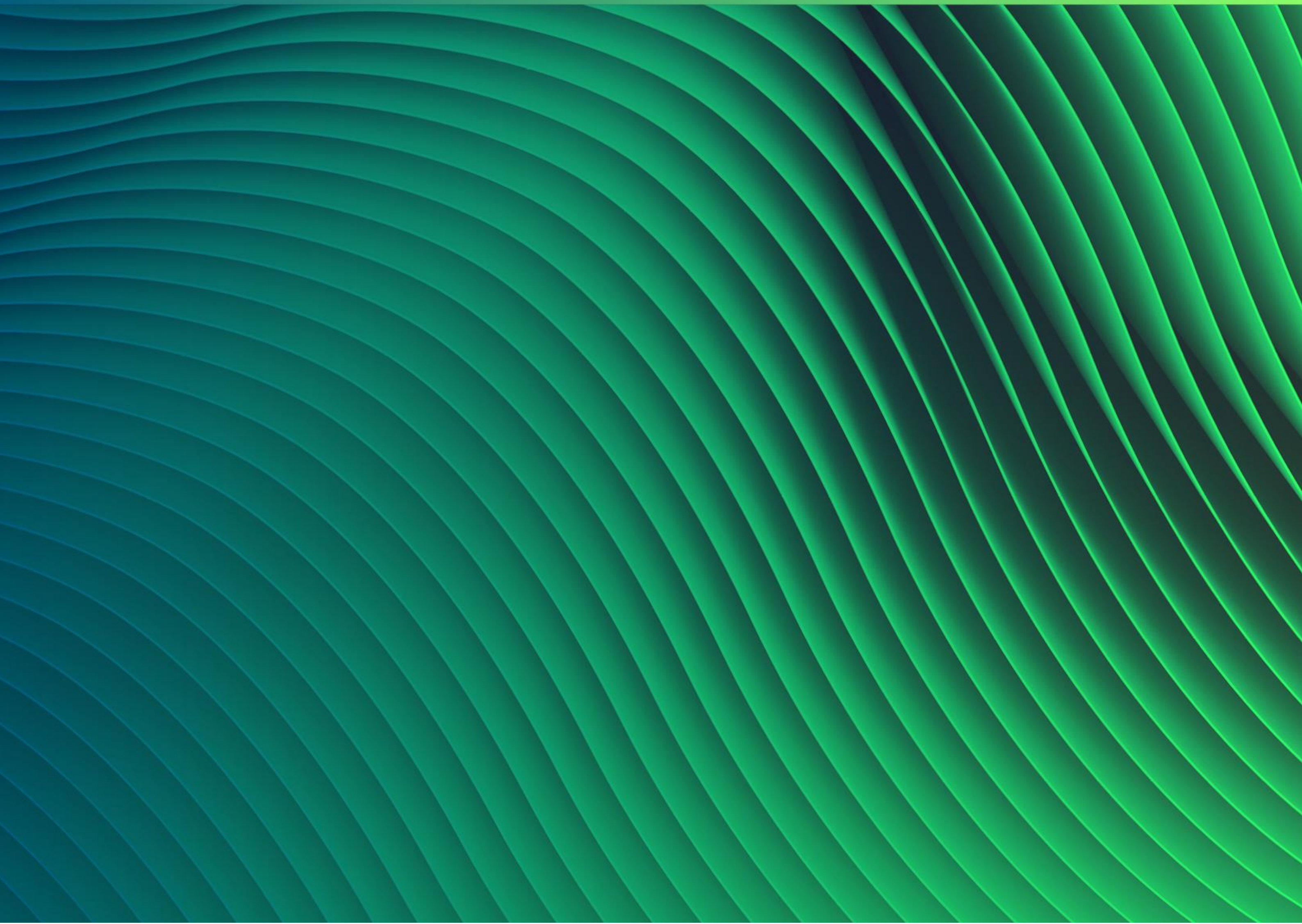
worldpay
from FIS

Magento 2 ACCESS WORLDPAY GraphQL API Guide



Contents

Introduction 4

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

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

Instant 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

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
Create Customer	One time activity to register new shopper with the createCustomer mutation. Once registered, you can create an authenticated token fo r existing shoppers using generateCustomerToken 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
Create an empty cart	Create customer cart (customerCart) for registered shoppers or use createemptycart 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 addSimpleProductsToCart 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 cart 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 placeOrder 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

