

Magento 2 ACCESS WORLDPAY GraphQL API Guide

July 2022

Contents

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

Payments plugin for Magento 2



Query to retrieve the existing Configurable Labels	23
Deferences	20
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 <u>GraphQL developer guide</u>. 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
Create Customer	One time activity to register new shopper with the <u>createCustomer</u> mutation. Once registered, you can create an authenticated token fo r existing shoppers using <u>generateCustomerToken</u> 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>'.</token>	Native
Create an empty cart	Create customer cart (customerCart) for registered shoppers or use <u>createemptycart</u> mutations for guest shoppers.	Native
Mapping email with cart for guest shopper	The card ID and the email ID mapping is done using the mutation setGuestEmailOnCart.	Native
Add products to cart	Add product using <u>addSimpleProductsToCart</u> and modify the cart ID as generated by create empty cart mutation.	Native
Set the shipping address	setShippingAddressesOnCart mutation to be used with appropriate cart ID.	Native
Set the delivery method	Use the setShippingMethodsOnCart mutation to set a delivery meth od.	Native
Set the billing address	Use the setBillingAddressOnCart mutation to set a billing address.	Native
Set the payment method	Use the <u>cart</u> 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
Place the order	The <u>placeOrder</u> 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
Credit Card integration	GraphQL support for save card, list cards, update card, delete card, place order with new or saved(tokenized) credit card.
Instant Purchase	Make the worldpayTokens query if the shopper is eligible for Instant Purchase, and make the payment with a saved card token.
Google Pay	Query to retrieve the Google Pay settings and set the payment method using Google pay.
Apple Pay	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 3lodst49pvg7n2d2fmyyjlp06kr8k6jf

1. Create an empty cart for a registered user.

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

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
mutation {	{
createEmptyCart	"data": {



```
"createEmptyCart": "oOZjoRvISf7emel0AQpkUg9AwhluEylR"
}
}
```

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
mutation {
                                     {
                                       "data": {
  setGuestEmailOnCart(
    input: {
                                         "setGuestEmailOnCart": {
                                           "cart": {
      cart_id: "
oOZjoRvISf7emel0AQpkUg9AwhluEylR"
                                              "email": "jdoe@example.com"
      email: "jdoe@example.com"
                                           }
    }
                                         }
  ) {
                                       }
    cart {
                                     }
      email
    }
  }
}
```

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
mutation {
                                                     addSimpleProductsToCart(
                                                     "addSimpleProductsToCart": {
    input: {
                                                     "cart": {
      cart_id: "sG1FYDc4oCdsXrU6RQyz1Y5aAhTrv3w8"
                                                     "items": [
      cart_items: [
                                                     {
"id": "57",
':c+":
        {
          data: {
                                                     "product": {
            quantity: 1
            sku: "24-MB02"
                                                     "name": "Fusion Backpack",
                                                     "sku": "24-MB02"
        }
                                                     },
      ]
                                                     "quantity": 2
    }
                                                     }
  ) {
                                                     ]
    cart {
      items {
                                                     }
}
        product {
          name
          sku
        quantity
```

```
} }
```

4. Add a shipping address.

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

```
Sample Request
                                                        Sample Response
                                                        {
"data": {
mutation {
  setShippingAddressesOnCart(
                                                        "setShippingAddressesOnCart": {
    input: {
      cart_id: "sG1FYDc4oCdsXrU6RQyz1Y5aAhTrv3w8"
                                                        "cart": {
                                                        "shipping_addresses": [
      shipping addresses: [
        {
                                                        address: {
            firstname: "Jacob"
                                                        "lastname": "Hunt",
            lastname: "Hunt"
                                                        "company": "Magento",
            company: "Magento"
                                                        "street": [
                                                        "Magento Pkwy",
            street: ["Magento Pkwy", "Main Street"]
                                                        "Main Street"
            city: "Austin"
                                                        ],
"city": "Austin",
'-". "7875
            region: "TX"
            postcode: "78758"
                                                        "postcode": "78758", "telephone": "8675309"
            country_code: "US"
            telephone: "8675309"
            save in address book: true
                                                        }
                                                        ]
        }
                                                        }
                                                        }
      ]
    }
                                                        }
  ) {
                                                        }
    cart {
      shipping_addresses {
        firstname
        lastname
        company
        street
        citv
        postcode
        telephone
      }
   }
 }
}
```

