

Implementation Guide - Payment Connector + Example Site

This guide will walk you through the installation and implementation of the [Payment Connector](#), either locally or via the commercetools Marketplace. Additionally, you'll be able to explore its implementation using the [Example Site](#).

1. Project Setup

1.1. Create commercetools Project with Sample Data

Before you start, ensure you have a commercetools project with API client details. If you don't already have a project, follow these steps:


Steps:

1. Create a project with sample data (optional):

- Start a project with sample data.

Hi Aaron! Let's create a new project. ×

How would you like to begin?




- ☐ Start from scratch
Already have your own data ready?
Perfect, let's get you started with a blank project.
- ☒ Start with sample data
Don't want to upload data just yet?
Get started instantly with our sample data.

Continue

- We are selecting the B2C data for the guide.

Hi Aaron! Let's set up your project. ×

What type of data would you like?



- ☒ B2C: Lifestyle and Home
Explore sample orders, customer data, product details and categorization, inventory management, and more.
- ☐ B2B: Heavy Machinery
See how you can manage business units, quotes, and buyer approval flows among other features.

Back

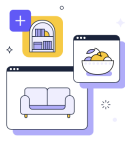
Continue

- Select the organization and name the project to be created.

Hi Aaron! Let's set up your project.

×

What type of data would you like?



☒ **B2C: Lifestyle and Home**

Explore sample orders, customer data, product details and categorization, inventory management, and more.

☐ **B2B: Heavy Machinery**

See how you can manage business units, quotes, and buyer approval flows among other features.

Back

Continue

2. API Client Configuration

You will need to create two API clients: one for the payment connector and another for the example site.

2.1. API Client for the Payment Connector

1. Create the API Client:

- Fill in the name and select the template **"Payment Connector for commercetools Checkout"**.

Manage <input type="checkbox"/> Approval flows <input type="checkbox"/> Approval rules <input type="checkbox"/> Associate roles <input type="checkbox"/> Attribute groups <input type="checkbox"/> Audit Log <input type="checkbox"/> Business units <input type="checkbox"/> Cart discounts <input type="checkbox"/> Categories <input checked="" type="checkbox"/> Checkout payment intents <input type="checkbox"/> Checkout sessions <input type="checkbox"/> Connectors <input type="checkbox"/> Connectors deployments <input type="checkbox"/> Customer groups <input type="checkbox"/> Customers <input type="checkbox"/> Discount codes <input type="checkbox"/> Extensions <input type="checkbox"/> Import Containers <input type="checkbox"/> Order Edits <input checked="" type="checkbox"/> Orders <input checked="" type="checkbox"/> Payments <input type="checkbox"/> Product Selections	View <input type="checkbox"/> Approval flows <input type="checkbox"/> Approval rules <input type="checkbox"/> Associate roles <input type="checkbox"/> Attribute groups <input type="checkbox"/> Audit Log <input type="checkbox"/> Business units <input type="checkbox"/> Cart discounts <input type="checkbox"/> Categories <input checked="" type="checkbox"/> Checkout sessions <input type="checkbox"/> Connectors <input type="checkbox"/> Connectors deployments <input type="checkbox"/> Customer groups <input type="checkbox"/> Customers <input type="checkbox"/> Discount codes <input type="checkbox"/> Import Containers <input type="checkbox"/> Messages <input type="checkbox"/> Order Edits <input checked="" type="checkbox"/> Orders <input checked="" type="checkbox"/> Payments <input type="checkbox"/> Product Selections <input type="checkbox"/> Products (all)	Manage My <input type="checkbox"/> Business units <input type="checkbox"/> Orders <input type="checkbox"/> Payments <input type="checkbox"/> Profile <input type="checkbox"/> Quote requests <input type="checkbox"/> Quotes <input type="checkbox"/> Shopping Lists	Tokens <input type="checkbox"/> Create Anonymous Token <input checked="" type="checkbox"/> Introspect OAuth Tokens <input type="checkbox"/> Manage API clients <input checked="" type="checkbox"/> View API clients
--	--	--	--