Sample Request	Sample Response

`{`
`  customerCart{`
`    id`
`  }`
`}`
	`{`
	`    "data": {`
	`        "customerCart": {`
	`            "id": "sG1FYDc4oCdsXrU6RQyzlY5aAhTrv3w8"`
	`        }`
	`    }`
	`}`

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

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(
    input: {
      cart_id: "sG1FYDc4oCdsXrU6RQyzlY5aAhTrv3w8"
      cart_items: [
        {
          data: {
            quantity: 1
            sku: "24-MB02"
          }
        }
      ]
    }
) {
    cart {
      items {
        id
        product {
          name
          sku
        }
        quantity
	{
	"data": {
	"addSimpleProductsToCart": {
	"cart": {
	"items": [
	{
	"id": "57",
	"product": {
	"name": "Fusion Backpack",
	"sku": "24-MB02"
	},
	"quantity": 2
	}
	]
	}
	}
	}
	}

```
        }
      }
    }
}
```

4. Add a shipping address.

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

Sample Request

```
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
      }
    }
  }
}
```

Sample Response

```
{
"data": {
"setShippingAddressesOnCart": {
"cart": {
"shipping_addresses": [
{
"firstname": "Jacob",
"lastname": "Hunt",
"company": "Magento",
"street": [
"Magento Pkwy",
"Main Street"
],
"city": "Austin",
"postcode": "78758",
"telephone": "8675309"
}
]
}
}
}
}
```

5. Set the shipping method.

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

Sample Request

```
mutation {
  setShippingMethodsOnCart(
    input: {
      cart_id: "sG1FYDc4oCdsXrU6RQyzlY5aAhTrv3w8",
      shipping_methods: [
        {
          carrier_code: "tablerate"
          method_code: "bestway"
```

Sample Response

```
{
"data": {
"setShippingMethodsOnCart": {
"cart": {
"shipping_addresses": [
{
"selected_shipping_method": {
"carrier_code": "tablerate",
        }
      ]
    }
  ) {
    cart {
      shipping_addresses {
        selected_shipping_method {
          carrier_code
          carrier_title
          method_code
          method_title
          amount {
            value
            currency
          }
        }
      }
    }
  }
}
```

```
"carrier_title": "Best Way",
"method_code": "bestway",
"method_title": "Table Rate",
"amount": {
"value": 0,
"currency": "USD"
}
}
}
]
}
}
}
}
```

6. Add the billing address.

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

Sample Request

```
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
      }
    }
  }
}
```

Sample Response

```
{
    "data": {
        "setBillingAddressOnCart": {
            "cart": {
                "billing_address": {
                    "firstname": "Jacob",
                    "lastname": "Hunt",
                    "company": "Magento",
                    "street": [
                        "Magento Pkwy",
                        "Main Street"
                    ],
                    "city": "Austin",
                    "postcode": "78758",
                    "telephone": "8675309"
                }
            }
        }
    }
}
```

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
Set the Payment method using a new card	Mutation - setPaymentMethodOnCart Pre requisite – Set the Worldpay Integration Mode to Direct. Inputs - 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 shopper	Query - worldpayTokens 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.
Set the Payment method using the saved card (token only)	Mutation - setPaymentMethodOnCart Inputs - Code - "worldpay_cc", tokenID– token retrieved through worldpayTokens. All other input values of "setPaymentMethodOnCart" should be passed em pty string"".
Set Payment method using the saved card with token and cvc enabled	Mutation - setPaymentMethodOnCart Inputs - 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"".
Set the Payment method using saved card with token and cvcHref	Mutation - setPaymentMethodOnCart Inputs - 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"".

Set the Payment method for new card using token URL(tokenHref)	Mutation - setPaymentMethodOnCart Inputs - 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””.
Set the Payment method using Google Pay	Mutation - setPaymentMethodOnCart Inputs - Code - ”worldpay_wallets”, googlepayToken – Google Pay Token. All other input values of “setPaymentMethodOnCart” should be passed as an empty string””.
Set the Payment method using Apple Pay	Mutation - setPaymentMethodOnCart Inputs - 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 Worldpay Payment GraphQL.

