

# **Magento 2 ACCESS WORLDPAY GraphQL API Guide**

---

July 2022

## Contents

<b>Introduction .....</b>	<b>4</b>
Before you start .....	4
If you need support .....	5
Overview of Worldpay Plugin GraphQL Support .....	5
How to get started .....	5
Worldpay Payment GraphQL .....	11
<i>Set the Payment method using a new card .....</i>	<i>11</i>
<i>Query to retrieve all saved cards for the shopper .....</i>	<i>12</i>
<i>Set the Payment method using the saved card (token only) .....</i>	<i>12</i>
<i>Set Payment method using the saved card with cvc enabled .....</i>	<i>13</i>
<i>Set Payment method using the saved card with cvc enabled (cvcHref) .....</i>	<i>13</i>
<i>Set Payment method for new card using token URL (tokenHref) .....</i>	<i>14</i>
GraphQL: List/Update/Delete Credit Cards .....	15
Instant Purchase .....	17
Google Pay .....	17
<i>Query to retrieve Google Pay Settings .....</i>	<i>17</i>
<i>Set Payment method with Google Pay wallet .....</i>	<i>18</i>
<i>Sample Google Pay Token .....</i>	<i>19</i>
Apple Pay .....	19
<i>Query to retrieve Apple Pay Settings .....</i>	<i>19</i>
<i>Set Payment method with Apple Pay wallet .....</i>	<i>20</i>
<i>Sample Apple Pay Token .....</i>	<i>21</i>
Configurable Alert Messages .....	21
<i>Query to retrieve the existing Configurable Alert Messages .....</i>	<i>21</i>
Configurable Labels .....	23

*Query to retrieve the existing Configurable Labels* ..... 23

*References:* ..... 28

## Introduction

This document explains the step by step process to utilize the Worldpay payment gateway for making payments.

### Before you start

**Prerequisite:** For the GraphQL introduction please visit the [GraphQL developer guide](#). You can use any API collaboration platform like POSTMAN to check the Magento or Worldpay exposed GraphQL endpoints.

For making requests to Magento through GraphQL, please visit the [guide](#).

GraphQL endpoints to be executed in order	Description	Magento native/ Worldpay customized
<b>Create Customer</b>	One time activity to register new shopper with the <a href="#">createCustomer</a> mutation. Once registered, you can create an authenticated token for existing shoppers using <a href="#">generateCustomerToken</a> mutation.  To send requests on behalf of the shopper, you must supply the generated token as a header in your GraphQL browser. The name of the header is 'Authorization' and the value is 'Bearer <token>'.	Native
<b>Create an empty cart</b>	Create customer cart (customerCart) for registered shoppers or use <a href="#">createemptycart</a> mutations for guest shoppers.	Native
<b>Mapping email with cart for guest shopper</b>	The card ID and the email ID mapping is done using the mutation <a href="#">setGuestEmailOnCart</a> .	Native
<b>Add products to cart</b>	Add product using <a href="#">addSimpleProductsToCart</a> and modify the cart ID as generated by create empty cart mutation.	Native
<b>Set the shipping address</b>	<a href="#">setShippingAddressesOnCart</a> mutation to be used with appropriate cart ID.	Native
<b>Set the delivery method</b>	Use the <a href="#">setShippingMethodsOnCart</a> mutation to set a delivery method.	Native
<b>Set the billing address</b>	Use the <a href="#">setBillingAddressOnCart</a> mutation to set a billing address.	Native
<b>Set the payment method</b>	Use the <a href="#">cart</a> query to determine which payment methods are available for your order. The Worldpay payment method specific mutation are explained under their respective sections.	Worldpay customized payment types
<b>Place the order</b>	The <a href="#">placeOrder</a> mutation places an order when cart ID is specified.	Native

## If you need support

Please get in touch with your Worldpay support contact. If they're not available, call our customer service team on 0800 096 3997.

## Overview of Worldpay Plugin GraphQL Support

The table below lists the main features of our plugin support on GraphQL.

Features	Description
<b>Credit Card integration</b>	GraphQL support for save card, list cards, update card, delete card, place order with new or saved(tokenized) credit card.
<b>Instant Purchase</b>	Make the worldpayTokens query if the shopper is eligible for Instant Purchase, and make the payment with a saved card token.
<b>Google Pay</b>	Query to retrieve the Google Pay settings and set the payment method using Google pay.
<b>Apple Pay</b>	Query to retrieve the Apple Pay settings and set the payment method using Apple pay.

## How to get started

In headers under Authorization: put the customer token for registered and logged in users.

