

# Payments Configuration

- [Manage Deposit Types](#)
- [Managing Payment Types](#)
- [Managing Convenience Fees](#)
- [Taxes](#)
- [The Invoice Template](#)
- [Setup payments with Flex Dollars](#)
- [Stripe Payment Provider Setup](#)

# Manage Deposit Types

**Deposit Types** allow administrators to define and manage various categories of deposits (e.g., for permits, equipment, or specific services). This feature is essential for accurately tracking refundable amounts, providing clear financial oversight, and ensuring proper accounting for deposited funds within the system.

## Using this Feature

The **Manage Deposits** page provides a straightforward interface to add, edit, and view the usage of various deposit types.

1. Hover over **System Configuration**, then **Payments**, and click **Manage Deposit Types**.

## Adding a Deposit Type

1. Click **Add Deposit Type**.
2. Fill out the information as needed:
  - **Label:** This is the name that will be used to identify the deposit type (e.g., "Permit Deposit," "Key Fob Deposit").
  - **Placeholder Text:** This field is optional. It provides a prompt or sample information that will appear in the deposit information text field when creating a deposit record.
  - **Flags:** This toggle controls the ability to edit the **Assignable Text** field in the resulting active deposit record. This text will appear in the user's profile associated with the deposit.
  - **Amount:** The amount that the deposit will cost.

- **Active checkbox:** This determines if the deposit is currently available to be assigned.

3. Click **Save Changes**.

### Editing a Deposit Type

1. Locate the deposit type and click the **Edit** button next to it.
2. Make any necessary changes to the fields.
3. Click **Save Changes**.

### Deleting a Deposit Type

1. If no records are associated with this deposit you can click the **Delete** button.
- 

## Best Practices & Considerations

- **Clear Labeling:** Use descriptive labels for your deposit types (e.g., "Annual Permit Deposit," "Visitor Pass Deposit") to ensure clarity for both administrators and users.
- **Placeholder Text Utility:** Utilize the **Placeholder Text** to provide clear instructions or examples for administrators when entering deposit information.
- **Understanding "Flags":** The "Flags" setting for `Assignable Text` can be useful for adding specific, customizable notes to individual deposit records on user profiles, allowing for unique details per transaction.
- **Financial Reconciliation:** Accurately tracking deposit types is crucial for financial reconciliation and managing refundable amounts.

# Managing Payment Types

**Managing Payment Types** allows administrators to control which payment methods are available throughout the system. This comprehensive control enables you to define new payment options, assign them to specific user types (e.g., Staff, Students, Public), and configure their usage for subscriptions or guest payments, thereby streamlining transactions and ensuring alignment with your organization's financial policies.

## Using this Feature

The **Payment Types** screen gives you a comprehensive interface to create, edit, and view details about different payment methods available in OPSCOM.

1. Go to **System Configuration**, then **Payments**, and click **\$ Setup Payment Types**.

## Adding Payment Types

**Use Case** - For this example, we will setup a Payment type that will use the provider Always Successful. This will allow payments to immediately be flagged as fully processed with no intervention of any admin. This is often used for things like Payroll deduction. For example, you want your users to be able to purchase a permit through Payroll deduction that will immediately be valid upon checkout. The actual payments will take place through payroll and have nothing to do with OPSCOM.

1. Click the **Add Type** button.

2. The list of **Providers** is hard coded and controlled by Tomahawk. You will not be able to change this picklist. The tooltip below the provider will tell you what each provider means.
3. Pick **Always Successful** for this example.
4. Click **Next Step**.
5. Provide a **Type Name**.
6. Choose the **User Types** this will be available to.
7. On the **Settings** tab, Enable the checkboxes as you required.
8. In the section **Allowed Payment Types**, you will see any payment types that are currently active first, then a list of inactive payment types after.
9. Click **Update Payment Type**.

## Hosted Payment Providers

The following links will detail the process of setting up a Hosted Payment provider, of which currently support. Click the link matching the payment provider you are adding below and follow the steps outlined on those articles.

- [OpenEdge](#):
  - We need the following credentials to work with OpenEdge;
  - **Auth Key**: a multi-digit string key
  - **Terminal ID**: a 8 digit numeric key
  - **X-Web ID**: a 12 digit numeric key
- [Moneris Checkout](#):
  - We need the following credentials to work with Moneris Checkout;
  - **store\_id**: *store#*
  - **api\_token**: *xxxxxxx*
  - **checkout\_id**: *xxxxxxxxStore#*

- [Bambora Checkout](#):

- We need the following credentials to work with Bambora Checkout (JSBambora):
  - **Payment Authorization Header:** An alphanumeric passcode
  - **Profile Authorization Header:** An alphanumeric passcode

Once you retrieve the required values from your hosted payment provider of choice, save the credentials securely and contact OPSCOM Support for further instructions.

The information you obtain in the above documentation are **credentials that secure the information exchanged between OPSCOM and the provider**. Please save these in a secure manner, it is important they are stored properly to ensure operational security.

## Extended Payment Information

For each payment type, you can view extended information on the Payment Types screen.

- A **general description** of the payment type.
- An **indication of what the user will be prompted for** when selecting that payment type. For example, in a hosted payment scenario like EdgeExpress, the user might be prompted to enter an email address before being redirected to the payment provider to enter credit card information.
- The number of users associated to this payment type.

There are also three columns on the far right that use an **X** or **checkmark** to indicate specific uses of the payment type:

- **Column 1:** Indicates if this payment type can be used for **guest payments**.
- **Column 2:** Indicates if this payment type can be used for **permit and invoice renewal payments**.
- **Column 3:** Indicates if this payment type can be used for **Text2ParkMe subscriptions**.

## Configuring Payment Types for Subscriptions and Guest Payments

You can fine-tune how specific payment types are used for subscriptions and guest payments.

1. Find the payment type you wish to configure and click the **Pencil icon**.
2. Click the **Settings** tab.
3. Select the items you wish to allow the payment type to be used for (e.g., enable **Text2ParkMe, subscriptions to permit renewals**, as well as **guest payments**).
4. Prompt Information can be changed as well. You can enable only an **email address** or **full address information**. Full address info is required when the provider is doing Address Verification using AVS, 3DS or other similar service. This controls what shows up on the **QR Code reservations** screen after the **Payment Type** is selected.
5. Click **Update Payment Type**.

## Processing as an Admin

When processing payments as an administrator, the payment types presented will be split into two categories:

- What the user would be allowed to use for their specific User Type.
  - What an administrator can choose (which may include additional options not visible to the user).
- 

## Best Practices & Considerations

- **Granular Control:** Leverage User Types to restrict payment methods, ensuring users only see relevant options. For example, you might only allow "Payroll Deduction" for Staff users.
- **Clear User Experience:** By limiting available payment types based on context or user, you simplify the checkout process and reduce confusion for users.
- **Refund Alignment:** Ensure that appropriate refund types are linked to their corresponding payment types to facilitate smooth and accurate refund processing.
- **Subscription Compatibility:** When enabling payment types for subscriptions (recurring billing), verify that the chosen payment gateway provider fully supports recurring transactions.
- **Regular Review:** Periodically review your payment type configurations and their associated settings to ensure they remain aligned with your organization's financial policies and evolving needs.

# Managing Convenience Fees

**Convenience Fees** in OPSCOM allow administrators to add an additional flat-rate charge directly to user payments, managed entirely within the OPSCOM system. This feature provides flexibility to offset processing costs, incentivize certain payment methods, or cover administrative burdens, distinct from any transaction fees imposed by external payment providers.

## Difference Between Transaction Fees and Convenience Fees

It's crucial to understand the distinction between these two types of fees:

- **Transaction Fees:** These are charges applied by your **payment provider** (e.g., credit card processor) for processing a payment. They are independent of OPSCOM's internal settings. Depending on your setup, OPSCOM and end-users may not be directly aware of these fees as they are managed solely by the payment provider.
- **Convenience Fees:** These are additional charges that are **managed and applied directly through OPSCOM**. For example, your OPSCOM system can add a \$5 convenience fee onto a payment. This fee amount is visible within OPSCOM and is added to the grand total before being sent to the payment provider.