2. Connector .env example:

```
CTP_PROJECT_KEY=connector-plus-example-site
CTP_CLIENT_SECRET=QBuw60PQ3KBkzSD7iLjdQzQT-raqCqU4
CTP_CLIENT_ID=tc9sUh8VukwevJZ7eB0Aujan
CTP_AUTH_URL=https://auth.us-central1.gcp.commercetools.com
CTP_API_URL=https://api.us-central1.gcp.commercetools.com
CTP_SCOPES=manage_payments:connector-plus-example-site
view_sessions:connector-plus-example-site introspect_oauth_tokens:connector-plus-example-site
view_api_clients:connector-plus-example-site
manage_checkout_payment_intents:connector-plus-example-site
manage_orders:connector-plus-example-site
```

2.2. API Client for the Example Site

1. Create the API Client:

- Fill in the name and select the template **"Frontend B2C"**.

Manage	View	Manage My	Tokens
<input type="checkbox"/> Approval flows	<input type="checkbox"/> Approval flows	<input type="checkbox"/> Business units	<input checked="" type="checkbox"/> Create Anonymous Token
<input type="checkbox"/> Approval rules	<input type="checkbox"/> Approval rules	<input type="checkbox"/> Orders	<input type="checkbox"/> Introspect OAuth Tokens
<input type="checkbox"/> Associate roles	<input type="checkbox"/> Associate roles	<input type="checkbox"/> Payments	<input type="checkbox"/> Manage API clients
<input type="checkbox"/> Attribute groups	<input type="checkbox"/> Attribute groups	<input type="checkbox"/> Profile	<input type="checkbox"/> View API clients
<input type="checkbox"/> Audit Log	<input type="checkbox"/> Audit Log	<input type="checkbox"/> Quote requests	
<input type="checkbox"/> Business units	<input type="checkbox"/> Business units	<input type="checkbox"/> Quotes	
<input type="checkbox"/> Cart discounts	<input checked="" type="checkbox"/> Cart discounts	<input type="checkbox"/> Shipping Lists	
<input type="checkbox"/> Categories	<input checked="" type="checkbox"/> Categories		
<input type="checkbox"/> Checkout payment intents	<input checked="" type="checkbox"/> Checkout sessions		
<input type="checkbox"/> Checkout sessions	<input type="checkbox"/> Connections		
<input type="checkbox"/> Connectors	<input type="checkbox"/> Connections deployments		
<input type="checkbox"/> Connectors deployments	<input type="checkbox"/> Customer groups		
<input type="checkbox"/> Customer groups	<input checked="" type="checkbox"/> Customers		
<input checked="" type="checkbox"/> Customers	<input checked="" type="checkbox"/> Discount codes		
<input type="checkbox"/> Discount codes	<input type="checkbox"/> Import Containers		
<input type="checkbox"/> Extensions	<input type="checkbox"/> Messages		
<input type="checkbox"/> Import Containers	<input type="checkbox"/> Order Edits		
<input checked="" type="checkbox"/> Order Edits	<input checked="" type="checkbox"/> Orders		
<input checked="" type="checkbox"/> Orders	<input type="checkbox"/> Payments		
<input type="checkbox"/> Payments	<input checked="" type="checkbox"/> Product Selections		
<input type="checkbox"/> Product Selections	<input checked="" type="checkbox"/> Products (all)		
<input type="checkbox"/> Products	<input checked="" type="checkbox"/> Products (published)		
<input type="checkbox"/> Project	<input checked="" type="checkbox"/> Project Settings		
<input type="checkbox"/> Project Settings	<input type="checkbox"/> Quote requests		
<input type="checkbox"/> Quote requests	<input type="checkbox"/> Quotes		
<input type="checkbox"/> Quotes	<input checked="" type="checkbox"/> Shipping methods		
<input type="checkbox"/> Shipping methods	<input type="checkbox"/> Shipping Lists		
<input checked="" type="checkbox"/> Shopping Lists	<input type="checkbox"/> Staged quotes		
<input checked="" type="checkbox"/> Staged quotes	<input checked="" type="checkbox"/> Standalone Prices		
<input type="checkbox"/> Standalone Prices	<input type="checkbox"/> States		
<input type="checkbox"/> States	<input type="checkbox"/> Stores		
<input type="checkbox"/> Stores	<input checked="" type="checkbox"/> Tax categories		
<input type="checkbox"/> Subscriptions	<input checked="" type="checkbox"/> Types		
<input type="checkbox"/> Tax categories			
<input type="checkbox"/> Types			