For example: Authorization: Bearer 3lodst49pvg7n2d2fmyyjl06kr8k6jf

1. Create an empty cart for a registered user.

See: <https://devdocs.magento.com/guides/v2.4/graphql/mutations/create-empty-cart.html>

Sample Request	Sample Response
<pre>{   customerCart{     id   } }</pre>	<pre>{   "data": {     "customerCart": {       "id": "sG1FYDc4oCdsXrU6RQyz1Y5aAhTrv3w8"     }   } }</pre>

2. Create an empty cart for a guest user.

See: <https://devdocs.magento.com/guides/v2.4/graphql/mutations/create-empty-cart.html>

Sample Request	Sample Response
<pre>mutation {   createEmptyCart</pre>	<pre>{   "data": {</pre>

<pre> }</pre>	<pre>       "createEmptyCart": "oOZjoRvISf7emel0AQpkUg9AwhluEylR"     }   } }</pre>
---------------	---

Map the guest user email to guest user cart created above

See: <https://devdocs.magento.com/guides/v2.4/graphql/mutations/set-guest-email.html>

Sample Request	Sample Response
<pre> mutation {   setGuestEmailOnCart(     input: {       cart_id: " oOZjoRvISf7emel0AQpkUg9AwhluEylR"       email: "jdoe@example.com"     }   ) {     cart {       email     }   } }</pre>	<pre> {   "data": {     "setGuestEmailOnCart": {       "cart": {         "email": "jdoe@example.com"       }     }   } }</pre>

3. Add a product to the cart.

See: <https://devdocs.magento.com/guides/v2.4/graphql/mutations/add-products-to-cart.html>

Sample Request	Sample Response
<pre> mutation {   addSimpleProductsToCart(     input: {       cart_id: "sG1FYDc4oCdsXrU6RQyzlY5aAhTrv3w8"       cart_items: [         {           data: {             quantity: 1             sku: "24-MB02"           }         }       ]     }   ) {     cart {       items {         id         product {           name           sku         }         quantity       }     }   } }</pre>	<pre> {   "data": {     "addSimpleProductsToCart": {       "cart": {         "items": [           {             "id": "57",             "product": {               "name": "Fusion Backpack",               "sku": "24-MB02"             },             "quantity": 2           }         ]       }     }   } }</pre>



```

    }
  }
}

```

4. Add a shipping address.

See: <https://devdocs.magento.com/guides/v2.4/graphql/mutations/set-shipping-address.html>

Sample Request	Sample Response
<pre> mutation {   setShippingAddressesOnCart(     input: {       cart_id: "sG1FYDc4oCdsXrU6RQyzlY5aAhTrv3w8"       shipping_addresses: [         {           address: {             firstname: "Jacob"             lastname: "Hunt"             company: "Magento"             street: ["Magento Pkwy", "Main Street"]             city: "Austin"             region: "TX"             postcode: "78758"             country_code: "US"             telephone: "8675309"             save_in_address_book: true           }         }       ]     }   ) {     cart {       shipping_addresses {         firstname         lastname         company         street         city         postcode         telephone       }     }   } } </pre>	<pre> {   "data": {     "setShippingAddressesOnCart": {       "cart": {         "shipping_addresses": [           {             "firstname": "Jacob",             "lastname": "Hunt",             "company": "Magento",             "street": [               "Magento Pkwy",               "Main Street"             ],             "city": "Austin",             "postcode": "78758",             "telephone": "8675309"           }         ]       }     }   } } </pre>

5. Set the shipping method.

See: <https://devdocs.magento.com/guides/v2.4/graphql/mutations/set-shipping-method.html>

Sample Request	Sample Response
<pre> mutation {   setShippingMethodsOnCart(     input: {       cart_id: "sG1FYDc4oCdsXrU6RQyzlY5aAhTrv3w8",       shipping_methods: [         {           carrier_code: "tablerate"           method_code: "bestway"         }       ]     }   ) {     cart {       shipping_addresses {         firstname         lastname         company         street         city         postcode         telephone       }     }   } } </pre>	<pre> {   "data": {     "setShippingMethodsOnCart": {       "cart": {         "shipping_addresses": [           {             "selected_shipping_method": {               "carrier_code": "tablerate", </pre>

<pre>         }       ]     }   ) {     cart {       shipping_addresses {         selected_shipping_method {           carrier_code           carrier_title           method_code           method_title           amount {             value             currency           }         }       }     }   } } </pre>	<pre>       "carrier_title": "Best Way",       "method_code": "bestway",       "method_title": "Table Rate",       "amount": {         "value": 0,         "currency": "USD"       }     }   } ] } } } } </pre>
---	---

6. Add the billing address.

See: <https://devdocs.magento.com/guides/v2.4/graphql/mutations/set-billing-address.html>

Sample Request	Sample Response
<pre> mutation {   setBillingAddressOnCart(     input: {       cart_id: "sG1FYDc4oCdsXrU6RQyzlY5aAhTrv3w8"       billing_address: {         address: {           firstname: "Jacob"           lastname: "Hunt"           company: "Magento"           street: ["Magento Pkwy", "Main Street"]           city: "Austin"           region: "TX"           postcode: "78758"           country_code: "US"           telephone: "8675309"           save_in_address_book: true         }         use_for_shipping: true       }     }   ) {     cart {       billing_address {         firstname         lastname         company         street         city         postcode         telephone       }     }   } } </pre>	<pre> {   "data": {     "setBillingAddressOnCart": {       "cart": {         "billing_address": {           "firstname": "Jacob",           "lastname": "Hunt",           "company": "Magento",           "street": [             "Magento Pkwy",             "Main Street"           ],           "city": "Austin",           "postcode": "78758",           "telephone": "8675309"         }       }     }   } } </pre>



## 7. Set the payment method.

Worldpay provides a customized mutation (“**setPaymentMethodOnCart**”) to set the payment method before the order is placed.