5. Set the shipping method.

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

```
"carrier_title": "Best Way",
                                                      "method_code": "bestway",
      ]
                                                      "method title": "Table Rate",
    }
                                                      "amount": {
  ) {
                                                      "value": 0,
    cart {
                                                      "currency": "USD"
      shipping_addresses {
        selected_shipping_method {
          carrier code
                                                      }
          carrier_title
          method_code
                                                      ĺ
          method_title
                                                      }
          amount {
                                                      }
            value
                                                      }
            currency
                                                      }
} }
          }
}
```

6. Add the billing address.

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

```
Sample Request
                                                      Sample Response
mutation {
                                                      {
                                                          "data": {
  setBillingAddressOnCart(
                                                               "setBillingAddressOnCart": {
      cart_id: "sG1FYDc4oCdsXrU6RQyz1Y5aAhTrv3w8"
                                                                   "cart": {
                                                                       "billing_address": {
      billing_address: {
                                                                           "firstname": "Jacob",
"lastname": "Hunt",
        address: {
          firstname: "Jacob"
          lastname: "Hunt"
                                                                           "company": "Magento",
          company: "Magento"
                                                                           "street": [
                                                                                "Magento Pkwy",
          street: ["Magento Pkwy", "Main Street"]
                                                                                "Main Street"
          city: "Austin"
          region: "TX"
          postcode: "78758"
                                                                           "city": "Austin",
          country_code: "US"
                                                                           "postcode": "78758",
                                                                           "telephone": "8675309"
          telephone: "8675309"
          save_in_address_book: true
                                                                       }
                                                                  }
                                                              }
        use_for_shipping: true
      }
                                                          }
                                                      }
    }
  ) {
    cart {
      billing address {
        firstname
        lastname
        company
        street
        city
        postcode
        telephone
      }
   }
 }
}
```



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



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

Each of the above mutations are provided with sample request and response within the section <u>Worldpay</u> <u>Payment GraphQL</u>.

8. Place the order.

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

```
Sample Request
                                       Sample Response
mutation {
                                         "data": {
 placeOrder(
   input: {
                                           "placeOrder": {
     cart_id: "
                                             "order": {
sG1FYDc4oCdsXrU6RQyz1Y5aAhTrv3w8"
                                               "order number": "000000006"
 ) {
                                          }
                                         }
   order {
     order_number
 }
}
```

Worldpay Payment GraphQL

Set the Payment method using a new card

Mutation	Sample Request	Sample Response
*Note 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	<pre>mutation { setPaymentMethodOnCart(input: { cart_id: "sG1FYDc4oCdsXrU6RQyzlY5aAhTrv3w 8" 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: ""</pre>	<pre>Sample Response { "data": { "setPaymentMethodOnCart": { "cart": { "selected_payment_method" : { "code": "worldpay_cc", "title": "Credit Cards" } } } }</pre>
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	<pre>applepayToken: "" googlepayToken: "" cvcHref:"" sessionHref:"" tokenId:"" } }) { cart { selected_payment_method { code title } } }</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
                                        {
query {
 worldpayTokens {
                                            "data": {
    cards {
                                                "worldpayTokens": {
      id
                                                    "cards": [
      tokenid
                                                        {
      cardnumber
                                                            "id": 5,
      cardholdername
                                                            "tokenid": "9905143360240686982",
      cardexpirymonth
                                                            "cardnumber": "3434*****3434",
      cardexpiryyear
                                                            "cardholdername": "3D.Authorised",
      method
                                                            "cardexpirymonth": 10,
                                                            "cardexpiryyear": 2025,
   }
                                                            "method": "worldpay_cc"
 }
}
                                                        }
                                                    ]
                                                }
                                            }
                                        }
```

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

Mutation	Sample Request	Sample Response
*Note cart_id: Refer create empty cart mutation tokenId: tokenId from response of worldpayTokens query	<pre>mutation { setPaymentMethodOnCart(input: { cart_id: "sG1FYDc4oCdsXrU6RQyz1Y5aAhTrv3w 8" payment_method: { code: "worldpay_cc" worldpay_cc: {</pre>	<pre>{ "data": { "setPaymentMethodOnCart": { "cart": { "selected_payment_method" : { "code": "worldpay_cc", "title": "Credit Cards" } } } }</pre>

}	
}	
}	
}	

Set Payment method using the saved card with cvc enabled

	data":{ setPaymentMethodOnCart":
<pre>8" payment_method: { code: "worldpay_cc" worldpay_cc: { cc_name: "" cc_number: "" cc_exp_month: "" </pre> <pre>"setPaymentMethodOnCart</pre>	<pre>"cart": { 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-assession-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
*Note cart_id: Refer create empty cart mutation tokenId: tokenID from response of worldpayTokens query cvcHref: pass the cvcHref returned from Worldpay websdk.	<pre>mutation { setPaymentMethodOnCart(input: { cart_id: "sG1FYDc4oCdsXrU6RQyz1Y5aAhTrv3w8" payment_method: { code: "worldpay_cc" worldpay_cc: {</pre>	<pre>{ "data": { "setPaymentMethodOnCart ": { "cart": { "selected_payment_metho d": { "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