8. Place the order.

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

Sample Request	Sample Response

```
mutation {
  placeOrder(
    input: {
      cart_id: "
sG1FYDc4oCdsXrU6RQyzlY5aAhTrv3w8"
    }
  ) {
    order {
      order_number
    }
  }
}
```
	```
	{
	  "data": {
	    "placeOrder": {
	      "order": {
	        "order_number": "000000006"
	      }
	    }
	  }
	}
	```

Worldpay Payment GraphQL

Set the Payment method using a new card

Mutation	Sample Request	Sample Response

setPaymentMethodOnCart
 	
*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 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
	```
	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: ""
	applepayToken: ""
	googlepayToken: ""
	cvcHref:""
	sessionHref:""
	tokenId:""
	}
	}
	}) {
	cart {
	selected_payment_method {
	code
	title
	}
	}
	}
	}
	```
		```
		{
		"data": {
		"setPaymentMethodOnCart":
		{
		 "cart": {

		"selected_payment_method"
		: {
		   "code": "worldpay_cc",
		   "title": "Credit Cards"
		    }
		   }
		  }
		 }
		}
		```

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 {`
    `cards {`
      `id`
      `tokenid`
      `cardnumber`
      `cardholdername`
      `cardexpirymonth`
      `cardexpiryyear`
      `method`
    `}`
  `}`
`}`
	`{`
	    `"data": {`
	        `"worldpayTokens": {`
	            `"cards": [`
	                `{`
	                    `"id": 5,`
	                    `"tokenid": "9905143360240686982",`
	                    `"cardnumber": "3434******3434",`
	                    `"cardholdername": "3D.Authorised",`
	                    `"cardexpirymonth": 10,`
	                    `"cardexpiryyear": 2025,`
	                    `"method": "worldpay_cc"`
	                `}`
	            `]`
	        `}`
	    `}`
	`}`

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

Mutation	Sample Request	Sample Response

setPaymentMethodOnCart
 	
***Note**
`cart_id: Refer create empty cart mutation`
`tokenId: tokenId from response of worldpayTokens query`
	`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`
		`{`
		`"data": {`
		`"setPaymentMethodOnCart":`
		`{`
		 `"cart": {`
		
		`"selected_payment_method"`
		`: {`
		   `"code": "worldpay_cc",`
		   `"title": "Credit Cards"`
		    `}`
		   `}`
		 `}`
		`}`
		`}`

	} } } }	

Set Payment method using the saved card with cvc enabled

Mutation	Sample Request	Sample Response

setPaymentMethodOnCart
 	
*Note
`cart_id: Refer create empty cart mutation`
 	
`tokenId: tokenId from response of worldpayTokens query`
 	
`cvc: enter cvc/cvv number.`
	`mutation {`
	`setPaymentMethodOnCart(input: {`
	`cart_id: "sG1FYDc4oCdsXrU6RQyzlY5aAhTrv3w8"`
	`payment_method: {`
	`code: "worldpay_cc"`
	`worldpay_cc: {`
	`cc_name: ""`
	`cc_number: ""`
	`cc_exp_month: ""`
	`cc_exp_year: ""`
	`cvc: "123"`
	`save_card: ""`
	`tokenUrl: ""`
	`applepayToken: ""`
	`googlepayToken: ""`
	`cvcHref: ""`
	`sessionHref:""`
	`tokenId:" 9905143360240686982"`
	`}`
	`}`
	`}) {`
	`cart {`
	`selected_payment_method {`
	`code`
	`title`
	`}`
	`}`
	`}`
	`}`
		`{`
		`"data": {`
		`"setPaymentMethodOnCart":`
		`{`
		`"cart": {`
		
		`"selected_payment_method"`
		`: {`
		`"code": "worldpay_cc",`
		`"title": "Credit Cards"`
		`}`
		`}`
		`}`
		`}`
		`}`