A single mutation (“**setPaymentMethodOnCart**”) with various inputs/query is used and described below.

Worldpay GraphQL endpoints	Description
<b>Set the Payment method using a new card</b>	<p><u>Mutation</u> - <b>setPaymentMethodOnCart</b></p> <p><u>Pre requisite</u> – Set the Worldpay Integration Mode to Direct.</p> <p><u>Inputs</u> -</p> <p><b>Code</b> :“worldpay_cc”, <b>cc_name</b> – credit card holder name, <b>cc_number</b>- credit card number, <b>cc_exp_month</b>- credit card expiry month, <b>cc_exp_year</b>- credit card expiry year, <b>cvc</b>- card verification code, <b>save_card</b> – an option for registered user to save their card(Accepted values “0” or “1”).</p> <p>All other input values of “<b>setPaymentMethodOnCart</b>” should be passed empty strings”.</p>
<b>Query to retrieve all the saved cards for the shopper</b>	<p><u>Query</u> - <b>worldpayTokens</b></p> <p>Before an order is placed with a saved card, , the query-“<b>worldpayTokens</b>” is used to retrieve all the saved card tokens of the specific shopper.</p>
<b>Set the Payment method using the saved card (token only)</b>	<p><u>Mutation</u> - <b>setPaymentMethodOnCart</b></p> <p><u>Inputs</u> -</p> <p><b>Code</b> - “worldpay_cc”, <b>tokenID</b>– token retrieved through worldpayTokens.</p> <p>All other input values of “<b>setPaymentMethodOnCart</b>” should be passed empty string”.</p>
<b>Set Payment method using the saved card with token and cvc enabled</b>	<p><u>Mutation</u> - <b>setPaymentMethodOnCart</b></p> <p><u>Inputs</u> -</p> <p><b>Code</b>“worldpay_cc”, <b>tokenID</b> – token retrieved through worldpayTokens, <b>cvc</b>- entered by the shopper.</p> <p>All other input values of “<b>setPaymentMethodOnCart</b>” should be passed as an empty string”.</p>
<b>Set the Payment method using saved card with token and cvcHref</b>	<p><u>Mutation</u> - <b>setPaymentMethodOnCart</b></p> <p><u>Inputs</u> -</p> <p><b>Code</b> - “worldpay_cc”, <b>tokenID</b> – token retrieved through worldpayTokens, <b>cvcHref</b> - Generated by WebSDK integration developer.</p> <p>All other input values of “<b>setPaymentMethodOnCart</b>” should be passed as an empty string”.</p>

<b>Set the Payment method for new card using token URL(tokenHref)</b>	<p><u>Mutation</u> - <code>setPaymentMethodOnCart</code></p> <p><u>Inputs</u> -</p> <p><b>Code</b> - "worldpay_cc", <code>tokenUrl</code> – token href retrieved by creating verified tokens with namespace, <code>save_card</code> – an option for a registered user to save their card (Accepted values "0" or "1").</p> <p>All other input values of "<code>setPaymentMethodOnCart</code>" should be passed as an empty string".</p>
<b>Set the Payment method using Google Pay</b>	<p><u>Mutation</u> - <code>setPaymentMethodOnCart</code></p> <p><u>Inputs</u> -</p> <p><b>Code</b> - "worldpay_wallets", <code>googlepayToken</code> – Google Pay Token.</p> <p>All other input values of "<code>setPaymentMethodOnCart</code>" should be passed as an empty string".</p>
<b>Set the Payment method using Apple Pay</b>	<p><u>Mutation</u> - <code>setPaymentMethodOnCart</code></p> <p><u>Inputs</u> -</p> <p><b>Code</b> - "worldpay_wallets", <code>applepayToken</code> – Apple Pay Token.</p> <p>All other input values of "<code>setPaymentMethodOnCart</code>" should be passed as an empty string".</p>

Each of the above mutations are provided with sample request and response within the section [Worldpay Payment GraphQL](#).

- Place the order.

See: <https://devdocs.magento.com/guides/v2.4/graphql/mutations/place-order.html>

Sample Request	Sample Response
<pre>mutation {   placeOrder(     input: {       cart_id: " SG1FYDc4oCdsXrU6RQyzlY5aAhTrv3w8"     }   ) {     order {       order_number     }   } }</pre>	<pre>{   "data": {     "placeOrder": {       "order": {         "order_number": "000000006"       }     }   } }</pre>