```
mutation {
                            setPaymentMethodOnCart(input: {
                                                                           "data": {
                            cart id: "sG1FYDc4oCdsXrU6ROyz1Y5aAhTrv3w8
                                                                           "setPaymentMethodOnCart":
                            payment method: {
                                                                            "cart": {
                            code: "worldpay_cc"
                            worldpay_cc: {
                                                                           "selected_payment_method"
                            cc name: ""
                            cc_number: ""
                                                                             "code": "worldpay_cc",
                            cc_exp_month: ""
setPaymentMethodOnCar
                                                                             "title": "Credit Cards"
                            cc_exp_year: ""
                            cvc: "'
                            save_card: ""
                                                                             }
*Note
                            tokenUrl: "
                                                                            }
cart_id: Refer create
                                                                           }
                            https://try.access.worldpay.com/tokens/
empty cart mutation
                            eyJrIjoxLCJkIjoiZkY0YklGT0xGWjR
tokenUrl: pass the token
                            PWE9ZVWwrNF15djdodnBpVzdaY3VORVRUUGgwMm1Z
url obtained by creating
                            c0JHTTgzUWdNcC9sV0Zjb1lXSHJ6cyJ9
a verified token.
                            applepayToken: ""
save_card: "1", saves
                            googlepayToken: ""
the card
                            cvcHref:"
                            sessionHref:""
save_card: "0" does not
                            tokenId:""
save the card
                            }
                            }
                            }) {
                            cart {
                            selected_payment_method {
                            code
                            title
                            }
                            }
                            }
                            }
```

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
List Vault
            query {
payment
                                                       "data": {
              customerPaymentTokens {
tokens
                                                       "customerPaymentTokens": {
                items {
(Provided
                                                                    "items": [
                  details
                  public hash
                                                                        {
by
                  payment_method_code
                                                        "details": "{\"type\":\"VISA-
Magento):
                                                       SSL\",\"maskedCC\":\"1111\",\"expirationDate\":
                                                       \"3\\/2025\"}",
                }
                                                       "public_hash": "af97c2d5b4365bf69e0a57535cfb3fe
              }
            }
                                                       b358837da1e8f572be269fe1423935ac9",
```



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

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
Create an empty cart	Create customer cart (customerCart).	Native
Add products to cart	Add a product using addSimpleProductsToCart and modify the cart I D as generated by create empty cart mutation.	Native
Set the shipping address	Use the setShippingAddressesOnCart mutation with an appropriate cartId.	Native
Set the billing address	Use the setBillingAddressOnCart mutation to set a billing address.	Native
Set the delivery method	Use the setShippingMethodsOnCart mutation to set a delivery method. Magento usually chooses the lowest priced delivery method.	Native
Set the payment method	See the WorldpayToken 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 Set Payment method using the saved card(token only) function.	Worldpay customized payment types
Place the order	The placeOrder mutation places an order when cart Id is specified.	Native