2. Example Site .env example:

```
CTP_PROJECT_KEY=connector-plus-example-site
CTP_CLIENT_SECRET=rt3wSXHqsuZllG_0fwQgtdif7Wt1h-P_
CTP_CLIENT_ID=yyujtVMw7FusXqPbj5QpYLkx
CTP_AUTH_URL=https://auth.us-central1.gcp.commercetools.com
CTP_API_URL=https://api.us-central1.gcp.commercetools.com
CTP_SCOPES=view_categories:connector-plus-example-site
create_anonymous_token:connector-plus-example-site
view_products:connector-plus-example-site
manage_staged_quotes:connector-plus-example-site
manage_order_edits:connector-plus-example-site
view_tax_categories:connector-plus-example-site
view_standalone_prices:connector-plus-example-site
view_published_products:connector-plus-example-site
view_project_settings:connector-plus-example-site
manage_orders:connector-plus-example-site view_sessions:connector-plus-example-site
view_cart_discounts:connector-plus-example-site
manage_customers:connector-plus-example-site
manage_shopping_lists:connector-plus-example-site
view_discount_codes:connector-plus-example-site view_types:connector-plus-example-site
view_product_selections:connector-plus-example-site
view_shipping_methods:connector-plus-example-site
```

3. Stripe Account Setup and Webhooks

To proceed, you'll need a Stripe account and a webhook endpoint to process payment events.

3.1. Required Stripe Information

1. Account ID:

- Navigate to **Settings > Business > Account details**.
- Example: `acct_1P6Fr2Aw0TvR8PUy`

2. Publishable Key:

- Navigate to **Developers > API keys > Publishable key**.
- Example:
`pk_test_51P6F3kLPATvR8PUyWBp3jqT231GEKq51LC5JvLQi52mQm1rPTor0BdBsDMDqx0gZ614dbHEuVafo207cU6b008fjUMUYG`

3. Stripe Private Key:

- Navigate to **Developers > API keys > Secret key**.
- Example:
`sk_test_51P6F3kLPATv3tF3DLvA1VwxkTk1ANsDGWj6UUTSI2SYaS0RkfTbjFJG5MoEggQ5e3Y0N03F0vkUNj0vMN3T9v7wxDM400zYFFrx6l`

4. Webhooks ID:

- This value is retrieved after creating the webhooks endpoint.
- Example: `we_1p4qIaLPATvR8PUyhuW6B3E6`

5. Webhooks Signing:

- This value is retrieved after creating the webhooks endpoint.
- Example: `whsec_sBvf8NHTAA8b4RzF4hApAwNtLYEohEDn`

3.2. Webhooks Endpoint URL Setup

If you're running the Payment Connector locally, you'll need to use NGROK on your machine to create a secure gateway that listens for and processes webhook events from Stripe. This allows you to effectively test and handle Stripe's webhook events during development.

Steps:

1. Create the NGROK gateway on your local machine:

- Command: `ngrok http 8080`
- Example URL: `https://fb93-104-129-159-229.ngrok-free.app`

2. Create the Webhooks endpoint in Stripe:

- Select the events to be used in the Payment Connector processor:
 - `charge.refunded`
 - `charge.succeeded`
 - `payment_intent.canceled`
 - `payment_intent.payment_failed`
 - `payment_intent.requires_action`
 - `payment_intent.succeeded`

3. Fill in the live "Endpoint URL":

- Use the created URL with NGROK and add `/stripe/webhooks` to the end.
- Example: `https://fb94-14-169-149-229.ngrok-free.app/stripe/webhooks`

4. Run the Connector

4.1. Locally

To run the connector locally, configure the respective `.env` files for the processor and the enabler.