## Worldpay Payment GraphQL

### Set the Payment method using a new card

Mutation	Sample Request	Sample Response
<p><b>setPaymentMethodOnCart</b></p> <p><b>*Note</b> code: All Worldpay credit card payments should be given code as "worldpay_cc" cart_id: Refer create empty cart mutation cc_name: Card holder name provided by shopper in the frontend cc_number: Credit card number provided by shopper in the frontend cc_exp_month: Card expiry month cc_exp_year: Card expiry year cvc: cvc save_card: "1", saves the card save_card: "0" does not save the card</p>	<pre>mutation {   setPaymentMethodOnCart(input: {     cart_id: "SG1FYDc4oCdsXrU6RQyzlY5aAhTrv3w8"     payment_method: {       code: "worldpay_cc"     }     worldpay_cc: {       cc_name: "John"       cc_number: "4444333322221111"       cc_exp_month: "12"       cc_exp_year: "2029"       cvc: "221"       save_card: "0"       tokenUrl: ""       applepayToken: ""       googlepayToken: ""       cvcHref: ""       sessionHref: ""       tokenId: ""     }   }) {     cart {       selected_payment_method {         code         title       }     }   } }</pre>	<pre>{   "data": {     "setPaymentMethodOnCart": {       "card": {         "selected_payment_method": {           "code": "worldpay_cc",           "title": "Credit Cards"         }       }     }   } }</pre>

## Query to retrieve all saved cards for the shopper

Use this query to list all the stored cards available to a registered user.

Sample Request	Sample Response
<pre> query {   worldpayTokens {     cards {       id       tokenId       cardnumber       cardholdername       cardexpirymonth       cardexpiryyear       method     }   } } </pre>	<pre> {   "data": {     "worldpayTokens": {       "cards": [         {           "id": 5,           "tokenId": "9905143360240686982",           "cardnumber": "3434*****3434",           "cardholdername": "3D.Authorised",           "cardexpirymonth": 10,           "cardexpiryyear": 2025,           "method": "worldpay_cc"         }       ]     }   } } </pre>

## Set the Payment method using the saved card (token only)

Mutation	Sample Request	Sample Response
<b>setPaymentMethodOnCart</b>  <b>*Note</b> cart_id: Refer create empty cart mutation tokenId: tokenId from response of worldpayTokens query	<pre> mutation {   setPaymentMethodOnCart(input: {     cart_id: "sG1FYDc4oCdsXrU6RQyzlY5aAhTrv3w8"     payment_method: {       code: "worldpay_cc"     }     worldpay_cc: {       cc_name: ""       cc_number: ""       cc_exp_month: ""       cc_exp_year: ""       cvc: ""       save_card: ""       tokenUrl: ""       applepayToken: ""       googlepayToken: ""       cvcHref: ""       sessionHref: ""       tokenId: "9905143360240686982"     }   }) {     cart {       selected_payment_method {         code         title       }     }   } } </pre>	<pre> {   "data": {     "setPaymentMethodOnCart": {       "card": {         "selected_payment_method": {           "code": "worldpay_cc",           "title": "Credit Cards"         }       }     }   } } </pre>

	<pre> } } } } </pre>	
--	----------------------	--

## Set Payment method using the saved card with cvc enabled

Mutation	Sample Request	Sample Response
<p><b>setPaymentMethodOnCart</b></p> <p><b>*Note</b>  <b>card_id:</b> Refer create empty card mutation</p> <p><b>tokenId:</b> tokenId from response of worldpayTokens query</p> <p><b>cvc:</b> enter cvc/cvv number.</p>	<pre> mutation {   setPaymentMethodOnCart(input: {     card_id: "sG1FYDc4oCdsXrU6RQyzlY5aAhTrv3w8"     payment_method: {       code: "worldpay_cc"       worldpay_cc: {         cc_name: ""         cc_number: ""         cc_exp_month: ""         cc_exp_year: ""         cvc: "123"         save_card: ""       }     }     tokenId: ""     applepayToken: ""     googlepayToken: ""     cvcHref: ""     sessionHref: ""     tokenId: "9905143360240686982"   }) {     card {       selected_payment_method {         code         title       }     }   } } </pre>	<pre> {   "data": {     "setPaymentMethodOnCart": {       "card": {         "selected_payment_method": {           "code": "worldpay_cc",           "title": "Credit Cards"         }       }     }   } } </pre>

## Set Payment method using the saved card with cvc enabled (cvcHref)

Do the following:

1. Go to- <https://developer.worldpay.com/docs/access-worldpay/checkout/web/card-and-cvc#create-a-session-for-cvc-only>.
2. Enter a sample cvc and click the **Confirm** button to submit the form.
3. Observe the cvcHref provided by the Web SDK within the alert.
4. The frontend/integration developer generates the cvcHref after Web SDK integration for a previously received saved card token.
5. Post this integration once the cvcHref is generated, you can use the same one in the payload attribute "cvcHref".