Google Pay

Query to retrieve Google Pay Settings

Sample Request	Sample Response
query {	{

```
googlepaySettings {
                                   "data": {
settings {
                                   "googlepaySettings": {
enabled
                                   "settings": [
paymentmethods {method}
                                   {
supportedauthentication {option}
                                   "enabled": 1,
gatewayname
                                   "paymentmethods": {
                                   "method": [
gatewaymerchantid
googlemerchantid
                                   "AMEX",
                                   "VISA",
googlemerchantname
}
                                   "DISCOVER",
                                   "JCB",
}
                                   "MASTERCARD"
}
                                   1
                                   },
                                   "supportedauthentication": {
                                   "option": [
                                   "PAN_ONLY",
                                   "CRYPTOGRAM 3DS"
                                   ]
                                   },
                                   "gatewayname": "worldpay",
                                   "gatewaymerchantid": "627392fb1cafcb1",
                                   "googlemerchantid": null,
                                   "googlemerchantname": "worldpay"
                                   }
                                   1
                                   }
                                   }
                                   }
```

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 <u>Google Pay integration</u>.

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



```
applepayToken: ""
googlepayToken:"Enter Valid Google
                                          }
Pay Token"
cvcHref: ""
sessionHref:""
tokenId:""
}
}
}) {
cart {
selected payment method {
code
title
}
}
}
}
```

Sample Google Pay Token

{\"signature\":\"MEYCIQCM1sIyXiM7BxnQe2GVP0WQK5mcUpcTjibASXD9yAixogIhAPcccY0MEEatecNMVEKwA Ty+8sOREVDvQ36OWQ4gIGKl\",\"protocolVersion\":\"ECv1\",\"signedMessage\":\"{\\\"encryptedM essage\\":\\"+mmAI3sWoThxgZ6K8eqYjkym88v0nBnSlMA2f0/17/9yYSmjz5u5iZL2GEn/kutMJ0UiIQ9+S+z hmFclTMgLVZhipCduL6yUkz4GAVtTgzQ/h/J27svx+P0pi7PMhFLKTmz3vVyjFiudYv2sEeHSgIvP4W4cRAh260ZEA i8X8Xur+14iuw6udckjNRUq457VrNjr7qWz+1BgToqi3qczCERYoj1tnf3EmOblgaSKpWtcyY7UMyfYUE3ty2mk3pJ m+8YvdhpkAyN/glD5sWR9zBAYsc5qeAt4MazZGedlGTbvGhUfCN0PUXRvX2014pr4tKrxBeaI+iT+6ijRf2sjxpYp/UbUXtXNxqZun2vYuKeiPhiaFfvBUv2woOUxWHLJSNqoZ2Uek+qDsKuxZeFsctJV+tHzhVJW95skVWjmnqyzPIX2JCf Wg9R65FYN+ib8DM1xxn6XrPK6CA\\\u003d\\\",\\\"ephemeralPublicKey\\\":\\"BHcOlQC4g ttnXQtMUCef1Jw/X44fvEKS2YpvNBNzNl/qFUiArVPqRd+D+JsndddzWXuqXtWq01DBZfY0yE75y6w\\\u003d\\\",\\\"taq\\\":\\"TNgnyjs878KS8UGHz1jQriU/+MOSz4EX1tQL0HccM/s\\\u003d\\\"}\"}