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

setPaymentMethodOnCart
 	
*Note
cart_id: Refer create empty cart mutation
 	
tokenId: tokenID from response of worldpayTokens query
 	
cvcHref: pass the cvcHref returned from Worldpay websdk.
	`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:"https://try.access.worldpay.com`
	`/sessions/`
	`eyJrIjoxLCJkIjoiWUsyaDVxV0lhblhUcVB`
	`GZW1RVmpHdkx6dU1S`
	`bFcwY2JMbHVVNE9VZ1Bhc2NNT0tCR3QyYVEy`
	`ZDBCTWRiUDN4YVhMYnh1WnB4TGhtT0s4N1RySnFhcFE9P`
	`SJ9"`
	`sessionHref:""`
	`tokenId:"9905143360240686982"`
	`}`
	`}`
	`}) {`
	`cart {`
	`selected_payment_method {`
	`code`
	`title`
	`}`
	`}`
	`}`
	`}`
		`{`
		`"data": {`
		`"setPaymentMethodOnCart`
		`": {`
		`"cart": {`
		
		`"selected_payment_metho`
		`d": {`
		`"code": "worldpay_cc",`
		`"title": "Credit Cards"`
		`}`
		`}`
		`}`
		`}`
		`}`

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

setPaymentMethodOnCart
 	
*Note
cart_id: Refer create empty cart mutation
 	
tokenUrl: pass the token url obtained by creating a verified token.
 	
save_card: "1", saves the card
save_card: "0" does not save the card
	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: "
	https://try.access.worldpay.com/tokens/
	eyJrIjoxLCJkIjoiZkY0YklGT0xGWjR
	PWE9ZVWwrNFl5djdodnBpVzdaY3VORVRUUGgwMm1Z
	c0JHTTgzUWdNcC9sV0Zjb1lXSHJ6cyJ9
	"
	applepayToken: ""
	googlepayToken: ""
	cvcHref:""
	sessionHref:""
	tokenId:""
	}
	}
	}) {
	cart {
	selected_payment_method {
	code
	title
	}
	}
	}
	}
		{
		"**data**": {
		"setPaymentMethodOnCart":
		{
		"cart": {
		
		"selected_payment_method"
		: {
		"code": "worldpay_cc",
		"title": "Credit Cards"
		}
		}
		}
		}
		}

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

		"payment_method_code": "worldpay_cc", "type": "card" }] } } }

List all Worldpay saved cards	query {
	worldpayTokens {
	cards {
	id
	tokenid
	cardnumber
	cardholdername
	cardexpirymonth
	cardexpiryyear
	method
	}
	}
	}
		{
		"data": {
		"worldpayTokens": {
		"cards": [
		{
		"id": 5,
		"tokenid": "9905143360240686982",
		"cardnumber": "3434******3434",
		"cardholdername": "3D.Authorised",
		"cardexpirymonth": 10,
		"cardexpiryyear": 2025,
		"method": "worldpay_cc"
		}
		]
		}
		}
		}

Update card	mutation {
	updateWorldpayToken (
	id: 5,
	input : {
	tokenid: "9910076467795747092"
	cardholdername: "John"
	cardexpirymonth: 10
	cardexpiryyear: 2025
	}
	) {
	card {
	id
	tokenid
	cardnumber
	cardholdername
	cardexpirymonth
	cardexpiryyear
	method
	}
	}
	}
		{
		"data": {
		"updateWorldpayToken": {
		"card": {
		"id": 5,
		"tokenid": "9910076467795747092",
		"cardnumber": "6011********0000",
		"cardholdername": "John",
		"cardexpirymonth": 10,
		"cardexpiryyear": 2025,
		"method": "worldpay_cc"
		}
		}
		}
		}

Delete card	
	```
	mutation {
	deleteWorldpayToken (
	id: 3
	tokenid: "9905143360240686982"
	){
	result
	}
	}
	```
		```
		{
		"data": {
		"deleteWorldpayToken": {
		"result": true
		        }
		    }
		}
		```