Mutation	Sample Request	Sample Response
<b>setPaymentMethodOnCard</b>  <b>*Note</b> <b>card_id:</b> Refer create empty card mutation  <b>tokenId:</b> tokenId from response of worldpayTokens query  <b>cvchref:</b> pass the cvchref returned from Worldpay websdk.	<pre> mutation {   setPaymentMethodOnCard(input: {     card_id: "sG1FYDc4oCdsXrU6RQyzlY5aAhTrv3w8"     payment_method: {       code: "worldpay_cc"     }     worldpay_cc: {       cc_name: ""       cc_number: ""       cc_exp_month: ""       cc_exp_year: ""       cvc: ""       save_card: ""     }     tokenUrl: ""     applepayToken: ""     googlepayToken: ""     cvchref: "https://try.access.worldpay.com/sessions/eyJrIjoxLCJkIjoiwUsyaDVxV0lhblhUcVBGZW1RVmpHdkx6dU1SbFcwY2JMbHVvNE9VZ1Bhc2NNT0tCR3QyYVEyZDBCTWRiUDN4YVhmYnh1WnB4TGhtT0s4N1RySnFhcFE9PSJ9"     sessionHref: ""     tokenId: "9905143360240686982"   }) {     card {       selected_payment_method {         code         title       }     }   } }</pre>	<pre> {   "data": {     "setPaymentMethodOnCard": {       "card": {         "selected_payment_method": {           "code": "worldpay_cc",           "title": "Credit Cards"         }       }     }   } }</pre>

## Set Payment method for new card using token URL (tokenHref)

1. Go to-<https://developer.worldpay.com/docs/access-worldpay/checkout/web/card-only>.
2. Enter the sample card details click the **Pay Now** button to submit the form.
3. Observe the sessionHref provided by Web SDK within the alert.
4. The frontend/integration developer generates the sessionHref after Web SDK integration and creates a verified token with namespace.
5. To create a verified token using sessionHref go to <https://developer.worldpay.com/docs/access-worldpay/verified-tokens/create-verified-token> and see the *Session with a namespace* section.
6. From the response you get, retrieve the href url present under \_links->tokens:token->href. Pass this value against tokenUrl field of the below mutation.
7. Post this integration once the tokenurl is generated, you can use the same one in the payload attribute "tokenUrl".

Mutation	Sample Request	Sample Response
----------	----------------	-----------------



<p><b>setPaymentMethodOnCart</b></p> <p><b>*Note</b> cart_id: Refer create empty cart mutation</p> <p>tokenUrl: pass the token url obtained by creating a verified token.</p> <p>save_card: "1", saves the card</p> <p>save_card: "0" does not save the card</p>	<pre>mutation {   setPaymentMethodOnCart(input: {     cart_id: "sG1FYDc4oCdsXrU6RQyz1Y5aAhTrv3w8"     payment_method: {       code: "worldpay_cc"       worldpay_cc: {         cc_name: ""         cc_number: ""         cc_exp_month: ""         cc_exp_year: ""         cvc: ""         save_card: ""         tokenUrl: " <a href="https://try.access.worldpay.com/tokens/eyJrIjoxLCJkIjoizkY0Yk1GT0xGWjRPWE9ZVWwrNF15djdodnBpVzdaY3VORVRUUGgwMm1Zc0JHTTgzUWdNcC9sV0Zjb1lXSHJ6cyJ9">https://try.access.worldpay.com/tokens/eyJrIjoxLCJkIjoizkY0Yk1GT0xGWjRPWE9ZVWwrNF15djdodnBpVzdaY3VORVRUUGgwMm1Zc0JHTTgzUWdNcC9sV0Zjb1lXSHJ6cyJ9</a> "         applepayToken: ""         googlepayToken: ""         cvcHref: ""         sessionHref: ""         tokenId: ""       }     }   }) {     cart {       selected_payment_method {         code         title       }     }   } }</pre>	<pre>{   "data": {     "setPaymentMethodOnCart": {       "card": {         "selected_payment_method": {           "code": "worldpay_cc",           "title": "Credit Cards"         }       }     }   } }</pre>
--	---	--

## GraphQL: List/Update/Delete Credit Cards

Replace the values (ID and tokenId) with the values obtained when you run *List all Worldpay saved cards* *graphql*.

Mutation	Sample Request	Sample Response
<p><b>List Vault payment tokens (Provided by Magento):</b></p>	<pre>query {   customerPaymentTokens {     items {       details       public_hash       payment_method_code       type     }   } }</pre>	<pre>{   "data": {     "customerPaymentTokens": {       "items": [         {           "details": "{\"type\":\"VISA-SSL\", \"maskedCC\":\"1111\", \"expirationDate\":\"3\\/2025\"}",           "public_hash": "af97c2d5b4365bf69e0a57535cfb3feb358837da1e8f572be269fe1423935ac9",</pre>

		<pre> "payment_method_code": "worldpay_cc", "type": "card"     }   ] } } } </pre>
<b>List all Worldpay saved cards</b>	<pre> query {   worldpayTokens {     cards {       id       tokenId       cardnumber       cardholdername       cardexpirymonth       cardexpiryyear       method     }   } } </pre>	<pre> {   "data": {     "worldpayTokens": {       "cards": [         {           "id": 5,           "tokenId": "9905143360240686982",           "cardnumber": "3434*****3434",           "cardholdername": "3D.Authorised",           "cardexpirymonth": 10,           "cardexpiryyear": 2025,           "method": "worldpay_cc"         }       ]     }   } } </pre>
<b>Update card</b>	<pre> mutation {   updateWorldpayToken (     id: 5,     input : {       tokenId: "9910076467795747092"       cardholdername: "John"       cardexpirymonth: 10       cardexpiryyear: 2025     }   ) {     card {       id       tokenId       cardnumber       cardholdername       cardexpirymonth       cardexpiryyear       method     }   } } </pre>	<pre> {   "data": {     "updateWorldpayToken": {       "card": {         "id": 5,         "tokenId": "9910076467795747092",         "cardnumber": "6011*****0000",         "cardholdername": "John",         "cardexpirymonth": 10,         "cardexpiryyear": 2025,         "method": "worldpay_cc"       }     }   } } </pre>