Apple Pay

Query to retrieve Apple Pay Settings

```
Sample Request
                                   Sample Response
query {
                                   {
                                   "data": {
applepaySettings {
settings {
                                   "applepaySettings": {
                                   "settings": [
enabled
certificationkey
                                   "enabled": 1,
certificationcrt
                                   "certificationkey": " ",
certificationpassword
merchantname
                                   "certificationcrt": " ",
                                   "certificationpassword": " ",
domainname
                                   "merchantname": "worldpay",
1
                                   "domainname": " "
}
}
                                   }
```

```
*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
setPaymentMethodOnCart	mutation {	{
	setPaymentMethodOnCart(input: {	"data": {
	cart id: "Enter Customer Cart ID"	"setPaymentMethodOnCart":
	payment_method: {	{
	code: "worldpay_wallets"	"cart": {
	worldpay_cc: {	"selected_payment_method'
	cc_name: ""	
	cc_number: ""	"code": worldpay_wallets'
	cc_exp_month: ""	"title": "Wallets"
	cc_exp_year: ""	}
	cvc: ""	}
	save_card: ""	}
	applepayToken: " Enter Valid Apple	}
	Pay Token "	}
	googlepayToken: ""	
	cvcHref: ""	
	sessionHref:""	
	tokenId:""	
	}	
	}	
	}) {	
	cart {	
	selected payment method {	
	code	
	title	
	}	
	}	
	}	
	}	



Sample Apple Pay Token

{\"signature\":\"MIAGCSqGSIb3DQEHAqCAMIACAQExDzANBqlqhkqBZQMEAgEFADCABgkqhkiG9w0BBwEAAKCAM IID5jCCA4uqAwIBAqIIaGD2mdnMpw8wCqYIKoZIzj0EAwIwejEuMCwGA1UEAwwlQXBwbGUqQXBwbGljYXRpb24qSW5 kFwcGxlIEluYy4xCzAJBqNVBAYTAlVTMB4XDTE2MDYwMzE4MTY0MFoXDTIxMDYwMjE4MTY0MFowYjEoMCYGAlUEAww fZWNjLXNtcC1icm9rZXItc2lnbl9VQzQtU0FOREJPWDEUMBIGA1UECwwLaU9TIFN5c3RlbXMxEzARBqNVBAoMCkFwc GxlIEluYy4xCzAJBgNVBAYTAlVTMFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEgjD9q8Oc914gLFDZm0US5jfiqQH dbLPqsc1LUmeY+M9OveqaJajCHkwz3c6OKpbC9q+hkwNFxOh6RCbOlRsSlaOCAhEwqqINMEUGCCsGAQUFBwEBBDkwN zA1BqqrBqEFBQcwAYYpaHR0cDovL29jc3AuYXBwbGUuY29tL29jc3AwNC1hcHBsZWFpY2EzMDIwHQYDVR00BBYEFAI kMAua7u1GMZekplopnkJxghxFMAwGA1UdEwEB/wQCMAAwHwYDVR0jBBgwFoAUI/JJxE+T508n5sT2KGw/orv9Lkswg gEdBgNVHSAEggEUMIIBEDCCAQwGCSqGSIb3Y2QFATCB/jCBwwYIKwYBBQUHAgIwgbYMgbNSZWxpYW5jZSBvbiB0aG1 $\verb|ziGNlcnRpZmljYXRliGJ5iGFueSBwYXJ0eSBhc3N1bWVziGFjY2VwdGFuY2Ugb2YgdGhliHRoZW4gYXBwbGljYWJsZiGFjY2VwdGFuY2Ugb2YgdGhliHRoZW4gYXBwbGljYWJsZiGFjY2VwdGFuY2Ugb2YgdGhliHRoZW4gYXBwbGljYWJsZiGFjY2VwdGFuY2Ugb2YgdGhliHRoZW4gYXBwbGljYWJsZiGFjY2VwdGFuY2Ugb2YgdGhliHRoZW4gYXBwbGljYWJsZiGFjY2VwdGFuY2Ugb2YgdGhliHRoZW4gYXBwbGljYWJsZiGFjY2VwdGFuY2Ugb2YgdGhliHRoZW4gYXBwbGljYWJsZiGFjY2VwdGFuY2Ugb2YgdGhliHRoZW4gYXBwbGljYWJsZiGFjY2VwdGFuY2Ugb2YgdGhliHRoZW4gYXBwbGljYWJsZiGFjY2VwdGFuY2Ugb2YgdGhliHRoZW4gYXBwbGljYWJsZiGFjYYVwdGFuY2Ugb2YgdGhliHRoZW4gYXBwbGljYWJsZiGFjYYVwdGFuY2Ugb2YgdGhliHRoZW4gYXBwbGljYWJsZiGFjYYVwdGFuY2Ugb2YgdGhliHRoZW4gYXBwbGljYWJsZiGFjYYVwdGFuY2Ugb2YgdGhliHRoZW4gYXBwbGljYWJsZiGFjYYVwdGFuY2Ugb2YgdGhliHRoZW4gYXBwbGljYWJsZiGFjYYVwdGFuY2Ugb2YgdGhliHRoZW4gYXBwbGljYWJsZiGFjYYVwdGFuY2Ugb2YgdGhliHRoZW4gYXBwbGljYWJsZiGFjYYVwdGFuY2Ugb2YgdGhliHRoZW4gYXBwbGljYWJsZiGFjYYVwdGFuY2Ugb2YgdGhliHRoZW4gYXBwbGljYWJsZiGFjYYVwdGFuY2Ugb2YgdGhliHRoZW4gYXBwbGljYWJsZiGFjYYVwdGfuY2Ugb2YgdGhliHRoZW4gYXBwbGljYWJsZiGFjYWJsZiGFyYWdGfuYYUgb2YgdGhliHRoZW4gYXBwbGljYWJsZiGFyYWdGfuYYUgb2YgdGhliHRoZW4gYXBwbGljYWJsZiGFyYWdGfuYYUgb2YgdGhliHRoZW4gYXBwbGljYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYYWdfuYY$ SBzdGFuZGFvZCB0ZXJtcvBhbmOgY29uZG10aW9ucvBvZiB1c2UsIGN1cnRpZmliYXR1IHBvbGlieSBhbmOgY2VvdG1 maWNhdGlvbiBwcmFjdGljZSBzdGF0ZW1lbnRzLjA2BggrBgEFBQcCARYgaHR0cDovL3d3dy5hcHBsZS5jb20vY2Vyd GlmaWNhdGVhdXRob3JpdHkvMDQGA1UdHwQtMCswKaAnoCWGI2h0dHA6Ly9jcmwuYXBwbGUuY29tL2FwcGx1YWljYTM uY3JsMA4GA1UdDwEB/wQEAwIHgDAPBgkqhkiG92NkBh0EAgUAMAoGCCqGSM49BAMCA0kAMEYCIQDaHGOui+X2T44R6 GVpN7m2nEcr6T6sMjOhZ5NuSo1eqwIhAL1a+/hp88DKJ0sv3eT3FxWcs71xmbLKD/QJ3mWaqrJNMIIC7jCCAnWqAwI BAqIISW0vvzqY2pcwCqYIKoZIzj0EAwIwZzEbMBkGA1UEAwwSQXBwbGUqUm9vdCBDQSAtIEczMSYwJAYDVQQLDB1Bc HBsZSBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTETMBEGA1UECqwKQXBwbGUqSW5jLjELMAkGA1UEBhMCVVMwHhcNMTQ wNTA2MjM0NjMwWhcNMjkwNTA2MjM0NjMwWjB6MS4wLAYDVQQDDCVBcHBsZSBBcHBsaWNhdGlvbiBJbnR1Z3JhdGlvb $\verb|ibdQSAtIEczMSYwJAYDVQQLDB1BcHBsZSBDZXJ0aWZpY2F0aW9uIEF1dGhvcm10eTETMBEGA1UECgwKQXBwbGUgSW5| \\$ jLjELMAkGA1UEBhMCVVMwWTATBgcqhkjOPQIBBggqhkjOPQMBBwNCAATwFxGEGddkhdUaXiWBB3bogKLv3nuuTeCN/ EuT4TNW1WZbNa4i0Jd2DSJOe7oI/XYXzojLdrtmcL7I6CmE/1RFo4H3MIH0MEYGCCsGAQUFBwEBBDowODA2BggrBgE FBQcwAYYqaHR0cDovL29jc3AuYXBwbGUuY29tL29jc3AwNC1hcHBsZXJvb3RjYWczMB0GA1UdDqQWBBQj8knET5Pk7 yfmxPYobD+iu/0uSzAPBgNVHRMBAf8EBTADAQH/MB8GA1UdIwQYMBaAFLuw3qFYM4iapIqZ3r6966/ayySrMDcGA1U dHwQwMC4wLKAqoCiGJmh0dHA6Ly9jcmwuYXBwbGUuY29tL2FwcGxlcm9vdGNhZzMuY3JsMA4GA1UdDwEB/wQEAwIBB jAQBgoqhkiG92NkBgIOBAIFADAKBggqhkjOPQQDAgNnADBkAjA6z3KDURaZsYb7NcNWymK/9Bft2Q91TaKOvvGcgV5 QAAMYIBizCCAYcCAQEwqYYwejEuMCwGA1UEAwwlQXBwbGUqQXBwbGljYXRpb24qSW50ZWdyYXRpb24qQ0EqLSBHMzE AYTAlVTAqhoYPaZ2cynDzANBqlqhkqBZQMEAqEFAKCBlTAYBqkqhkiG9w0BCQMxCwYJKoZIhvcNAQcBMBwGCSqGSIb wYJKoZIhvcNAQkEMSIEICkrcfQNgJwC+ObwK2PDu2WvF+Itf+NH4k9YQpnZS8+WMAoGCCqGSM49BAMCBEYwRAIgJiP 5KbLUmKjCpcKrcOLHqmDe4WTnxOLpX09MtdAS1z0CIEyiKhZ1i+WRRwYH3fAnd+U21dAlj+FaeodSJQ/s5cJPAAAAA AAA\",\"version\":\"EC v1\",\"data\":\"ifGVeXI0UNe7VWqvU2x09FkNu/ibyRybqhrZs1rIxfRKlflwcm0 E61u7H6WtY6OZQVxWGO6WVsspGxLGMlCNwnLJvOb/Z9HkEIrR2bKmMGjzkUqL7xwIeTt7JRvrFN54sTqBKeMftLH8K lW06N6fiG//jfk+cqc4h2PJ4pOJuuQDwjRPOeCFDyxml3Xo6OYQhC//iFNyMAVUSnjtTx9uPhJDwPSpPALfrRjMyk5 vsJRkMa5dC0nDK9TIEUgbxqEVVcZniHz6/qMnacT0xyrUsiW2CVFhZ1uPTKRt2Gu5mIq4qHLGlP+zRjspFNSIoGirC 39K8Jq7NZRrvTjUI=\",\"header\":{\"ephemeralPublicKey\":\"MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQ gAEp70KFIRTiBv8aGErDvJC6F2fUwh8VuD5AGe7GMLoY8LFsB7qFSpLcAhw/k2KgQbbpudj1axSmVagn7VbHqcmJg= =\",\"publicKeyHash\":\"vjtnw1HBhuEUhYWDobbbWf6qxhj8kGxh0dLh045enV4=\",\"transactionId\":\ "29b13103be6794bcc2b64bc9f53d9d3032e3d7718b03a61e4723d3d50f855b8f\"}}

Configurable Alert Messages

Query to retrieve the existing Configurable Alert Messages

Use the query below to retrieve all the configured Worldpay alert messages.