It is possible for a system to utilize both Convenience Fees (managed by OPSCOM) and Transaction Fees (managed by the payment provider), but recognizing their distinct origins and management is important.

---

## Using this Feature

The **Fees Page** allows you to set up various convenience fees.

You can only use Convenience fees if you have Bambora/Worldline as your payment provider.

## Fee Limitations

Before configuring, understand these limitations:

- **Applied Once Per Order:** Fees are applied once per fee type per order. This means if a fee is set for permits, only a single fee is applied to the order, regardless of whether it contains one permit or five permits.
- **Flat Rate Only:** Fees are currently a flat rate. There is no percentage-based fee option at this time.
- **Grand Total Sent:** Fees are not sent to the payment provider as a secondary item. Only the grand total (item price + convenience fee) is sent for processing.

## Fee Examples and Setup

### Admin Collected Fee for Collecting Cash and Cheques (Deterrent Fee)

- **Scenario:** The administrator wants to encourage online payments. A fee is set up as a deterrent when users pay with cash or cheque, covering the increased administrative burden.
- **Setup:** The fee would apply to all item types but only to payment types of **Cash** and **Cheque**.

### Admin Collected Fee for Permits

- **Scenario:** Permits are sent to a third party for enforcement, and that third party charges \$0.50 per permit they enforce. The client wishes to pass this cost onto the user.
- **Setup:** The fee would apply to all payment types, but only to **Permits** and **Temp Permits**. The setup would specify  as the fee amount.

### Admin Charges a Convenience Fee to Cover an OPSCOM Collected Transaction Fee

- **Scenario:** There's an agreement with the payment provider that all permits will incur a \$2.95 transaction fee paid to OPSCOM. The client still wants to receive the full \$150 for a purchased permit, so they set up a convenience fee to cover this cost.
- **Setup:** The convenience fee will be charged when selecting the specific payment type that incurs the \$2.95 transaction fee. For example, the setup would apply to the relevant payment provider and define the convenience fee as .

### Admin Includes the Transaction Fee in the Price of the Item (No Fee Setup in OPSCOM)

- **Scenario:** An agreement exists where all permits incur a \$2.95 transaction fee paid to OPSCOM. However, the client chooses to *absorb* this fee by increasing the item's price. If a permit costs \$150, then \$2.95 of that will be sent to OPSCOM, leaving the client with \$147.05.
- **Setup: There is no specific setup for fees in this scenario within OPSCOM**, as the fee is already embedded in the item's base price. If

you want to inform users that a transaction fee is included in the final amount, you will need to update a relevant footer message (e.g., in email templates or on the user portal) stating this.

---

## Setting Up the Fees

You can access the Fees page to accommodate any of the scenarios above.

1. Hover over **System Configuration**, then **Payments**, and click **Fees**.
2. Click **Add Fee** and follow the diagram below.
3. Click the **Update** button to save.

# Fees

Back

Enabled



The flat rate applied to each purchase

Amount

0.85

| Text      | Language         | Translation |
|-----------|------------------|-------------|
| Fee Label | English (en)     | Payment Fee |
|           | Français (fr_ca) |             |

The name of the fee that shows up on the cart.

Apply to purchases with at least one of the following items:

Apply to all

Access Cards

Adjustments/Refunds

Deposits

Invoices

Lockers

Permits

Temp Permits

Violations

You can have the fee only show up if a specific type of item is being purchased.

Apply to purchases made with the following payment types:

Apply to all

Cash

Payroll Deduction

Other Refund

HP: EdgeExpress

Banner Payment

Electronic Funds Transfer

HP: Authorize

HP: PayPal

JS: Moneris Checkout

JS: Bambora Checkout

HP: Touchnet

HP: Exact

HP: SchoolPay

Visa

MasterCard

American Express

GW: Moneris

You can define which

---

## Best Practices & Considerations

- **Clarity for Users:** Clearly communicate any convenience fees to your users before they complete a transaction. Transparency builds trust.
- **Strategic Application:** Use convenience fees strategically. They can deter cash payments, offset third-party costs, or cover transaction fees, but ensure they align with your overall pricing strategy.
- **Flat Rate Limitation:** Remember that fees are flat rate, not percentage-based. Plan your fee amounts accordingly.
- **Impact on Grand Total:** Be aware that OPSCOM adds the convenience fee to the item's price to form the grand total before sending it to the payment provider.
- **External Transaction Fees:** Distinguish between OPSCOM's convenience fees and any external transaction fees imposed by your payment gateway. Understand where each fee is applied and how it impacts your revenue.
- **Reporting:** Consider how convenience fees will be reported and accounted for in your financial reconciliation processes.

# Taxes

The **Taxes** configuration page in OPSCOM allows administrators to define and manage the tax percentages applied to payments made within the system. This feature is crucial for ensuring accurate tax calculation during checkout, maintaining compliance with local tax regulations, and providing transparency to users regarding applicable taxes.

## Setup & Configuration

You must be a Primary Admin with the permission to **Manage System Configuration** to be able to edit the system settings.

If you do not see the settings mentioned in this article, please contact [support@ops-com.com](mailto:support@ops-com.com) to have them enabled for you. You will need, in writing, the permission of your Primary Admin.

## Using this Feature

The **Taxes** page allows you to configure multiple settings for each tax type, ensuring accurate calculation and display.

1. Hover over **System Configuration**, then **Payments**, and click **Taxes**.

## Editing a Tax Type

When editing an existing tax type, the following settings are available:

- **Enable**: Check this box to **enable** the tax type, making it active and applied to transactions. Uncheck to **disable** it.

- **Percentage:** Enter the numerical percentage of tax to be calculated (e.g.,  for 13%).
  - **Tax Label & Translation:** This field defines how this tax will be displayed to users during the cart and checkout process (e.g., "HST," "Sales Tax," "VAT"). Multiple languages are supported for translation to ensure clarity for diverse users.
  - **Apply to the following items:** Check the specific system modules or item types to which you want this tax to be applied (e.g., Permits, Violations, Lockers, Access Cards).
- 

## Best Practices & Considerations

- **Compliance:** Ensure your tax configurations (percentages, labels, and applicability) are always compliant with current local, provincial/state, and national tax regulations.
- **Clarity for Users:** Use clear and understandable **Tax Labels** that resonate with your users (e.g., "HST" for Canadian users where applicable).
- **Module Applicability:** Carefully select the "**Apply to the following items**" checkboxes to ensure taxes are only applied to the correct modules and items, preventing incorrect charges.
- **Testing:** After making any changes to tax settings, perform test transactions in a [Preview Space](#) to verify that taxes are calculated and displayed correctly in the cart and on receipts.
- **Communication with Finance:** Coordinate closely with your organization's finance or accounting department to ensure OPSCOM's tax configuration aligns with their reconciliation processes.

# The Invoice Template

## Editing the template

To access the template, go to the **System Config** menu and click on **Email Templates**.

The screen will refresh with a list of possible templates to edit. In our case, we want to click on the **Edit** button for **Permit Invoice** to access the template.

| Email Templates          |                      |                            |          |   |   |
|--------------------------|----------------------|----------------------------|----------|---|---|
| Title                    |                      | Type                       | Language | Subject   | Notes   |
| Appeal Submitted by User | <a href="#">Edit</a> | Appeal Submitted           | English  | Appeal Submitted                                | Appeal submitted by user  |
| Incident Report          | <a href="#">Edit</a> | Incident Report            | English  | Incident Report [INCIDENTNO]                    | Not used at this time.  |
| Lost Password            | <a href="#">Edit</a> | Lost Password              | English  | Your Password                                   | The lost my password email  |
| Permit Invoice           | <a href="#">Edit</a> | Permit Invoice             | English  | Invoice is ready: [invoice account="name"]      | Send emailed invoices to companies that have outstanding permits.   |
| User Receipt             | <a href="#">Edit</a> | User Receipt               | English  | Payment Request Received                        | The receipt the user receives after a payment has been submitted.   |
| Waitlist Selected        | <a href="#">Edit</a> | Selected From Waiting List | English  | Your Permit Is Ready [waitlist show="fullName"] | The notification to the user that the admin has either selected them from the waitlist to receive a permit, or extended the time they have to pay for a permit before picking someone else. |

The screen will refresh with the complete Template form as seen below.