<b>Delete card</b>	<pre>mutation {   deleteWorldpayToken (     id: 3     tokenId: "9905143360240686982"   ){     result   } }</pre>	<pre>{   "data": {     "deleteWorldpayToken": {       "result": true     }   } }</pre>
--------------------	--	--

## Instant Purchase

Magento provides the flexibility to registered shoppers who have saved credit cards in their account to make an Instant Purchase. To make a Worldpay integration through GraphQL, integrate the below endpoints in sequence.

GraphQL endpoints to be executed in order	Description	Magento native/ Worldpay customized
<b>Create an empty cart</b>	Create customer cart (customerCart).	Native
<b>Add products to cart</b>	Add a product using addSimpleProductsToCart and modify the cart ID as generated by create empty cart mutation.	Native
<b>Set the shipping address</b>	Use the setShippingAddressesOnCart mutation with an appropriate cartId.	Native
<b>Set the billing address</b>	Use the setBillingAddressOnCart mutation to set a billing address.	Native
<b>Set the delivery method</b>	Use the setShippingMethodsOnCart mutation to set a delivery method. Magento usually chooses the lowest priced delivery method.	Native
<b>Set the payment method</b>	See the <a href="#">WorldpayToken</a> Query section to view all the stored cards available for a registered user and fetch the tokenId from the query response. Set payment method using the <a href="#">Set Payment method using the saved card(token only)</a> function.	Worldpay customized payment types
<b>Place the order</b>	The placeOrder mutation places an order when cart Id is specified.	Native

## Google Pay

### Query to retrieve Google Pay Settings

Sample Request	Sample Response
<pre>query {</pre>	<pre>{</pre>

<pre> googlepaySettings {   settings {     enabled     paymentmethods {method}     supportedauthentication {option}     gatewayname     gatewaymerchantid     googlemerchantid     googlemerchantname   } } </pre>	<pre> "data": {   "googlepaySettings": {     "settings": [       {         "enabled": 1,         "paymentmethods": {           "method": [             "AMEX",             "VISA",             "DISCOVER",             "JCB",             "MASTERCARD"           ]         },         "supportedauthentication": {           "option": [             "PAN_ONLY",             "CRYPTOGRAM_3DS"           ]         },         "gatewayname": "worldpay",         "gatewaymerchantid": "627392fb1cafcbl",         "googlemerchantid": null,         "googlemerchantname": "worldpay"       }     ]   } } </pre>
--	---

## Set Payment method with Google Pay wallet

Make a Google Pay integration in the front end and generate the Google Pay token. Use this token to set the payment method. See the Worldpay documentation for information on the [Google Pay integration](#).

Mutation	Sample Request	Sample Response
<b>setPaymentMethodOnCart</b>	<pre> mutation {   setPaymentMethodOnCart(input: {     cart_id: "Enter Customer Cart ID"     payment_method: {       code: "worldpay_wallets"       worldpay_cc: {         cc_name: ""         cc_number: ""         cc_exp_month: ""         cc_exp_year: ""         cvc: ""         save_card: ""       }     }   }) } </pre>	<pre> {   "data": {     "setPaymentMethodOnCart": {       {         "cart": {           "selected_payment_method": {             {               "code": worldpay_wallets",               "title": "Wallets"             }           }         }       }     }   } } </pre>

	<pre> applepayToken: "" googlepayToken:"Enter Valid Google Pay Token" cvcHref: "" sessionHref:"" tokenId:"" } } )) {   cart {     selected_payment_method {       code       title     }   } } } </pre>	<pre> } } </pre>
--	---	------------------

## Sample Google Pay Token

```

{"signature\":"MEYCIQCMlsIyXiM7BxnQe2GVP0WQK5mcUpctJibASXD9yAixogIhAPcccY0MEEatecNMVEKwA
Ty+8sOREVDvQ36OWQ4gIGK1\","protocolVersion\":"ECv1\","signedMessage\":"{\\\\"encryptedM
essage\\\\"":"\\\\"+mmAI3sWoThxgZ6K8eqYjkyM88v0nBnSlMA2f0/17/9yYSmjz5u5iZL2GEn/kutMJ0UiIQ9+S+z
hmFclTMgLVZhipCduL6yUkz4GAVtTgzQ/h/J27Svx+P0pi7PMhFLKTMz3vVyjFiudYv2sEeHSgIvP4W4cRAh260ZEA
i8X8Xur+l4iuw6udckjNRUq457VrNjr7qWz+lBgToqi3qczCERYoj1tnf3EmOblgaSKpWtcyY7UMyfYUE3ty2mk3pJ
m+8YvdhpkAyN/g1D5sWR9zBAYsc5qeAt4MazZGedlGTbvGhUfCN0PUXRvX2014pr4tKrxBeaI+iT+6ijRf2sjxpYp/
UbUXtXNxxqZun2vYuKeiPhiaFfvBUv2woOUxWHLJSNqoZ2Uek+qDsKuxZeFscTJV+tHzhVJW95skVWjmnqyzPIX2JCf
Wg9R65FYN+ib8DM1xxn6XrPK6CA\\\\u003d\\\\u003d\\\\","\\\\"ephemeralPublicKey\\\\"":"\\\\"BHcOlQC4g
ttnXQtMUCeflJw/X44fvEKS2YpvNBNzN1/qFuiArVPqRd+D+JsndddzWXuqXtWq01DBZfy0yE75y6w\\\\u003d\\\\
","\\\\"tag\\\\"":"\\\\"TNgnyjs878KS8UGHz1jQriU/+MOSz4EX1tQL0HccM/s\\\\u003d\\\\\\"}\\\"}