```
Query
                                   Sample Response
query {
                                   {
 camSettings {
                                      "data": {
                                         "camSettings": {
   settings {
     accountLevelMessage {
                                            "settings": [
         setting {
                                                "accountLevelMessage": {
             list {
                                                   "setting": {
                   messageCode
                                                      "list": [
                   actualMessage
                   customMessage
                                                         "messageCode": "MCAM0",
           }
                                                         "actualMessage": "Please verify
                                   the Billing Address in your Address Book before
     accessWorldPayMessage {
                                   adding new card!",
                                                         "customMessage": ""
         setting {
             list {
                                                         },
                                                         {
                   messageCode
                   actualMessage
                                                         "messageCode": "MCAM1",
                                                         "actualMessage": "Token doesnot
                   customMessage
                                   exist anymore, Please delete the card.",
                                                         "customMessage": ""
                                                         },
     checkoutMessage {
                                                         "messageCode": "MCAM2",
         setting {
                                                         "actualMessage": "The card has
             list {
                   messageCode
                                   been updated.",
                   actualMessage
                                                         "customMessage": ""
                   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",
```



```
"actualMessage": "Detected
only whitespace character for code",
                         "customMessage": ""
           }
         },
        "checkoutMessage": {
           "setting": {
             "list": [
                        "messageCode": "CCAMO",
                        "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":
                            ]
                        }
                   }
               }
           ]
    }
}
```

Configurable Labels

Query to retrieve the existing Configurable Labels

Use the query below to retrieve all the configured Worldpay labels.