# Edit Email Template

[Return to Template List](#)

Send emailed invoices to companies that have outstanding permits.

**Label**

**Type** Permit Invoice

**Language** English Français

**Layout**

**Subject**

## HTML Content

This email uses special tokens that are replaced when sent. The tokens available for this email template include:

**[invoice account="number"]** The account number for the company.

**[invoice account="name"]** The account name for the company.

**[invoice account="fulladdress"]** The full address for the company.

**[invoice account="invoice\_comments"]** The invoice comment value that is editable for companies only. Any line breaks will be preserved.

**[invoice show="invoiceno"]** Invoice Number.

**[invoice show="duedate"]** Invoice due date.

**[invoice show="billdate"]** Invoice bill date.

**[invoice show="subtotal"]** Subtotal of all items before tax.

**[invoice show="taxtotal"]** Total amount of tax.

**[invoice show="amounttotal"]** The total sum amount for all outstanding permits for this invoice.

**[invoice show="counttotal"]** The total number of permits for the selected sales window.


**[invoice show="details"]** The list of permits, cost and who has the permit.

File Edit Insert View Format Table Tools

← → + Formats B I A A ☰ ☷ ☹ ☺ ☻ ☼ ☽ ☾ ☿

🔗 🖼️ {;} ?

92 Bridge Street  
Carleton Place, Ontario



**TOMAHAWK UNIVERSITY**  
Home of the GPS-CCM demo  
**Tomahawk University Parking**

Invoice

|          |   |              |                            |
|----------|---|--------------|----------------------------|
| Bill To: | [invoice account="fulladdress"]<br>Account Number: [invoice account="number"] | Invoice No.: | [invoice show="invoiceno"] |
|          |   | Date:        | [invoice show="billdate"]  |
|          |   | Due Date:    | [invoice show="duedate"]   |

[invoice show="details"]

|          |                                  |
|----------|----------------------------------|
| Subtotal | [\$[invoice show="subtotal"]]    |
| Taxes    | [\$[invoice show="taxtotal"]]    |
| Total    | [\$[invoice show="amounttotal"]] |

**Balance Due** \$[invoice show="amounttotal"]

This is a test of the text block at the bottom of the page:

[invoice account="invoice\_comments"]

Powered by TinyMCE

table » tbody » tr » td » p Words: 86

## Text Content

[Save Email Template](#)

The form contains 4 areas:

At the top, **Meta Data** for this specific template.

On the left, a list of possible **Short Codes** that can be used on this template form.

On the right, the actual **Template**.

At the bottom, a **Text Content field** (This field can be configured for a plain text version of the

## Meta Data

**Label** - Enter a name for this Template.

**Type** - This field auto fills.

**Language** - Choose the language of the template.

**Layout** - Choose default layout.

**Subject** - This text will appear as the Email Subject. Note that the use of short code is permitted in this field. In this case, we are calling up the account name for the company being invoiced.

## Short Code List

The template allows the use of special tokens, or short codes, to pull information from the database to appear on the invoice when sent.

These include:

**[invoice account="number"]** The account number for the company.

**[invoice account="name"]** The account name for the company.

**[invoice account="fulladdress"]** The full address for the company.

**[invoice account="invoice\_comments"]** The invoice comment value that is

editable for companies only. Any line breaks will be preserved.

**[invoice show="invoiceno"]** Invoice Number.

**[invoice show="duedate"]** Invoice due date.

**[invoice show="billdate"]** Invoice bill date.

**[invoice show="subtotal"]** Subtotal of all items before tax.

**[invoice show="taxtotal"]** Total amount of tax.

**[invoice show="amounttotal"]** The total sum amount for all outstanding permits for this invoice.

**[invoice show="counttotal"]** The total number of permits for the selected sales window.


**[invoice show="details"]** The list of permits, cost and who has the permit.

HTML Content

This email uses special tokens that are replaced when sent. The tokens available for this email template include:

- [invoice account="number"]** The account number for the company.
- [invoice account="name"]** The account name for the company.
- [invoice account="fulladdress"]** The full address for the company.
- [invoice account="invoice\_comments"]** The invoice comment value that is editable for companies only. Any line breaks will be preserved.
- [invoice show="invoiceno"]** Invoice Number.
- [invoice show="duedate"]** Invoice due date.
- [invoice show="billdate"]** Invoice bill date.
- [invoice show="subtotal"]** Subtotal of all items before tax.
- [invoice show="taxtotal"]** Total amount of tax.
- [invoice show="amounttotal"]** The total sum amount for all outstanding permits for this invoice.
- [invoice show="counttotal"]** The total number of permits for the selected sales window.
- [invoice show="details"]** The list of permits, cost and who has the permit.

92 Bridge Street  
Carleton Place, Ontario



**TOMAHAWK UNIVERSITY**  
Home of the OPS-COM demo  
**Tomahawk University Parking**

|          |   |              |                            |
|----------|---|--------------|----------------------------|
| Bill To: | [invoice account="fulladdress"]<br>Account Number: [invoice account="number"] | Invoice No.: | [invoice show="invoiceno"] |
|          |   | Date:        | [invoice show="billdate"]  |
|          |   | Due Date:    | [invoice show="duedate"]   |

[invoice show="details"]

|          |                                |
|----------|--------------------------------|
| Subtotal | \$[invoice show="subtotal"]    |
| Taxes    | \$[invoice show="taxtotal"]    |
| Total    | \$[invoice show="amounttotal"] |

|                    |                                       |
|--------------------|---------------------------------------|
| <b>Balance Due</b> | <b>\$[invoice show="amounttotal"]</b> |
|--------------------|---------------------------------------|

This is a test of the text block at the bottom of the page:

[invoice account="invoice\_comments"]

These tokens can be placed anywhere on the template and will pull data from the system, similar to a mail merge functionality.

For example, in the **Bill To:** area, the system will retrieve the **Full Address of the Company** as well as the **Company Account Number** using these two short code tokens:

**[invoice account="fulladdress"]** The full address for the company.

**[invoice account="number"]** The account number for the company.

The screenshot shows an HTML email template editor. On the left, a list of tokens is provided with their descriptions:

- [invoice account="number"]** The account number for the company.
- [invoice account="name"]** The account name for the company.
- [invoice account="fulladdress"]** The full address for the company.
- [invoice account="invoice\_comments"]** The invoice comment value that is editable for companies only. Any line breaks will be preserved.
- [invoice show="invoiceno"]** Invoice Number.
- [invoice show="duedate"]** Invoice due date.
- [invoice show="billdate"]** Invoice bill date.
- [invoice show="subtotal"]** Subtotal of all items before tax.
- [invoice show="taxtotal"]** Total amount of tax.
- [invoice show="amounttotal"]** The total sum amount for all outstanding permits for this invoice.
- [invoice show="counttotal"]** The total number of permits for the selected sales window.
- [invoice show="details"]** The list of permits, cost and who has the permit.

The main editor area shows a preview of the invoice template. The header includes the address "92 Bridge Street, Carleton Place, Ontario" and the Tomahawk University logo with the text "TOMAHAWK UNIVERSITY Home of the OPS-COM demo Tomahawk University Parking". The word "Invoice" is positioned on the right. The "Bill To:" section is highlighted with a red box and contains the tokens: "[invoice account='fulladdress']", "Account Number: [invoice account='number']". To the right, the invoice details are listed: "Invoice No.: [invoice show='invoiceno']", "Date: [invoice show='billdate']", and "Due Date: [invoice show='duedate']". Below this is a table with columns for "Subtotal", "Taxes", and "Total", with corresponding values like "\$[invoice show='subtotal']". A large "Balance Due" section is shown with the value "\$[invoice show='amounttotal']". At the bottom, there is a text block "This is a test of the text block at the bottom of the page:" and a token "[invoice account='invoice\_comments']".