4.1.1. Example of the `.env` Processor File:

```
CTP_AUTH_URL=https://auth.us-central1.gcp.commercetools.com
CTP_API_URL=https://api.us-central1.gcp.commercetools.com
CTP_SESSION_URL=https://session.us-central1.gcp.commercetools.com
CTP_CLIENT_ID=tc9sUh8VukwevJZ7eB0Aujan
CTP_CLIENT_SECRET=QBuW60PQ3KBkzSD7iLjdQzQT-raqCqU4
CTP_PROJECT_KEY=connector-plus-example-site
CTP_JWKS_URL=http://localhost:9000/jwt/.well-known/jwks.json
CTP_JWT_ISSUER=https://mc-api.us-central1.gcp.commercetools.com
MERCHANT_RETURN_URL=[Merchant-website-return-url]
STRIPE_SECRET_KEY=sk_test_51P6F3kLPATv3tF3DLvA1VwxkTk1ANsDGWj6UUTSI2SYaS0Rkf
TbjFJG5MoEggQ5e3Y0N03F0vkUNj0vMN3T9v7wxDM400zYFFrx6l
STRIPE_CAPTURE_METHOD=automatic
STRIPE_APPEARANCE_PAYMENT_ELEMENT=
STRIPE_APPEARANCE_EXPRESS_CHECKOUT=
STRIPE_WEBHOOK_SIGNING_SECRET=whsec_sBvf8NHTAA8b4RzF4hApAwNtLYEohEDn
STRIPE_WEBHOOK_ID=we_1p4qIaLPATvR8PUyhuW6B3E6
```

4.1.2. Example of the `.env` Enabler File:

```
VITE_CTP_AUTH_URL=https://auth.us-central1.gcp.commercetools.com
VITE_CTP_API_URL=https://api.us-central1.gcp.commercetools.com
VITE_CTP_SESSION_URL=https://session.us-central1.gcp.commercetools.com
VITE_CTP_CLIENT_ID=tc9sUh8VukwevJZ7eB0Aujan
VITE_CTP_CLIENT_SECRET=QBuW60PQ3KBkzSD7iLjdQzQT-raqCqU4
VITE_CTP_PROJECT_KEY=connector-plus-example-site
VITE_PROCESSOR_URL=http://localhost:8080
VITE_STRIPE_PUBLISHABLE_KEY=pk_test_51P6F3kLPATvR8PUyWBp3jqT231GEKq51LC5JvLQ
i52mQm1rPTor0BdBsDMDqx0gZ614dbHEuVafo207cU6b008fjUMUYG
```

4.1.3. Run the Command:


- Processor:
 1. Install dependencies:
 - `npm install`
 2. Build the application:
 - `npm run build`
 3. Run the processor:
 - `npm run start`
- Enabler:
 1. Install dependencies:
 - `npm install`
 2. Build the application:
 - `npm run build`
 3. Run the enabler:
 - `npm run start`

This process will enable the JavaScript module to be implemented and listen to any webhooks event through the NGROK URL created on your local machine.

4.2. commercetools Merchant Center Marketplace

4.2.1. Install the connector to the project connect dashboard

1. Navigate to “Manage Organizations & Teams” in the commercetools dashboard.

**Aaron Meza**
aaron.meza@gluo.mx

[My Profile](#)
[Manage organizations & teams](#)
[Manage projects](#)

[Privacy Policy](#)
[Support](#)

[Logout](#)

2. Select your organization, go to “Connect”.

[Teams](#) [Settings](#) [Connect](#) [Custom Applications](#) [Custom Views](#)

Teams 1 results
Select a team from the table below to invite members and manage permissions.