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 {
settings {
enabled
paymentmethods {method}
supportedauthentication {option}
gatewayname
gatewaymerchantid
googlemerchantid
googlemerchantname
}
}
}
	"data": {
	"googlepaySettings": {
	"settings": [
	{
	"enabled": 1,
	"paymentmethods": {
	"method": [
	"AMEX",
	"VISA",
	"DISCOVER",
	"JCB",
	"MASTERCARD"
	]
	},
	"supportedauthentication": {
	"option": [
	"PAN_ONLY",
	"CRYPTOGRAM_3DS"
	]
	},
	"gatewayname": "worldpay",
	"gatewaymerchantid": "627392fb1cafcb1",
	"googlemerchantid": null,
	"googlemerchantname": "worldpay"
	}
	]
	}
	}
	}

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

setPaymentMethodOnCart	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: ""
		{
		"data": {
		"setPaymentMethodOnCart":
		{
		"cart": {
		"selected_payment_method":
		{
		"code": worldpay_wallets",
		"title": "Wallets"
		}
		}
		}

	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+l4iuw6udckjNRUq457VrNjr7qWz+lBgToqi3qczCERYoj1tnf3EmOblgaSKpWtcyY7UMyfYUE3ty2mk3pJ
m+8YvdhpkAyN/glD5sWR9zBAYsc5qeAt4MazZGedlGTbvGhUfCN0PUXRvX2014pr4tKrxBeaI+iT+6ijRf2sjxpYp/
UbUXtXNxqZun2vYuKeiPhiaFfvBUv2woOUxWHLJSNqoZ2Uek+qDsKuxZeFsctJV+tHzhVJW95skVWjmnqyzPIX2JCf
Wg9R65FYN+ib8DM1xxn6XrPK6CA\\\u003d\\\u003d\\\",\\\"ephemeralPublicKey\\\":\\\"BHcOlQC4g
ttnXQtMUCef1Jw/X44fvEKS2YpvNBNzNl/qFUiArVPqRd+D+JsndddzWXuqXtWq01DBZfY0yE75y6w\\\u003d\\\
",\\\"tag\\\":\\\"TNgnyjs878KS8UGHz1jQriU/+MOSz4EX1tQL0HccM/s\\\u003d\\\"}\"}
```

Apple Pay

Query to retrieve Apple Pay Settings

Sample Request	Sample Response

query {
applepaySettings {
settings {
enabled
certificationkey
certificationcrt
certificationpassword
merchantname
domainname
}
}
}
	{
	"data": {
	"applepaySettings": {
	"settings": [
	{
	"enabled": 1,
	"certificationkey": " ",
	"certificationcrt": " ",
	"certificationpassword": " ",
	"merchantname": "worldpay",
	"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: {`
	`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`
	`}`
	`}`
	`}`
	`}`
		`{`
		`"data": {`
		`"setPaymentMethodOnCart":`
		`{`
		`"cart": {`
		`"selected_payment_method":`
		`{`
		`"code": worldpay_wallets",`
		`"title": "Wallets"`
		`}`
		`}`
		`}`
		`}`
		`}`

Sample Apple Pay Token