## The Template

There are a number of areas on the template that can be edited by the admin. To add branding to the template there are two areas that will be of concern.

### The Company Logo Field:

This is where you would place your company logo. Using a .JPEG or .PNG file format is recommended.

**HTML Content**

This email uses special tokens that are replaced when sent. The tokens available for this email template include:

**[invoice account="number"]** The account number for the company.

**[invoice account="name"]** The account name for the company.

**[invoice account="fulladdress"]** The full address for the company.

**[invoice account="invoice\_comments"]** The invoice comment value that is editable for companies only. Any line breaks will be preserved.

**[invoice show="invoiceno"]** Invoice Number.

**[invoice show="duedate"]** Invoice due date.

**[invoice show="billdate"]** Invoice bill date.

**[invoice show="subtotal"]** Subtotal of all items before tax.

**[invoice show="taxtotal"]** Total amount of tax.

**[invoice show="amounttotal"]** The total sum amount for all outstanding permits for this invoice.


**[invoice show="counttotal"]** The total number of permits for the selected sales window.

**[invoice show="details"]** The list of permits, cost and who has the permit.

File Edit Insert View Format Table Tools

← → + Formats **B** *I* A A [List Icons]

92 Bridge Street  
Carleton Place, Ontario



**TOMAHAWK UNIVERSITY**  
Home of the OPS-COM demo  
**Tomahawk University Parking**

**Invoice**

Bill To: [invoice account="fulladdress"]      Invoice No.: [invoice show="invoiceno"]  
Account Number: [invoice account="number"]      Date: [invoice show="billdate"]  
Due Date: [invoice show="duedate"]

[invoice show="details"]

|          |                                |
|----------|--------------------------------|
| Subtotal | \$[invoice show="subtotal"]    |
| Taxes    | \$[invoice show="taxtotal"]    |
| Total    | \$[invoice show="amounttotal"] |

**Balance Due**    **\$[invoice show="amounttotal"]**

This is a test of the text block at the bottom of the page:

[invoice account="invoice\_comments"]

## Your Company Address:

Enter your company address here.


HTML Content

This email uses special tokens that are replaced when sent. The tokens available for this email template include:

- [invoice account="number"]** The account number for the company.
- [invoice account="name"]** The account name for the company.
- [invoice account="fulladdress"]** The full address for the company.
- [invoice account="invoice\_comments"]** The invoice comment value that is editable for companies only. Any line breaks will be preserved.
- [invoice show="invoiceno"]** Invoice Number.
- [invoice show="duedate"]** Invoice due date.
- [invoice show="billdate"]** Invoice bill date.
- [invoice show="subtotal"]** Subtotal of all items before tax.
- [invoice show="taxtotal"]** Total amount of tax.
- [invoice show="amounttotal"]** The total sum amount for all outstanding permits for this invoice.
- [invoice show="counttotal"]** The total number of permits for the selected sales window.
- [invoice show="details"]** The list of permits, cost and who has the permit.

File Edit Insert View Format Table Tools

92 Bridge Street  
Carleton Place, Ontario



**TOMAHAWK UNIVERSITY**  
Home of the OPS-COM demo  
Tomahawk University Parking

Invoice

Bill To: [invoice account="fulladdress"] Invoice No.: [invoice show="invoiceno"]  
Account Number: [invoice Date: [invoice show="billdate"]  
account="number"] Due Date: [invoice show="duedate"]

[invoice show="details"]

|          |                                  |
|----------|----------------------------------|
| Subtotal | [\$[invoice show="subtotal"]]    |
| Taxes    | [\$[invoice show="taxtotal"]]    |
| Total    | [\$[invoice show="amounttotal"]] |

**Balance Due** \$[invoice show="amounttotal"]

This is a test of the text block at the bottom of the page:

[invoice account="invoice\_comments"]

### Invoice Identification:

In this section use short code to pull in the **Invoice Number, Billing Date, and Due Date.**


HTML Content

This email uses special tokens that are replaced when sent. The tokens available for this email template include:

- [invoice account="number"]** The account number for the company.
- [invoice account="name"]** The account name for the company.
- [invoice account="fulladdress"]** The full address for the company.
- [invoice account="invoice\_comments"]** The invoice comment value that is editable for companies only. Any line breaks will be preserved.
- [invoice show="invoiceno"]** Invoice Number.
- [invoice show="duedate"]** Invoice due date.
- [invoice show="billdate"]** Invoice bill date.
- [invoice show="subtotal"]** Subtotal of all items before tax.
- [invoice show="taxtotal"]** Total amount of tax.
- [invoice show="amounttotal"]** The total sum amount for all outstanding permits for this invoice.
- [invoice show="counttotal"]** The total number of permits for the selected sales window.
- [invoice show="details"]** The list of permits, cost and who has the permit.

File Edit Insert View Format Table Tools

92 Bridge Street  
Carleton Place, Ontario



**TOMAHAWK UNIVERSITY**  
Home of the OPS-COM demo  
Tomahawk University Parking

Invoice

Bill To: [invoice account="fulladdress"] Invoice No.: [invoice show="invoiceno"]  
Account Number: [invoice Date: [invoice show="billdate"]  
account="number"] Due Date: [invoice show="duedate"]

[invoice show="details"]

|          |                                  |
|----------|----------------------------------|
| Subtotal | [\$[invoice show="subtotal"]]    |
| Taxes    | [\$[invoice show="taxtotal"]]    |
| Total    | [\$[invoice show="amounttotal"]] |

**Balance Due** \$[invoice show="amounttotal"]

This is a test of the text block at the bottom of the page:

[invoice account="invoice\_comments"]

## Invoice Details:

The short code **[invoice show="details"]** pulls in the list of permits that will appear on the Invoice. (See a sample of the Invoice below)

These include:

Parker's Name

Item type being Paid for (Permit, Access card, etc.)

Item number (Permit Number)

Sales Window

Lot Name

Quantity

Cost


**HTML Content**

This email uses special tokens that are replaced when sent. The tokens available for this email template include:

- [invoice account="number"]** The account number for the company.
- [invoice account="name"]** The account name for the company.
- [invoice account="fulladdress"]** The full address for the company.
- [invoice account="invoice\_comments"]** The invoice comment value that is editable for companies only. Any line breaks will be preserved.
- [invoice show="invoiceno"]** Invoice Number.
- [invoice show="duedate"]** Invoice due date.
- [invoice show="billdate"]** Invoice bill date.
- [invoice show="subtotal"]** Subtotal of all items before tax.
- [invoice show="taxtotal"]** Total amount of tax.
- [invoice show="amounttotal"]** The total sum amount for all outstanding permits for this invoice.
- [invoice show="counttotal"]** The total number of permits for the selected sales window.
- [invoice show="details"]** The list of permits, cost and who has the permit.

File Edit Insert View Format Table Tools

92 Bridge Street  
Carleton Place, Ontario



**TOMAHAWK UNIVERSITY**  
Home of the OPS-COM demo  
**Tomahawk University Parking**

Invoice

|          |   |              |                            |
|----------|---|--------------|----------------------------|
| Bill To: | [invoice account="fulladdress"]<br>Account Number: [invoice account="number"] | Invoice No.: | [invoice show="invoiceno"] |
|          |   | Date:        | [invoice show="billdate"]  |
|          |   | Due Date:    | [invoice show="duedate"]   |

[invoice show="details"]

|          |                                  |
|----------|----------------------------------|
| Subtotal | [\$[invoice show="subtotal"]]    |
| Taxes    | [\$[invoice show="taxtotal"]]    |
| Total    | [\$[invoice show="amounttotal"]] |

**Balance Due** \$[invoice show="amounttotal"]

This is a test of the text block at the bottom of the page:

[invoice account="invoice\_comments"]

## Payment Information (Billing Cost):


In this section use short code to pull in the billing information such as **Subtotal, Calculated Taxes, and Total Billing Amount.**

**HTML Content**

This email uses special tokens that are replaced when sent. The tokens available for this email template include:

- [invoice account="number"]** The account number for the company.
- [invoice account="name"]** The account name for the company.
- [invoice account="fulladdress"]** The full address for the company.
- [invoice account="invoice\_comments"]** The invoice comment value that is editable for companies only. Any line breaks will be preserved.
- [invoice show="invoiceno"]** Invoice Number.
- [invoice show="duedate"]** Invoice due date.
- [invoice show="billdate"]** Invoice bill date.
- [invoice show="subtotal"]** Subtotal of all items before tax.
- [invoice show="taxtotal"]** Total amount of tax.
- [invoice show="amounttotal"]** The total sum amount for all outstanding permits for this invoice.
- [invoice show="counttotal"]** The total number of permits for the selected sales window.
- [invoice show="details"]** The list of permits, cost and who has the permit.