```

## Apple Pay

### Query to retrieve Apple Pay Settings

Sample Request	Sample Response
<pre> query {   applepaySettings {     settings {       enabled       certificationkey       certificationcrt       certificationpassword       merchantname       domainname     }   } } </pre>	<pre> {   "data": {     "applepaySettings": {       "settings": [         {           "enabled": 1,           "certificationkey": " ",           "certificationcrt": " ",           "certificationpassword": " ",           "merchantname": "worldpay",           "domainname": " "         }       ]     }   } } </pre>

**\*Note**

**certificationkey:** The location of the key pem file placed on the web server

**certificationcrt:** The location of the crt pem file placed on the web server

```
]
}
}
}
```

## Set Payment method with Apple Pay wallet

Make an Apple Pay integration in the front end and generate the Apple Pay token. Use this token to set the payment method. See the Worldpay documentation for information on the Apple Pay integration for both [web](#) and [in-app](#).

Mutation	Sample Request	Sample Response
<b>setPaymentMethodOnCart</b>	<pre>mutation {   setPaymentMethodOnCart(input: {     cart_id: "Enter Customer Cart ID"     payment_method: {       code: "worldpay_wallets"       worldpay_cc: {         cc_name: ""         cc_number: ""         cc_exp_month: ""         cc_exp_year: ""         cvc: ""         save_card: ""         applepayToken: " Enter Valid Apple Pay Token "         googlepayToken: ""         cvcHref: ""         sessionHref: ""         tokenId: ""       }     }   }) {     cart {       selected_payment_method {         code         title       }     }   } }</pre>	<pre>{   "data": {     "setPaymentMethodOnCart": {       "card": {         "selected_payment_method": {           "code": worldpay_wallets",           "title": "Wallets"         }       }     }   } }</pre>



21

Query	Sample Response
<pre> query {   camSettings {     settings {       accountLevelMessage {         setting {           list {             messageCode             actualMessage             customMessage           }         }       }       accessWorldPayMessage {         setting {           list {             messageCode             actualMessage             customMessage           }         }       }       checkoutMessage {         setting {           list {             messageCode             actualMessage             customMessage           }         }       }     }   } } </pre>	<pre> {   "data": {     "camSettings": {       "settings": [         {           "accountLevelMessage": {             "setting": {               "list": [                 {                   "messageCode": "MCAM0",                   "actualMessage": "Please verify the Billing Address in your Address Book before adding new card!",                   "customMessage": ""                 },                 {                   "messageCode": "MCAM1",                   "actualMessage": "Token doesnot exist anymore, Please delete the card.",                   "customMessage": ""                 },                 {                   "messageCode": "MCAM2",                   "actualMessage": "The card has been updated.",                   "customMessage": ""                 },                 {                   "messageCode": "GMCAM17",                   "actualMessage": "Token data supplied does not exist for current customer",                   "customMessage": ""                 }               ]             }           },           "accessWorldPayMessage": {             "setting": {               "list": [                 {                   "messageCode": "ACAM12",                   "actualMessage": "Error Code %s already exist!",                   "customMessage": ""                 },                 {                   "messageCode": "ACAM13", </pre>

```
"actualMessage": "Detected  
only whitespace character for code",  
  
    "customMessage": ""  
}  
]  
  
},  
  "checkoutMessage": {  
    "setting": {  
      "list": [  
        {  
          "messageCode": "CCAM0",  
          "actualMessage": "Card  
number should contain between 12 and 20 numeric  
characters",  
  
          "customMessage": ""  
        },  
        {  
          "messageCode": "CCAM1",  
          "actualMessage": "Please,  
Verify the disclaimer! before saving the card",  
          "customMessage": ""  
        },  
        {  
          "messageCode": "  
GCCAM9",  
  
          "actualMessage": "  
Required parameter \"save_card\" for  
worldpay_cc\" is missing.",  
  
          "customMessage": ""  
        }  
      ]  
    }  
  }  
}]
```