```
{\"signature\":\"MIAGCSqGSIb3DQEHAqCAMIACAQExDzANBglghkgBZQMEAgEFADCABgkqhkiG9w0BBwEAAKCAM
IID5jCCA4ugAwIBAgIIaGD2mdnMpw8wCgYIKoZIzj0EAwIwejEuMCwGA1UEAwwlQXBwbGUgQXBwbGljYXRpb24gSW5
0ZWdyYXRpb24gQ0EgLSBHMzEmMCQGA1UECwwdQXBwbGUgQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkxEzARBgNVBAoMC
kFwcGxlIEluYy4xCzAJBgNVBAYTAlVTMB4XDTE2MDYwMzE4MTY0MFoXDTIxMDYwMjE4MTY0MFowYjEoMCYGA1UEAww
fZWNjLXNtcC1icm9rZXItc2lnbl9VQzQtU0FOREJPWDEUMBIGA1UECwwLaU9TIFN5c3RlbXMxEzARBgNVBAoMCkFwc
GxlIEluYy4xCzAJBgNVBAYTAlVTMFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEgjD9q8Oc914gLFDZm0US5jfiqQH
dbLPgsc1LUmeY+M9OvegaJajCHkwz3c6OKpbC9q+hkwNFxOh6RCbOlRsSlaOCAhEwggINMEUGCCsGAQUFBwEBBDkwN
zA1BggrBgEFBQcwAYYpaHR0cDovL29jc3AuYXBwbGUuY29tL29jc3AwNC1hcHBsZWFpY2EzMDIwHQYDVR0OBBYEFAI
kMAua7u1GMZekplopnkJxghxFMAwGA1UdEwEB/wQCMAAwHwYDVR0jBBgwFoAUI/JJxE+T5O8n5sT2KGw/orv9Lkswg
gEdBgNVHSAEggEUMIIBEDCCAQwGCSqGSIb3Y2QFATCB/jCBwwYIKwYBBQUHAgIwgbYMgbNSZWxpYW5jZSBvbiB0aGl
zIGNlcnRpZmljYXRlIGJ5IGFueSBwYXJ0eSBhc3N1bWVzIGFjY2VwdGFuY2Ugb2YgdGhlIHRoZW4gYXBwbGljYWJsZ
SBzdGFuZGFyZCB0ZXJtcyBhbmQgY29uZGl0aW9ucyBvZiB1c2UsIGNlcnRpZmljYXRlIHBvbGljeSBhbmQgY2VydGl
maWNhdGlvbiBwcmFjdGljZSBzdGF0ZW1lbnRzLjA2BggrBgEFBQcCARYqaHR0cDovL3d3dy5hcHBsZS5jb20vY2Vyd
GlmaWNhdGVhdXRob3JpdHkvMDQGA1UdHwQtMCswKaAnoCWGI2h0dHA6Ly9jcmwuYXBwbGUuY29tL2FwcGxlYWljYTM
uY3JsMA4GA1UdDwEB/wQEAwIHgDAPBgkqhkiG92NkBh0EAgUAMAoGCCqGSM49BAMCA0kAMEYCIQDaHGOui+X2T44R6
GVpN7m2nEcr6T6sMjOhZ5NuSo1egwIhAL1a+/hp88DKJ0sv3eT3FxWcs71xmbLKD/QJ3mWagrJNMIIC7jCCAnWgAwI
BAgIISW0vvzqY2pcwCgYIKoZIzj0EAwIwZzEbMBkGA1UEAwwSQXBwbGUgUm9vdCBDQSAtIEczMSYwJAYDVQQLDB1Bc
HBsZSBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTETMBEGA1UECgwKQXBwbGUgSW5jLjELMAkGA1UEBhMCVVMwHhcNMTQ
wNTA2MjM0NjMwWhcNMjkwNTA2MjM0NjMwWjB6MS4wLAYDVQQDDCVBcHBsZSBBcHBsaWNhdGlvbiBJbnRlZ3JhdGlvb
iBDQSAtIEczMSYwJAYDVQQLDB1BcHBsZSBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTETMBEGA1UECgwKQXBwbGUgSW5
jLjELMAkGA1UEBhMCVVMwWTATBgcqhkjOPQIBBggqhkjOPQMBBwNCAATwFxGEGddkhdUaXiWBB3bogKLv3nuuTeCN/
EuT4TNW1WZbNa4i0Jd2DSJOe7oI/XYXzojLdrtmcL7I6CmE/1RFo4H3MIH0MEYGCCsGAQUFBwEBBDowODA2BggrBgE
FBQcwAYYqaHR0cDovL29jc3AuYXBwbGUuY29tL29jc3AwNC1hcHBsZXJvb3RjYWczMB0GA1UdDgQWBBQj8knET5Pk7
yfmxPYobD+iu/0uSzAPBgNVHRMBAf8EBTADAQH/MB8GA1UdIwQYMBaAFLuw3qFYM4iapIqZ3r6966/ayySrMDcGA1U
dHwQwMC4wLKAqoCiGJmh0dHA6Ly9jcmwuYXBwbGUuY29tL2FwcGxlcm9vdGNhZzMuY3JsMA4GA1UdDwEB/wQEAwIBB
jAQBgoqhkiG92NkBgIOBAIFADAKBggqhkjOPQQDAgNnADBkAjA6z3KDURaZsYb7NcNWymK/9Bft2Q91TaKOvvGcgV5
Ct4n4mPebWZ+Y1UENj53pwv4CMDIt1UQhsKMFd2xd8zg7kGf9F3wsIW2WT8ZyaYISb1T4en0bmcubCYkhYQaZDwmSH
QAAMYIBizCCAYcCAQEwgYYwejEuMCwGA1UEAwwlQXBwbGUgQXBwbGljYXRpb24gSW50ZWdyYXRpb24gQ0EgLSBHMzE
mMCQGA1UECwwdQXBwbGUgQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkxEzARBgNVBAoMCkFwcGxlIEluYy4xCzAJBgNVB