92 Bridge Street  
Carleton Place, Ontario



**TOMAHAWK UNIVERSITY**  
Home of the OPS-COM demo  
Tomahawk University Parking

**Invoice**

|          |   |              |                            |
|----------|---|--------------|----------------------------|
| Bill To: | [invoice account="fulladdress"]<br>Account Number: [invoice account="number"] | Invoice No.: | [invoice show="invoiceno"] |
|          |   | Date:        | [invoice show="billdate"]  |
|          |   | Due Date:    | [invoice show="duedate"]   |

[invoice show="details"]

|          |                                  |
|----------|----------------------------------|
| Subtotal | [\$[invoice show="subtotal"]]    |
| Taxes    | [\$[invoice show="taxtotal"]]    |
| Total    | [\$[invoice show="amounttotal"]] |

**Balance Due** \$[invoice show="amounttotal"]

This is a test of the text block at the bottom of the page:

[invoice account="invoice\_comments"]

You may also wish to add comments to the invoice. Comments can be added in the company configuration.

# Company Edit Back

3 Users associated to this company. View Users

Account Number

Account Name

Receives Invoice

Invoice Emails

Invoice Comments

Address

City

## The Email with PDF attached:

Here is a sample of what the resulting email would look like. There is also a .PDF version of this invoice attached to the email.

92 Bridge Street  
Carleton Place, Ontario



## Invoice

Bill To:

Shannon Jones  
123 Clearview Lane  
Perth, ON  
K7H 3C8  
Account Number: 197102

Invoice No.: 1024  
Date: 10/22/2019  
Due Date: 11/21/2019

| Name          | Item   | Item Details                         | Quantity | Rate     | Amount   | Tax |
|---------------|--------|--------------------------------------|----------|----------|----------|-----|
| Jason Barnes  | Permit | 109 Window: Test Annual Lot: RED-STU | 1        | \$525.00 | \$525.00 | Tax |
| Julie Parsons | Permit | 108 Window: Test Annual Lot: RED-STU | 1        | \$525.00 | \$525.00 | Tax |

Subtotal \$1,050.00  
Taxes \$136.50  
Total \$1,186.50

**Balance Due \$1,186.50**

This is a test of the text block at the bottom of the page:

This is a test of the comment feature.

# Setup payments with Flex Dollars

This article describes the Flex Dollar and Payroll Deduction feature, a specialized payment method for handling partial payments with credits. This feature allows your users to apply a pre-defined amount of **flex dollars** towards a permit purchase, with the system automatically notifying administrators of the remaining balance to be processed via payroll.

## OPSCOM Support Configuration ▣

The **Flex Dollar Payment** feature is a *custom development*, tailored specifically for one client, Trent University. It is not intended to be useful to, or available to all clients. Its availability is controlled by specific configurations set up by an OPSCOM Support team member.

This feature requires configuration by an administration of your system, and must also be enabled by an OPSCOM support team member before it can be used. The setup involves defining the selectable flex dollar amounts, setting the notification email, and assigning the payment type to the appropriate users.

This feature is enabled by a system-level setting that is typically configured by OPSCOM support upon request. Once enabled, you may proceed with the steps below.

## Configuration and Functionality ▣

1. To enable flex dollars, hover over **System Configuration**, and click **System Settings**.
2. Click **Permits**.

3. Enable the checkbox beside **Enable Flex Dollars**.

### Configure Email Notifications ▢

An email address must be configured to receive a notification internally and each time this payment method is used (and the details of the transaction for accounting).

1. Hover over **System Configuration, Payments** and click **Manage Flex Dollars**
2. In the **Flex Dollars Alert Email** text box, enter an email address you will be monitoring.

This email address must be monitored for any changes to be made to student flex dollar balances.

### Define Flex Dollar Amount ▢

Create the specific flex dollar values that users can select during checkout.

1. Hover over **System Configuration, Payments**, and click **Manage Flex Dollars**.
2. Click the **Add New** button.
3. Enter a **Label**. This is the text the user will see in the drop-down menu (e.g., "Apply \$100 of Flex Credit").
4. Enter the corresponding numeric **Value** (e.g., ).
5. Click **Save**.
6. Repeat this process for all the pre-determined flex dollar amounts your organization offers.

### Assign Payment Method to User Types ▢

Finally, this payment method must be enabled for the specific user types who are permitted to use it.

1. Hover over **System Configuration, Users**, and click **User Types**.
2. Select the user type you wish to grant access to (e.g., Staff).
3. In the user type settings, assign the **Flex Dollars/Payroll Deduction** payment option.
4. For detailed instructions, please refer to the [User Types Administration wiki article](#).

## Using this Feature <sup>▢</sup>

Once configured, the feature provides a streamlined workflow for users and a clear notification process for administrators.

### User Purchase Workflow <sup>▢</sup>

1. An eligible user adds a standard term permit to their shopping cart.
2. On the checkout page, they select **Flex Dollars/Payroll Deduction** as the payment method.
3. A new drop-down menu appears, populated with the flex dollar **Labels** you configured.
4. The user selects an amount and completes the transaction. Their permit is considered valid and is issued immediately.

### Administrator Workflow (Post-Purchase) <sup>▢</sup>

1. Upon successful purchase, an email is automatically sent to the address configured in "**Flex Dollar Alert Email**" field.
2. This notification contains the key information needed for manual processing:

- User's Name and Contact Information
  - Permit Details
  - Flex Dollar Amount Applied
  - Remaining Balance Due
3. Your administrative team (e.g., Payroll or Parking department) uses this information to confirm users flex dollar balances the remaining balance via your organization's internal payroll system.
  4. For auditing purposes, a note detailing the flex dollar amount is also automatically added to the users transaction payment record.

The payroll deduction component is **not an automated process** and **neither is the updating of Flex Dollar balance**. The system's function is to calculate and report the remaining payroll deduction balance to administrators.

## Best Practices & Considerations

- This payment option will **only appear to eligible users when their cart contains only standard permits**. It will not be available if other items, such as violation payments or locker rentals, are also in the cart.
- **Use clear and descriptive labels** for the flex dollar amounts (e.g., "Use \$75 Payroll Credit"). This transparency is key to reducing user confusion and support requests during the checkout process.
- **Ensure the staff who manage payroll are aware of this workflow** and understand how to act on the email notifications. Clear internal communication is critical to the success of this feature.
- **External Verification & Reconciliation:** It is critical to understand that OPSCOM does not verify real-time flex dollar balances with any

external system. Administrators are solely responsible for independently verifying the spent flex dollar amounts with their HR/payroll department and managing any discrepancies or adjustments outside of the OPSCOM system.

**Note:** this type of payment is treated as a full, successful payment. The system does not have split-payment logic or backend validation of flex balances beyond recording the user's chosen amount.

# Stripe Payment Provider Setup

This feature shows how to integrate with Stripe to process credit card payments directly within the system. Its primary purpose is to provide a secure and seamless checkout experience for users while simplifying financial reconciliation for operations. This article is intended for **OPSCOM administrators** responsible for financial setup.

