch accepts the payment batch and returns a batch ID
 - Payment.ResultBatch returns status for each payment in the batch
 - Results include success/failure status per payment line
-

Scenario 8: Currency Rates

Area: Reference Data

Setup

1. Complete Scenario 2 (credentials configured)

Steps

1. POST a Cloud Event message via the Queue API:

- `type : Arionbanki.CurrencyRates.Get`
- `sendContent : {}`

2. Process the task via the Task API
3. Retrieve the response from the Data API

Expected Results

- The response contains current currency exchange rates from Arion Banki
 - Rates include buy/sell rates for common currencies (EUR, USD, GBP, DKK, etc.)
 - Rate data includes the effective date
-

Scenario 9: Account Verification

Area: Account Management

Setup

1. Complete Scenario 2 (credentials configured)

Steps

1. POST a Cloud Event message:
 - `type : Arionbanki.Account.Get`
 - `sendContent : {}`
2. Process and retrieve the account list
3. POST a verification message:
 - `type : Arionbanki.Account.Verify`
 - `sendContent : {"accountNumber": "<test-account-number>", "kennitala": "<test-kennitala>"}`
4. Process and retrieve the verification result

Expected Results

- Account.Get returns a list of accounts accessible with the configured credentials
 - Account.Verify confirms whether an account belongs to the specified kennitala
 - Verification result is a clear true/false with descriptive messaging
-

Scenario 10: Foreign Payment Submission

Area: Foreign Payments

Setup

1. Complete Scenario 2 (credentials configured)
2. Ensure a foreign currency account exists in the test environment

Steps

1. POST a Cloud Event message:
 - `type : Arionbanki.ForeignPayment.EnterBatch`
 - `sendContent : {"payments": [{"beneficiaryName": "Test Recipient", "beneficiaryIBAN": "DE89370400440532013000", "amount": 100, "currency": "EUR", "reference": "FPAY-001"}]}`

2. Process the task via the Task API
3. Retrieve the batch response
4. POST a receipt query:
 - `type : Arionbanki.ForeignPayment.GetReceipts`
 - `sendContent : {"dateFrom": "2026-07-01", "dateTo": "2026-07-07"}`
5. Process and retrieve receipts

Expected Results

- `ForeignPayment.EnterBatch` accepts the payment and returns a batch ID
 - `ForeignPayment.GetReceipts` returns available payment receipts
 - Foreign payment data includes IBAN, SWIFT/BIC, and currency information
-

Scenario 11: Foreign Currency Account Statements

Area: Foreign Payments

Setup

1. Complete Scenario 2 (credentials configured)
2. A foreign currency account exists in the test environment

Steps

1. POST a Cloud Event message:
 - `type : Arionbanki.ForeignStatement.Accounts.Get`
 - `sendContent : {}`
2. Process and retrieve the foreign account list
3. POST a transactions query:
 - `type : Arionbanki.ForeignStatement.Transactions.Get`
 - `sendContent : {"accountId": "<foreign-account-id>", "dateFrom": "2026-01-01", "dateTo": "2026-06-30"}`
4. Process and retrieve transactions

Expected Results

- `ForeignStatement.Accounts.Get` returns available foreign currency accounts
 - `ForeignStatement.Transactions.Get` returns transaction history for the specified account
 - Transactions include amounts in the foreign currency and ISK equivalents
-

Scenario 12: Error Handling — Invalid Credentials

Area: Error Handling

Setup

1. Complete Scenario 1 (extension installed, but with INVALID credentials configured)

Steps

1. Enter invalid API credentials in the Arion banki setup
2. POST a Cloud Event message:
 - `type : Arionbanki.Account.Get`

- `sendContent : {}`

3. Process the task via the Task API
4. Retrieve the response from the Data API

Expected Results

- The task processes without crashing Business Central
 - The response contains a structured error JSON with a clear error message
 - The error indicates authentication failure (401/403)
 - No sensitive credential information is exposed in the error message
-

Scenario 13: Error Handling — Connection Failure

Area: Error Handling

Setup

1. Complete Scenario 2 (valid credentials configured)
2. Simulate a connection failure (e.g., configure an invalid API endpoint URL)

Steps

1. Modify the API base URL to a non-existent endpoint
2. POST a Cloud Event message:
 - `type : Arionbanki.CurrencyRates.Get`
 - `sendContent : {}`
3. Process the task via the Task API
4. Retrieve the response from the Data API

Expected Results

- The task processes without crashing Business Central
 - The response contains a structured error JSON indicating connection failure
 - The error message is actionable (mentions connectivity or endpoint issue)
 - The call stack is included for troubleshooting
-

Scenario 14: Permission Sets

Area: Security & Permissions

Setup

1. Complete Scenario 2 (credentials configured)
2. Create a test user with no special permissions

Steps

1. Assign only the BankAccReconciliation permission set to the test user
2. Log in as the test user
3. Attempt to use the Arionbanki.Statement.Get message type
4. Remove the permission set
5. Attempt the same operation again

Expected Results

- With the permission set assigned, the user can execute bank statement operations
 - Without the permission set, the operation fails with a permission error
 - The permission error message clearly indicates which permission is missing
-

Scenario 15: Uninstallation

Area: Installation & Activation

Setup

1. Complete Scenario 2 (extension installed and configured)

Steps

1. Navigate to Extension Management in BC
2. Uninstall the "Origo Cloud Events Arionbanki" extension
3. Verify the Cloud Events Setup page still functions (Core remains)
4. Verify no Arion Banki message types appear in the type list

Expected Results

- The extension uninstalls cleanly without error
- The Cloud Events Core extension continues to function normally
- All Arion Banki-specific message types are removed
- No orphaned data or broken references remain