Name	Members
Administrators	4 members


3. Search for "Stripe" and click the "Install" button.

[Settings](#) [Connect](#) [Custom Applications](#) [Custom Views](#)

Discover all available connectors

[Public connectors](#) [Organization connectors](#)

[Filter by](#)

**stripe**
Stripe Payment for
Composable Commerce
Stripe payment connector for
commercetools Composable
Commerce.
[Install](#) [Learn more](#)

4. Choose the region and project to install the connector.

Installing Stripe Payment for Composable Commerce

1

commercetools configuration

2

Stripe Payment for Composable Commerce configuration

In which region and organization do you want to install the connector?

Region *

North America (Google Cloud)

Organization

Aaron

Now, select in which projects you want to install Stripe Payment for Composable Commerce.

☐ Install for all projects of this organization

☒ Install for selected projects

Select projects

1 selected ✕ Deselect all

Stripe Testing 3

stripe-testing-3

5. Complete the configuration required to run the connector.

- CTP_PROJECT_KEY: The key of commercetools composable commerce project.
- CTP_SCOPE: The scope constrains the endpoints to which the commercetools client has access, as well as the read/write access right to an endpoint.
- CTP_AUTH_URL: The URL for authentication in commercetools platform. It is used to generate OAuth 2.0 token which is required in every API call to commercetools composable commerce. The default value is `https://auth.europe-west1.gcp.commercetools.com`. For details, please refer to documentation [here](#).
- CTP_API_URL: The URL for commercetools composable commerce API. Default value is `https://api.europe-west1.gcp.commercetools.com`.
- CTP_SESSION_URL: The URL for session creation in commercetools platform. Connectors relies on the session created to be able to share information between enabler and processor. The default value is `https://session.europe-west1.gcp.commercetools.com`.
- CTP_JWKS_URL: The URL which provides JSON Web Key Set (set the region of your project): <https://mc-api.{region}.commercetools.com/.well-known/jwks.json>
- CTP_JWT_ISSUER: The issuer inside JSON Web Token which is required in JWT validation process (set the region of your project): <https://mc-api.{region}.commercetools.com>
- STRIPE_CAPTURE_METHOD: Stripe capture method (manual or automatic), default value: automatic.

- `STRIPE_APPEARANCE_PAYMENT_ELEMENT`: Stripe Elements supports visual customization, which allows you to match the design of your site with the `appearance` option. This value has the specific appearance of the Payment Element component.
- `STRIPE_APPEARANCE_EXPRESS_CHECKOUT`: Stripe Elements supports visual customization, which allows you to match the design of your site with the `appearance` option. This value has the specific appearance of the Express Checkout Element component.
- `CTP_CLIENT_SECRET`: The client secret of commercetools composable commerce user account. It is used in commercetools client to communicate with commercetools composable commerce via SDK.
- `CTP_CLIENT_ID`: The client ID of your commercetools composable commerce user account. It is used in commercetools client to communicate with commercetools composable commerce via SDK.
- `STRIPE_SECRET_KEY`: Stripe authenticates your API requests using your account's API keys
- `STRIPE_WEBHOOK_ID`: Stripe unique identifier for the [Webhook Endpoints](#)
- `STRIPE_WEBHOOK_SIGNING_SECRET`: Stripe Secret key to verify webhook signatures using the official libraries. This key is created in the [Stripe dashboard Webhook](#).

Installing Stripe Payment for Composable Commerce

- ☒ commercetools configuration
 ☒ 2 Stripe Payment for Composable Commerce configuration

Now, let's configure the connector in the configured projects

In this step, you have to enter the information the connector requires to configure it within the project. If you've selected several projects, you need to configure the connector in each. For more information, you can [consult this article](#).

Selected projects (1)

☐ Stripe Testing 3 >

processor configuration

CTP_API_URL *

Commercetools API URL

CTP_AUTH_URL *

Commercetools Auth URL

CTP_JWKS_URL *

JWKS url

CTP_JWT_ISSUER *

JWT Issuer for jwt validation

4.2.2. Retrieve Payment Connector deployment URL

After installation, you can access the Managed Connectors dashboard to monitor the installation status.