## Setup and Configuration

### Admin Portal

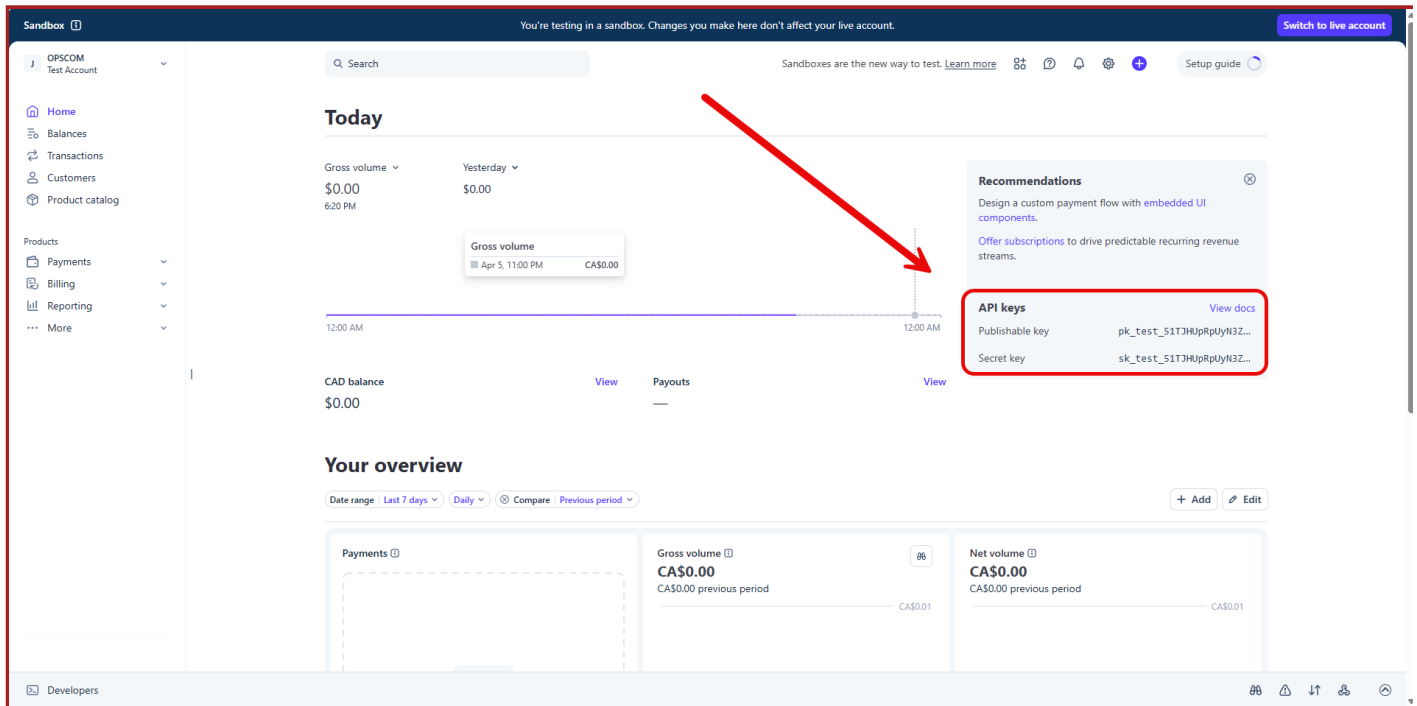
To begin the integration, an administrator must setup a payment type in OPSCOM, which will be linked to their Stripe account. This is handled through the central administration portal where you will input your unique credentials.

- Hover over **System Configuration > Payments**, then click **Setup Payment Types**.
- Click **Add Type**.
- Click the **Payment Provider** drop-down menu and choose **Stripe.js** from the available options.

Once the provider is selected, you can **name the payment type**, **select applicable user types**, and then **go to the settings tab**.

Now, you have to get the API keys for your Stripe account, and setup webhooks, which can be found on the stripe dashboard, which you should be logged into, from a previous step. Your **Publishable Key**, and **Secret Key**, can be found and copied here. You can click on these values to copy them, and

paste them into the payment type settings in OPSCOM.



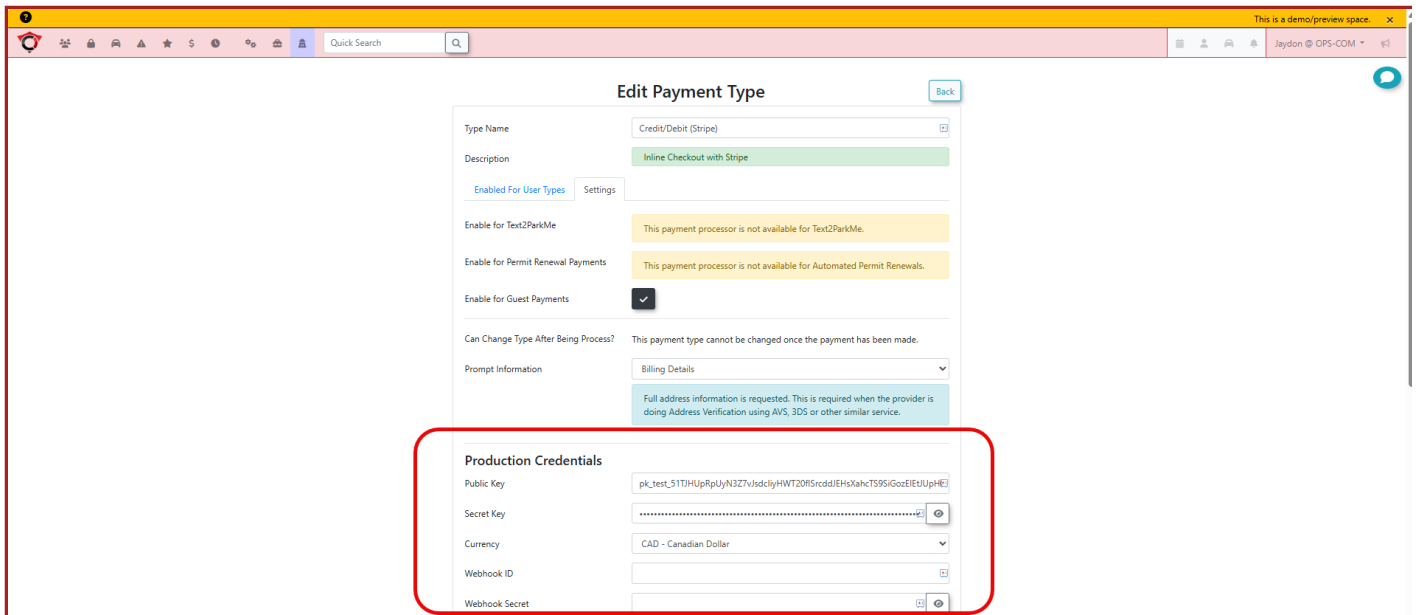
The screenshot shows the Stripe Sandbox dashboard for an account named 'OPSCOM Test Account'. A red arrow points from the top of the dashboard to the 'API keys' section in the right-hand sidebar. The 'API keys' section is highlighted with a red box and contains the following information:

| API keys        |                            | <a href="#">View docs</a> |
|-----------------|----------------------------|---------------------------|
| Publishable key | pk_test_51T3H0pRpUyN13Z... |                           |
| Secret key      | sk_test_51T3H0pRpUyN13Z... |                           |

The dashboard also displays a 'Today' summary with 'Gross volume' of \$0.00 and 'Yesterday' of \$0.00. Below this is a 'Your overview' section with various metrics like 'Payments', 'Gross volume', and 'Net volume', all showing CA\$0.00.

You must provide your unique identification keys. Enter your Stripe public key into the **Publishable Key** field and your Stripe secret key into the **Secret Key** field. Only enter the Testing Credentials until you are ready for your users to have access to this.

Take this time to ensure that the **Currency** setting is correct, in the payment type settings (the default is US dollars). The next steps will be about populating the **Webhook ID** and **Webhook Secret** fields, in OPSCOM, with the correct information from your Stripe dashboard.



## Third-Party Webhook Configuration

Webhooks are essential for Stripe to notify OPSCOM regarding payment successes or refund completions.

