



# Pay By ClassWallet API Integration

## The Service

Pay by ClassWallet service allows an existing eCommerce store to integrate the ClassWallet digital checkout services into their customer ordering process. Once a vendor has integrated Pay by ClassWallet service into their checkout, any user with a ClassWallet account can shop and spend their funds through the vendor's eCommerce store.

## 3-Step Workflow

ClassWallet's integration follows a simple 3-step workflow whereby:

1. Account data is sent to your store to establish a session
2. The user is redirected to checkout with ClassWallet and order data is made available to ClassWallet
3. Order payment confirmation is sent to your store

## Checkout URL

The Checkout URL is constructed using your Callback URL (listed below) and your vendor ID.

### GET

`https://app.classwallet.com/payby-checkout/?callback=<your_callback>&vendorId=<your_vendor_id>`

Example: <https://app.classwallet.com/payby-checkout/?callback=https%3A%2F%2Fwww.store.com%2Fapi%2FCWOrder%2F0a4e569fd18e&vendorId=56b9fce8639d568e2535173d>

**NOTE:** The callback parameter must be URL encoded

## Endpoints

### Authentication

Both of the **Callback URL's** methods are protected by HTTP basic authentication. In your implementation of GET and PUT to the **Callback URL**, you enforce http basic authentication with the credentials issued to you during setup.

**IMPORTANT!** You *must* have basic auth in place before going live in the marketplace.



## Start URL

The Start URL is provided by you and needs to be communicated to ClassWallet to complete the setup process. This URL can be anything you like, just ensure that it is publicly accessible and is awaiting POST requests.

**POST** <https://yourstore.com/<your-start-url>>

Data is 'application/x-www-form-urlencoded' with one field named 'request' containing the JSON payload:

```
{
  "id": "55e4dfa966304b4d06b2bff3",
  "email": "ebutler@classwallet.com",
  "institution": "ClassWallet University West Campus",
  "username": "Test Butler",
  "shipping": {
    "address": "1 Magnum Pass",
    "city": "Mobile",
    "state": "AL",
    "zip": "36618"
  }
}
```

## Postman script for testing

```
{
  "info": {
    "_postman_id": "4cbfe18a-ad6b-4bf4-a983-6b9543170bbd",
    "name": "ClassWallet API",
    "schema": "https://schema.getpostman.com/json/collection/v2.0.0/collection.json",
    "_exporter_id": "19075874"
  },
  "item": [
    {
      "name": "Start API",
      "request": {
        "method": "POST",
        "header": [],
        "body": {
          "mode": "urlencoded",
          "urlencoded": [
            {
              "key": "request",
              "value": "{\n\"id\": \"55e4dfa966304b4d06b2bff3\", \"email\": \"ebutler@classwallet.com\", \"institution\": \"ClassWallet University West Campus\", \"username\": \"Test Butler\", \"shipping\": {\n\"address\": \"6100 Hollywood BLVD\", \"city\": \"Hollywood\", \"state\": \"CA\", \"zip\": \"90028\"}}}",
              "type": "text"
            }
          ]
        }
      }
    }
  ]
}
```



# CLASSWALLET

```
    }
  ]
},
"url": "https://classwallet-dev.classwallet.org/api/start"
},
"response": []
}
]
}
```

## Callback URL

The callback URL can take any form you like as long as it includes an order Id. Since it is a GET request, you can either include the order Id in the URL or as a query parameter. For this example, the callback URL is `https://yourstore.com/api/invoice_process/<order_id>`

**GET** `https://yourstore.com/api/invoice_process/<order_id>`

Example response:

```
{
  "orderId": "110383737",
  "details": {
    "items": [
      {
        "itemId": "1129392",
        "description": "90W Halo PAR38 Dimmable Flood Light ",
        "quantity": 2,
        "price": 9.97
      },
      {
        "itemId": "1536392",
        "description": "TCP 25W Daylight LED Light Bulb",
        "quantity": 1,
        "price": 20.98
      }
    ],
    "shipping": 5.99,
    "tax": 2.76
  }
}
```

**PUT** `https://yourstore.com/api/invoice_process/<order_id>`

Data is encoding: 'application/json'

```
{
  "PurchaseOrderId": "123456789",
  "status": "complete"
}
```



If everything is ok, respond with `{"status"=>"complete"}`  
or `{"status"=>"failed"}` if there was an issue.

OR

```
{  
  "PurchaseOrderId": "123456789",  
  "status": "canceled"  
}
```

Delete the order and respond with `{"status"=>"complete"}`

## Recommended 3-step Integration

Integration decisions are completely up to you; use these steps as a guideline as your particular implementation goals and environment may require a modified approach.

### Step 1

- Create an endpoint to receive the POST data from ClassWallet referred to as the **Start URL**
- When you receive data at the **Start URL**, store the data in a session for use later during checkout. You may choose to connect the user to an existing account via the email address. If you choose to do so, store the data in the session and login the appropriate user by matching email address. Finally, redirect the user to the shopping page of your site.
- When the user goes to checkout, use the session data to determine if the user came from ClassWallet, if so, suppress other payment methods and show only Checkout with ClassWallet accompanied with the ClassWallet thumbnail found [here](#).

### Step 2

- Once the user clicks the checkout button, create an order in your store and set the status to payment pending.
- After the order is created, redirect the user to ClassWallet by constructing a **Checkout URL** to complete payment. See section **Checkout URL** for details.
- Respond to the callback GET request. ClassWallet will issue a GET request to the callback URL you provided. Extract the order ID from the request and create the response per the section **Callback URL**.

### Step 3

- Respond to order callbacks from ClassWallet. Once the order has been either approved or rejected, ClassWallet will send a PUT request to your **Callback**



**URL.** Extract the order ID from the URL and purchase order ID from the body of the request. If the response indicates the order is complete, update the order accordingly and record the purchase order ID in the order notes, or as the transaction ID if applicable.