AYTAlVTAghoYPaZ2cynDzANBglghkgBZQMEAgEFAKCBlTAYBgkqhkiG9w0BCQMxCwYJKoZIhvcNAQcBMBwGCSqGSIb
3DQEJBTEPFw0yMTAyMDEwNjQwMzRaMCoGCSqGSIb3DQEJNDEdMBswDQYJYIZIAWUDBAIBBQChCgYIKoZIzj0EAwIwL
wYJKoZIhvcNAQkEMSIEICkrcfQNgJwC+ObwK2PDu2WvF+Itf+NH4k9YQpnZS8+WMAoGCCqGSM49BAMCBEYwRAIgJiP
5KbLUmKjCpcKrcOLHqmDe4WTnxOLpX09MtdASlz0CIEyiKhZ1i+WRRwYH3fAnd+U21dAlj+FaeodSJQ/s5cJPAAAAA
AAA\",\"version\":\"EC_v1\",\"data\":\"ifGVeXI0UNe7VWqvU2x09FkNu/ibyRybghrZs1rIxfRKlflwcm0
E61u7H6WtY6OZQVxWGO6WVsspGxLGMlCNwnLJvOb/Z9HkEIrR2bKmMGjzkUqL7xwIeTt7JRvrFN54sTqBKeMftLH8K
exWA2NQ7TlnfBwtP6ra20JhWkNGAIPDnDKZ0O7p7kaTJfTQp+Mr+LXye+xjMvfoWOXc88tJWthYup55nh8MFdGHHI5
lW06N6fiG//jfk+cqc4h2PJ4pOJuuQDwjRPOeCFDyxml3Xo6OYQhC//iFNyMAVUSnjtTx9uPhJDwPSpPALfrRjMyk5
vsJRkMa5dC0nDK9TIEUgbxgEVVcZniHz6/qMnacT0xyrUsiW2CVFhZ1uPTKRt2Gu5mIg4gHLGlP+zRjspFNSIoGirC
39K8Jq7NZRrvTjUI=\",\"header\":{\"ephemeralPublicKey\":\"MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQ
gAEp7OKFIRTiBv8aGErDvJC6F2fUwh8VuD5AGe7GMLoY8LFsB7qFSpLcAhw/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

```
query {
  camSettings {
   settings {
     accountLevelMessage {
         setting {
             list {
                   messageCode
                   actualMessage
                   customMessage
               }
           }
       }
     accessWorldPayMessage {
         setting {
             list {
                   messageCode
                   actualMessage
                   customMessage
               }
           }
       }
     checkoutMessage {
         setting {
             list {
                   messageCode
                   actualMessage
                   customMessage
               }
           }
       }
   }
  }
}
```

Sample Response

```
{
   "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",
                         "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":
""
                                }
                            ]
                        }
                    }
                }
            ]
        }
    }
}
```

Configurable Labels

Query to retrieve the existing Configurable Labels

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

Query	Sample Response

`query {`
`customLabelsSettings {`
`settings {`
`accountLabels {`
`setting {`
`list {`
`labelCode`
`defaultLabel`
`customLabel`
`}`
`}`
`}`
`adminLabels{`
`setting {`
`list {`
`labelCode`
`defaultLabel`
`customLabel`
`}`
`}`
`}`
`checkoutLabels{`
`setting {`
`list {`
`labelCode`
`defaultLabel`
`customLabel`
`}`
`}`
	`{`
	`"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": ""`
	`},`

```
                }
            }
        }
}
```

```
                                {
                                    "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.