Without these Webhooks, the system **WILL NOT** communicate any transaction information to OPSCOM.

|     |     |
|-----|-----|
| Su  | Wh  |
| b-  | at  |
| se  | it  |
| cti | co  |
| on  | ver |
|     | s   |

**Re  
gis  
ter  
Your  
Event  
Endpoint**

Step-by-step instructions to create an event destination, select event types, and enter the OPSCOM workflow.

|                                     |   |
|-------------------------------------|---|
| <b>Retrieve Your Signing Secret</b> | How to reveal and copy the <input type="text" value="whs_"/> secret, and destination ID into OPSCOM |
|-------------------------------------|---|

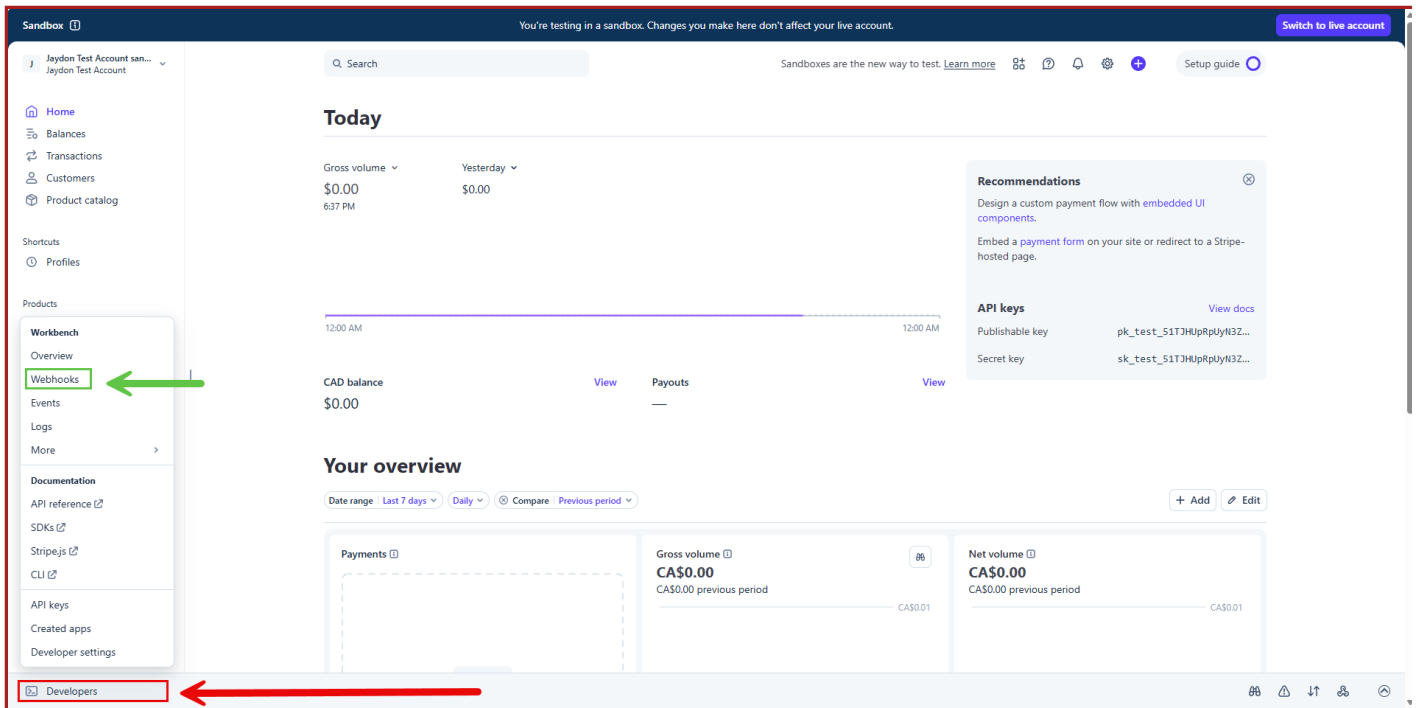
|                           |   |
|---------------------------|---|
| <b>Test Your Endpoint</b> | Stripe CLI commands to forward live test events locally before going live |
|---------------------------|---|

|   |   |
|---|---|
| <b>Se<br/>cur<br/>e<br/>Yo<br/>ur<br/>En<br/>dp<br/>oin<br/>t</b> | Sig<br>nat<br>ure<br>ver<br>ific<br>ati<br>on<br>exp<br>lan<br>ati<br>on<br>an<br>d<br>IP<br>allo<br>wli<br>stin<br>g<br>rec<br>om<br>me<br>nd<br>ati<br>on |
|---|---|

|            |      |
|------------|------|
| <b>Ma</b>  | Del  |
| <b>na</b>  | ive  |
| <b>ge</b>  | red  |
| <b>Ev</b>  | /Pe  |
| <b>en</b>  | ndi  |
| <b>t</b>   | ng/  |
| <b>De</b>  | Fail |
| <b>liv</b> | ed   |
| <b>ery</b> | sta  |
|            | tus  |
|            | des  |
|            | cri  |
|            | pti  |
|            | ons  |
|            | an   |
|            | d    |
|            | ho   |
|            | w    |
|            | to   |
|            | ma   |
|            | nu   |
|            | ally |
|            | res  |
|            | en   |
|            | d    |
|            | eve  |
|            | nts  |

## Register Your Endpoint

To begin receiving notifications, you must define where Stripe should send transaction data. In your Stripe Dashboard, click on **Developers** (bottom left) and select **Webhooks**.



Click the **Add Destination** button to open the configuration window.

You must select the specific event types the system needs to process. For a standard setup, ensure you select **payment\_intent.succeeded** for successful payments, **payment\_intent.payment\_failed** for failed payments, and **payment\_intent.canceled** for canceled payments. Now click Continue.

Payment Intent

8 events

Select all Payment Intent events

- `payment_intent.amount_capturable_updated`  
Occurs when a PaymentIntent has funds to be captured. Check the `amount_capturable` property on the PaymentIntent to determine the amount that can be captured. You may capture the PaymentIntent with an `amount_to_capture` value up to the specified amount. [Learn more about capturing PaymentIntents.](#)
- `payment_intent.canceled`  
Occurs when a PaymentIntent is canceled.
- `payment_intent.created`  
Occurs when a new PaymentIntent is created.
- `payment_intent.partially_funded`  
Occurs when funds are applied to a customer\_balance PaymentIntent and the 'amount\_remaining' changes.
- `payment_intent.payment_failed`  
Occurs when a PaymentIntent has failed the attempt to create a payment method or a payment.
- `payment_intent.processing`  
Occurs when a PaymentIntent has started processing.
- `payment_intent.requires_action`

Cancel

Continue →

Select **Webhook endpoint**, and click continue again.

## Choose where you want to send events

Stripe can send events to a webhook endpoint, Amazon EventBridge, or Azure Event Grid.

Destination type

### Webhook endpoint

Send webhook events to a hosted endpoint.



### Amazon EventBridge

Send events to your AWS account.



### Azure Event Grid

Send events to your Azure account.