Manage connectors

Organization connectors		Installations					
Filter by type		Select connector type ...					
Connector	Project name	Deployment key	Status	Deployment type	Date installed	Region	Build details
Stripe Payment for Composable Commerce	Stripe Testing 3		Installing	Production	8/27/24, 12:18 PM	North America	

Once the installation is complete, you can view the application type, the configuration details you provided, and the deployment URL by selecting the Connector project. Be sure to copy and save the deployment URLs for the enabler and processor applications.

Stripe Payment connector - stripe-testing-3

Installed North America

[Apps](#)

[App logs](#)

[Build details](#)

[Uninstall](#)

Stripe Payment connector apps (2)

- ☒ enabler >
- ☐ processor >

enabler

Application type

assets

URL

https://assets-31f4a21e-1a79-4ffc-a255-31c449660a

[Apps](#)

[App logs](#)

[Build details](#)

[Uninstall](#)

After saving, the secret is no longer visible for security reasons, but is still active unless you overwrite it with a new value.

Commercetools client secret

.....

STRIPE_SECRET_KEY *

After saving, the secret is no longer visible for security reasons, but is still active unless you overwrite it with a new value.

Stripe secret key

.....

STRIPE_WEBHOOK_SIGNING_SECRET *

After saving, the secret is no longer visible for security reasons, but is still active unless you overwrite it with a new value.

Stripe Webhook signing secret

.....

Application type

service

URL

https://service-14a18a7f-c667-4018-af23-ca4fa961f

4.2.3. Update configurations

If we need to modify some values from the connector configuration, we need to enter to the Managed Connectors dashboard and select the connector we want to update.

After modifying the configuration values, you'll see a 'Redeploy' button. Click it to redeploy the connector and apply the updated values.

Stripe Payment connector - stripe-testing-3

Installed North America

Apps App logs Build details

Stripe Payment connector apps (2)

- ✓ enabler >
- ✓ processor >

processor

CTP_API_URL *
Commercetools API URL
https://api.us-central1.gcp.commercetools.com

CTP_AUTH_URL *
Commercetools Auth URL
https://auth.us-central1.gcp.commercetools.com

CTP_JWKS_URL *
JWKS url
https://mc-api.us-central1.gcp.commercetools.com/.

CTP_JWT_ISSUER *
JWT Issuer for jwt validation
https://mc-api.us-central1.gcp.commercetools.com

CTP_PROJECT_KEY *
Commercetools project key
stripe-testing-3-test

Update values

Revert changes Redeploy

4.2.4. Connector Logs

To check the logs of an application, go to the Managed Connectors dashboard, select the app, and then navigate to the 'App Logs' tab. Here, you can view the logs generated by the

application.

Stripe Payment for Composable Commerce - Stripe Testing 3

Installed North America

Apps App logs Build details

Uninstall

Stripe Payment for Composable Commerce apps (2)

enabler >

processor >

Last 2 days

08/25/2024 12:30 PM

08/27/2024 12:30 PM

Default
8/27/2024, 12:22:57 PM
{
 "message": "",
 "payload": {
 "pid": 17,
 "hostname": "localhost",
 "msg": "Server listening at http://0.0.0.0:8080",
 "time": 1724782976859
 }
}

Info
8/27/2024, 12:22:55 PM
{
 "message": "[object Object]",
 "payload": {
 "severity": "info",
 "message": "[object Object]"
 }
}

Default

5. Run the Example Site Locally

To run the site locally, configure the `.env` files in the main folder and the client folder.

