Authorize.Net

> The Authorize Net Hosted Payment Setup process allows administrators to configure Authorize.Net as an integrated payment gateway within the system. Its primary purpose is to enable users to securely pay for items using an external hosted payment form while automatically syncing the transaction data back to the user's OPSCOM profile. This article is intended for OPSCOM administrators.

Setup and Configuration

Before configuring the payment type in OPSCOM, ensure your Authorize.Net merchant account is fully established and properly configured for online transactions.

Verifying Merchant Profile Settings

Your account must be properly configured for e-commerce transactions to process payments from the portal.

1. Log in to your Authorize.Net portal.
2. Click **Account** and then select **Merchant Profile**.
3. Verify that the Product Type is set to **Card Not Present (CNP)** and the Market Type is set to **eCommerce** under the Business Information heading.

Generating API Credentials

To connect the systems, you must provide three unique pieces of identification to the Tomahawk Technologies (OPSCOM) support team.

1. Click **Account** and navigate to the API credentials area.
2. Locate your **API Login ID** (this was defined during your initial account creation).
3. Select the option to generate a new transaction key and click **Submit** to reveal your **Transaction Key**.
4. Select the option to generate a new signature key and click **Submit** to reveal your **Signature Key**.
5. Copy these three specific values and supply them to your OPSCOM representative to complete the backend system integration.

Using this Feature

Administrators manage the appearance and security behavior of the hosted checkout page directly from the Authorize.Net portal. Because OPSCOM handles the transaction handoff automatically, administrators do not manually process these specific web payments on a daily basis.

Customizing the Payment Form

1. Click **Account** and select **Payment Form**.
2. Adjust the form styling to present a consistent look and feel for your end users.
3. Disable the shipping address fields to remove checkout confusion, as shipping is not required for parking transactions.

Configuring Address Verification Service

If you are utilizing Address Verification Service (AVS) for enhanced fraud protection, you must configure the hosted form to enforce these rules.

1. Navigate to the payment form settings.
2. Set the specific billing address fields to required. This ensures the gateway captures the data necessary to perform the AVS check against the provided credit card.

> **Keep Billing Details Visible:** Even though OPSCOM automatically connects the final payment to the logged-in user, **it is highly recommended to keep the billing details visible and editable** on the payment form. Users frequently use credit cards belonging to someone else (such as a spouse, parent, or corporate account), and the billing information must match the cardholder, not necessarily the OPSCOM user profile.

> **Receipt and Response Pages:** You do not need to manually configure the receipt page or response details within Authorize.Net. As part of the secure payment protocol, OPSCOM automatically supplies this routing information and handles the final receipt generation on the user portal.

Best Practices and Considerations

- **Environment Separation: Always use the Sandbox portal** (`[https://sandbox.authorize.net/](https://sandbox.authorize.net/)`) for initial integration and testing to ensure no transactions are accidentally pushed through a real payment processor. Only use the Production

portal (`[https://account.authorize.net/](https://account.authorize.net/)`) for live, active transactions.

- **Simulate Failure States:** When configuring your system in the Sandbox environment, review the official Authorize.Net developer testing guide. Use their designated test card numbers and specific penny amounts to generate failure states (like declined cards or AVS mismatches) to ensure OPSCOM handles the checkout errors gracefully before going live.

Revision #4
Created 14 May 2024 07:37:36
Updated 12 August 2026 12:02:29