

SmartCredit Partner API Guide

Agent Login As Customer & API Testing Reference — Production

Overview

This guide covers the end-to-end flow for programmatically logging in as a customer using the Partner API, then using the resulting Bearer token to test SmartCredit API endpoints. The full flow is:

1. Authenticate with the Partner API to get a PAPI access token
2. Use that token to generate a One-Time Code (OTC) for a specific customer
3. Exchange the OTC for a Bearer token on the SmartCredit API
4. Use the Bearer token to call SmartCredit API endpoints

■ IP Whitelist Required: ConsumerDirect must whitelist your IP address. It must be a public, static U.S. IP address.

Step 1 — Authenticate with the Partner API

Exchange your client key and secret for a PAPI JWT Access Token. Include this token in the Authorization header for all subsequent Partner API calls.

Production URLs

- Authorization URL: <https://auth.consumerdirect.io>
- Partner API URL: <https://papi.consumerdirect.io>

Request

```
curl --header "Content-Type: application/x-www-form-urlencoded" \
  --request POST \
  --user "<client_id>:<client_secret>" \
  --data "grant_type=client_credentials&scope=target-entity:<entity_id>" \
  https://auth.consumerdirect.io/oauth2/token
```

Response

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6Ikp1...",
  "expires_in": 59,
  "scope": "target-entity:<entity_id>",
  "token_type": "Bearer"
}
```

Step 2 — Generate a One-Time Code (OTC) for a Customer

Using the PAPI access token from Step 1, call the login-as endpoint for the target customer. The agentId value identifies who is making the request (e.g. company name, rep name).

Request

```
curl --request POST \
  --url https://papi.consumerdirect.io/v1/customers/<customer_token>/otcs/login-as \
  --header 'Content-Type: application/json' \
  --header 'Accept: application/json' \
  --header 'Authorization: Bearer <access_token>' \
  --data '{ "agentId": "<agent_id>" }'
```

Response

```
{
  "code": "pola-CUsUk7QeE0Qs34z3VpP8l65d",
  "expirationDateTime": "2023-11-30T18:58:09.706287-07:00",
  "type": "LOGIN_AS"
}
```

■■ The OTC has an expiration time referenced in the response. If the code is not exchanged before it expires, Step 3 will fail.

Step 3 — Exchange the OTC for a Bearer Token

Use the OTC from Step 2 to obtain a JWT Bearer token from the SmartCredit API. This token is used for all subsequent API calls.

```
GET https://api.smartcredit.com/v1/login?code={OTC}
```

- A valid OTC returns a JWT Bearer token
- Include this token in the Authorization header for all subsequent calls:

```
Authorization: Bearer {token}
```

■■ JWT is only valid for 5–15 minutes. Restart the OTC process if there are long delays.

Step 4 — Testing API Endpoints

With a valid Bearer token, use the following endpoints to verify the customer session and test functionality.

Testing Successful Login

Recommended first step to confirm a valid Bearer token. Will fail if token was incorrectly generated or has expired.

```
GET https://api.smartcredit.com/v1/customer
```

Testing Member Plan

Tests details about the test customer's member plan. Useful when a customer is signed up under a specific PID (e.g. planType=PREMIUM should have included3BReportsPerMonth > 0).

```
GET https://api.smartcredit.com/v1/customer/member-plan
```

Testing Remaining Statement Counts

Returns the test customer's remaining allowed activities for the current statement period. Common check: `threeBureauReportRemainingCount > 0` for any new statement period.

GET `https://api.smartcredit.com/v1/customer/statement`

Example response:

```
{
  "openDate": "2021-10-07",
  "closeDate": "2021-11-06",
  "actionRemainingCount": 1,
  "threeBureauReportRemainingCount": 0,
  "creditReportRemainingCount": 1,
  "privacyRemovalRemainingCount": 5
}
```

Testing 3B Metadata

Returns a list of 3Bs ordered in the past year. The first in the list is the latest — `orderDate` should be within the current statement. Useful for testing automatic provisioning of a 3B on statement renewal.

GET `https://api.smartcredit.com/v1/credit/3bs/metadata`

Testing 3B Polling (Current)

Tests that a previously ordered 3B can be properly instantiated and returned. Returns mock data for test customers, but still verifies that delivery services are reachable and online.

GET `https://api.smartcredit.com/v1/credit/3bs/current`

Testing 3B Polling (Historical)

Used for accessing historical reports. If the test customer has been used across PIDs, historical 3B orders may be from different bureaus. Best practice: retrieve `orderId` from the 3B Metadata endpoint.

GET `https://api.smartcredit.com/v1/credit/3bs/{orderId}`

Testing 3B Ordering

Tests that 3B order tracking is functional. A subsequent call to the statement endpoint should show a decremented `threeBureauReportRemainingCount`.

Recommended flow:

1. Call GET `/v1/customer/statement` and confirm `threeBureauReportRemainingCount > 0`
2. Call POST `/v1/credit/3bs` with the following body:

```
{ "isNonPaid": true }
```
3. Call GET `/v1/credit/3bs/current` to retrieve the report

■■■ **IMPORTANT:** Do not test 3B Ordering if there is any risk the test customer flag has been lost. Calling this endpoint on a production system with a non-test customer may result in a credit card charge. Always include `isNonPaid: true` in the request body to prevent this.