5.1. Example of the Configuration in the Main Folder:

Variable	Description	Localhost e.g.
REACT_APP_SK	Stripe Private Key	sk_test_51P6F3kLPATv3tF3DLvA1Vw
REACT_APP_ADMIN	Stripe Account ID	acct_1P6Fr2AwOTvR8PUy

Variable	Description	Localhost e.g.
REACT_APP_PORT	Port to run the server	8081
REACT_APP_BASE_URL	Base URL of the server	http://localhost:8081
REACT_APP_CT_PROJECT_KEY	commercetools Project Key	connector-plus-example-site
REACT_APP_CT_CLIENT_ID	commercetools Client ID	tc9sUh8VukwevJZ7eB0Aujan
REACT_APP_CT_SECRET	commercetools Secret Key	QBuW6OPQ3KBkzSD7iLjdQzQT-raqC
REACT_APP_CT_API_URL	commercetools API URL	https://api.us-central1.gcp.commercetools.com
REACT_APP_CT_AUTH_URL	commercetools Auth URL	https://auth.us-central1.gcp.commercetools.com

5.2. Example of the Configuration in the Client Folder:

Variable	Description	Localhost e.g.
REACT_APP_PK	Stripe Publishable Key	pk_test_51P6F3kLPATvR8PUyWB
REACT_APP_BASE_URL	Base URL of the server example site	http://localhost:8081
REACT_APP_PROJECT_KEY	commercetools Project Key	connector-plus-example-site
REACT_APP_CTP_CLIENT_ID	commercetools Client ID	tc9sUh8VukwevJZ7eB0Aujan
REACT_APP_CLIENT_SECRET	commercetools Secret Key	QBuW6OPQ3KBkzSD7iLjdQzQT-r
REACT_APP_SESSION_URL	commercetools API session url	https://api.us-central1.gcp.commercetools.com
REACT_APP_AUTH_URL	commercetools Auth URL	https://auth.us-central1.gcp.commercetools.com
REACT_APP_ENABLER_BUILD_URL	URL of the Connector enabler	local - http://localhost:3000/connector-enabler Marketplace - https://assets-mba1.gcp.commercetools.com

Variable	Description	Localhost e.g.
	deployed in commercetools or local	
REACT_APP_PROCESSOR_URL	URL of the Connector Processor deployed in commercetools or local	local - http://localhost:8080 Marketplace - https://service-mwze
REACT_APP_MODE	Flag to add header to Ngrok CORS	dev

5.3. Instructions to Run the Example Site

1. Install dependencies:

- Run `npm install` in both the main and client folders to install the necessary dependencies.

2. Start the server:

- In the main folder, run `npm run build && npm run start` to start the backend server.
- In the client folder, run `npm run start` to start the frontend development server.

3. Navigate to the Example Site:

- Open your browser and go to `http://localhost:3006` | `http://localhost:3000`.
- You should see the Example Site running locally, allowing you to interact with the integration.

6. Testing and Debugging

6.1. Testing the Integration

1. Place an Order:

- Go to the Example Site and add items to the cart.
- Proceed to checkout and complete the payment process.

2. Complete the Payment:

- Use Stripe's test card numbers for testing (e.g., 4242 4242 4242 4242 for a successful payment).

3. Verify the Order in commercetools:

- Log in to your commercetools account and check that the order has been created and that the payment status reflects the correct state.

6.2. Debugging Common Issues

1. Webhooks Events Not Received:

- Ensure that your NGROK tunnel is active and that the URL configured in Stripe's webhooks settings matches the current NGROK URL.

2. Authentication Issues:

- Double-check that the API client secrets and IDs are correctly set in the .env files for both the connector and the example site.

3. Stripe Payment Errors:

- Verify that the Stripe keys are correct and that you're using the appropriate environment (test or live) for the keys.
-
-

8. Conclusion

This guide provides comprehensive steps to implement and test a payment connector using commercetools and Stripe. By following the instructions, you can set up a local environment, thoroughly test the integration. Ensure that all environment variables and API keys are kept secure, and regularly monitor your deployed systems to catch any issues early.

Note: For further information or troubleshooting, refer to the official documentation for [commercetools](#) and [Stripe](#), or seek help from the respective support communities.