```
Query
                         Sample Response
query {
                         {
  customLabelsSetti
                             "data": {
ngs {
                                 "customLabelsSettings": {
        settings {
                                     "settings": [
           accountLa
                                         {
bels {
                                             "accountLabels": {
                 setti
                                                 "setting": {
ng {
                                                     "list": [
                     1
                                                         {
ist {
                                                             "labelCode": "AC1",
                                                             "defaultLabel": "Card Holder Name"
   labelCode
                                                             "customLabel": "Card Holder Name-
   defaultLabel
                         custom"
                                                        },
   customLabel
                     }
                                                             "labelCode": "AC2",
                                                             "defaultLabel": "Card Brand",
                                                             "customLabel": "Card Brand-custom"
             adminLabe
                                                         },
ls{
                                                         {
                 setti
                                                             "labelCode": "AC3",
                                                             "defaultLabel": "Card Number",
ng {
                                                             "customLabel": ""
                     1
ist {
                                                         },
                                                         {
   labelCode
                                                             "labelCode": "AC4",
                                                             "defaultLabel": "Card Expiry Month
   defaultLabel
                                                             "customLabel": ""
   customLabel
                                                        },
                     }
                                                             "labelCode": "AC5",
                 }
                                                             "defaultLabel": "Card Expiry Year"
             checkoutL
                                                             "customLabel": ""
abels{
                 setti
                                                        },
ng {
                                                             "labelCode": "AC6",
                     1
                                                             "defaultLabel": "Update",
ist {
                                                             "customLabel": ""
   labelCode
                                                        },
                                                             "labelCode": "AC7",
   defaultLabel
                                                             "defaultLabel": "Update Saved Card
   customLabel
                     }
                                                            "customLabel": ""
```



```
{
    }
                                                           "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
                      mer!",
                                                           "customLabel": ""
                                                       },
                                                            "labelCode": "AC16",
                                                            "defaultLabel": "Disclaimer!",
                                                           "customLabel": ""
                                                       },
                                                       {
```



```
"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": "CO1",
                                     "defaultLabel": "New Card",
                                     "customLabel": "New Card-custom"
                                 },
                                     "labelCode": "CO2",
                                     "defaultLabel": "We Accept",
                                     "customLabel": "We Accept-custom"
                                 },
                                     "labelCode": "CO3",
                                     "defaultLabel": "Card Number",
                                     "customLabel": ""
                                 },
                                     "labelCode": "CO4",
                                     "defaultLabel": "Card Holder Name"
                                     "customLabel": ""
                                 },
                                     "labelCode": "CO5",
```



```
"defaultLabel": "CVV",
                                     "customLabel": ""
                                 },
                                 {
                                     "labelCode": "CO6",
                                     "defaultLabel": "Month",
                                     "customLabel": ""
                                 },
                                 {
                                     "labelCode": "CO7",
                                     "defaultLabel": "Year",
                                     "customLabel": ""
                                 },
                                 {
                                     "labelCode": "CO8",
                                     "defaultLabel": "Save This Card",
                                     "customLabel": ""
                                 },
                                 {
                                     "labelCode": "CO9",
                                     "defaultLabel": "Important Disclai
mer!",
                                     "customLabel": ""
                                 },
                                     "labelCode": "CO10",
                                     "defaultLabel": "Pay Now",
                                     "customLabel": ""
                                 },
                                     "labelCode": "CO11",
                                     "defaultLabel": "MM/YY",
                                     "customLabel": ""
                                 },
                                     "labelCode": "CO12",
                                     "defaultLabel": "Saved cards",
                                     "customLabel": ""
                                 },
                                     "labelCode": "CO13",
                                     "defaultLabel": "Use Saved Card",
                                     "customLabel": ""
                                 },
                                     "labelCode": "CO14",
                                     "defaultLabel": "Place Order",
                                     "customLabel": ""
                                 },
                                 {
```



```
"labelCode": "CO15",
                                     "defaultLabel": "Saved Card featur
e will be available only if enabled by Merchant.",
                                     "customLabel": ""
                                },
                                 {
                                     "labelCode": "CO16",
                                     "defaultLabel": "Card Verification
 Number",
                                     "customLabel": ""
                                },
                                 {
                                     "labelCode": "CO17",
                                     "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.