Preview

Cancel

← Back

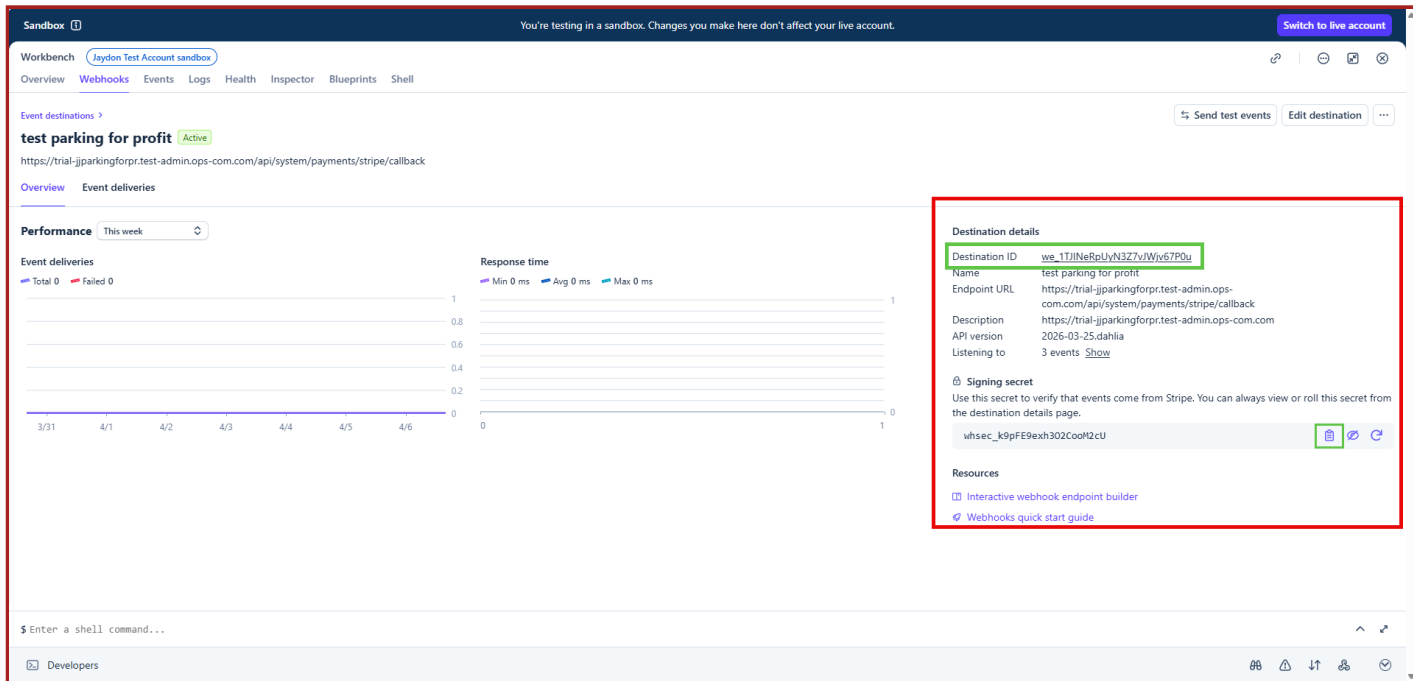
Continue →

Give the webhook a descriptive name, and in the **Endpoint URL** field, enter the specific URL provided by OPSCOM, in this format: `https://[CLIENTID].ops-com.com/api/system/payments/stripe/callback`

## Retrieve Your Signing Secret

Once the endpoint is created, Stripe generates a unique security key to verify that the data sent to your server is authentic. On the summary page for your

newly created webhook, locate the **Signing secret** section.



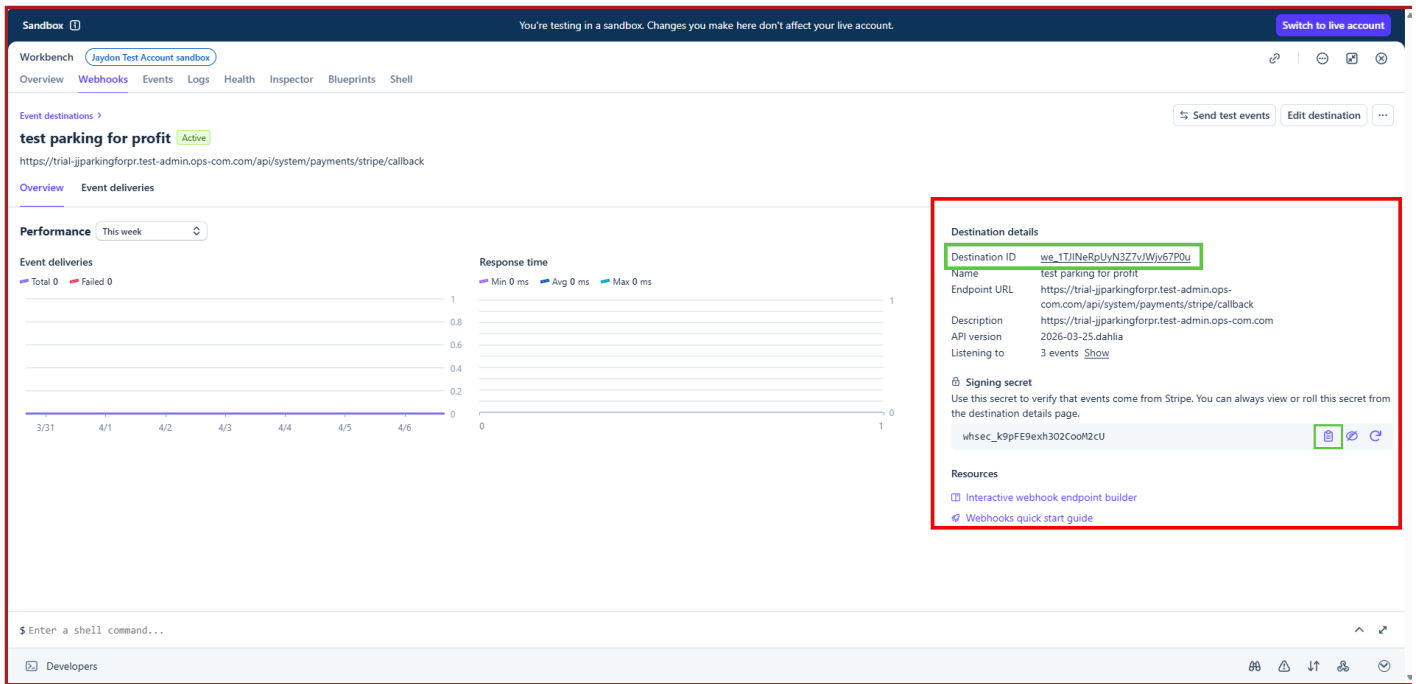
Click **Reveal** to display the string, which typically begins with the prefix

`whsec_`.

Copy (*highlighted in a green box in the picture above!*) this entire string and navigate back to your OPSCOM **Payment Gateway** settings. Paste the secret into the **Webhook Secret** field. This handshake ensures that OPSCOM only processes legitimate requests from Stripe and ignores any unauthorized attempts to alter transaction data.

Once the endpoint is saved, you'll see the **Signing secret**, and **Destination ID** provided by **Stripe**. Return to your **OPSCOM** configuration and paste the **Signing secret** into the **Webhook Secret** field, and the **Destination ID** into the **Webhook ID** field to ensure all incoming data is securely verified.

Ensure that your **Webhook Secret** is correctly matched between Stripe and OPSCOM to prevent "Pending" status errors on completed payments.



## User Experience and Handheld Use

### Web Portal Transactions

Users will interact with the Stripe interface during their standard checkout process. When they reach the final stage of a transaction, they simply click the **Pay with Credit Card** button to open the secure payment form. After they enter the credit card details into the form, they click the **Submit Payment** button to complete the purchase.

### Handheld App — *OPSCOM Parking Enforcement*

For administrators or staff using handheld units in the field, the process is optimized for touch interaction. Tap **Checkout** to open the payment summary within the app. From there, tap **Select Payment Method** and choose **Stripe**. To prevent accidental charges, the final step requires you to tap and hold the **Process Payment** button to finalize the transaction.

## Managing Transactions and Reports

Administrators can review all processed payments and manage financial returns through the standard reporting interfaces. Hover over **Financial Management** and click **Transaction Reports** then **Recent Payments** to view a live list of activity.

Within this report, the **Status** column displays the current state of each payment, while the **Payment Date** indicates exactly when the transaction occurred. If you need more detail, click the **View Details** icon next to a specific transaction to see the full Stripe receipt.

If a correction is needed, click the **Refund Payment** button to initiate a return of funds. You will then click the **Confirm Refund** drop-down menu and choose **Full Refund** to finalize the reversal.

You can quickly find a specific transaction by using the search bar to enter a user's name or a specific receipt order ID/Confirm Code.

---

## Best Practices and Considerations

- **Deployment Testing** Organizations should always keep **Live Mode** disabled during the initial deployment until all testing is verified with test credit card information. It is highly recommended to process a single, small test transaction in **Live Mode** using a real card before making the payment option available to the general public.
- **Security Requirements** Maintaining the security of your financial environment is critical. You should never share your **Stripe Secret Key** or **Webhook Secret** with anyone or store it in plain text outside of the secure administration configuration screens.

- **Change the signing secret periodically**, and only subscribe to required event types.