Query	Sample Response
<pre> query {   customLabelsSettings {     settings {       accountLabels {         setting {           list {             labelCode             defaultLabel             customLabel           }         }         adminLabels {           setting {             list {               labelCode               defaultLabel               customLabel             }           }           checkoutLabels {             setting {               list {                 labelCode                 defaultLabel                 customLabel               }             }           }         }       }     }   } } </pre>	<pre> {   "data": {     "customLabelsSettings": {       "settings": [         {           "accountLabels": {             "setting": {               "list": [                 {                   "labelCode": "AC1",                   "defaultLabel": "Card Holder Name"                   "customLabel": "Card Holder Name- custom"                 },                 {                   "labelCode": "AC2",                   "defaultLabel": "Card Brand",                   "customLabel": "Card Brand-custom"                 },                 {                   "labelCode": "AC3",                   "defaultLabel": "Card Number",                   "customLabel": ""                 },                 {                   "labelCode": "AC4",                   "defaultLabel": "Card Expiry Month"                   "customLabel": ""                 },                 {                   "labelCode": "AC5",                   "defaultLabel": "Card Expiry Year"                   "customLabel": ""                 },                 {                   "labelCode": "AC6",                   "defaultLabel": "Update",                   "customLabel": ""                 },                 {                   "labelCode": "AC7",                   "defaultLabel": "Update Saved Card"                   "customLabel": ""                 }               ]             }           }         }       ]     }   } } </pre>

<pre>         }     } } </pre>	<pre> ,  ",  mer!", </pre>	<pre> {     "labelCode": "AC8",     "defaultLabel": "Card Information"      "customLabel": "" }, {     "labelCode": "AC9",     "defaultLabel": "Expiry Month/Year"      "customLabel": "" }, {     "labelCode": "AC10",     "defaultLabel": "Delete",     "customLabel": "" }, {     "labelCode": "AC11",     "defaultLabel": "Add New Card",     "customLabel": "" }, {     "labelCode": "AC12",     "defaultLabel": "Credit Card Type"      "customLabel": "" }, {     "labelCode": "AC13",     "defaultLabel": "Save",     "customLabel": "" }, {     "labelCode": "AC14",     "defaultLabel": "My Saved Card",     "customLabel": "" }, {     "labelCode": "AC15",     "defaultLabel": "Important Disclai      "customLabel": "" }, {     "labelCode": "AC16",     "defaultLabel": "Disclaimer!",     "customLabel": "" }, { </pre>
--------------------------------	----------------------------	---

```

        "labelCode": "AC17",
        "defaultLabel": "CVV",
        "customLabel": ""
    },
    {
        "labelCode": "AC18",
        "defaultLabel": "Default Billing A
ddress",
        "customLabel": ""
    },
    {
        "labelCode": "AC19",
        "defaultLabel": "You have no Saved
Card.",
        "customLabel": ""
    }
]
},
"adminLabels": {
    "setting": {
        "list": []
    }
},
"checkoutLabels": {
    "setting": {
        "list": [
            {
                "labelCode": "C01",
                "defaultLabel": "New Card",
                "customLabel": "New Card-custom"
            },
            {
                "labelCode": "C02",
                "defaultLabel": "We Accept",
                "customLabel": "We Accept-custom"
            },
            {
                "labelCode": "C03",
                "defaultLabel": "Card Number",
                "customLabel": ""
            },
            {
                "labelCode": "C04",
                "defaultLabel": "Card Holder Name"
                "customLabel": ""
            },
            {
                "labelCode": "C05",

```



mer!",

```

        "defaultLabel": "CVV",
        "customLabel": ""
    },
    {
        "labelCode": "C06",
        "defaultLabel": "Month",
        "customLabel": ""
    },
    {
        "labelCode": "C07",
        "defaultLabel": "Year",
        "customLabel": ""
    },
    {
        "labelCode": "C08",
        "defaultLabel": "Save This Card",
        "customLabel": ""
    },
    {
        "labelCode": "C09",
        "defaultLabel": "Important Disclai

        "customLabel": ""
    },
    {
        "labelCode": "C010",
        "defaultLabel": "Pay Now",
        "customLabel": ""
    },
    {
        "labelCode": "C011",
        "defaultLabel": "MM/YY",
        "customLabel": ""
    },
    {
        "labelCode": "C012",
        "defaultLabel": "Saved cards",
        "customLabel": ""
    },
    {
        "labelCode": "C013",
        "defaultLabel": "Use Saved Card",
        "customLabel": ""
    },
    {
        "labelCode": "C014",
        "defaultLabel": "Place Order",
        "customLabel": ""
    },
    {

```

```
        "labelCode": "C015",
        "defaultLabel": "Saved Card feature will be available only if enabled by Merchant.",
        "customLabel": ""
    },
    {
        "labelCode": "C016",
        "defaultLabel": "Card Verification Number",
        "customLabel": ""
    },
    {
        "labelCode": "C017",
        "defaultLabel": "Disclaimer!",
        "customLabel": ""
    }
]
}
```

## References:

Magento default GraphQL: <https://devdocs.magento.com/guides/v2.4/graphql/index.html>

Worldpay Websdk Integration : <https://developer.worldpay.com/docs/access-worldpay/checkout/web>

\*Note: The 3DS Flex support of GraphQL interface is deferred to next significant Worldpay plugin release.