

Google Pay API PaymentRequest Tutorial

The Google Pay API can be used in conjunction with other supported payment methods in [PaymentRequest](#).

Important: Chrome for Android is currently the only web browser supporting the Payment Request API with third-party payment methods, including Google Pay. Sites wishing to implement Google Pay as a standalone payment method should use the [Google Pay API JavaScript client library](#) with a [branded Google Pay button](#) to reach a broader Google audience across multiple browsers and platforms.

Step 1: Create a PaymentDataRequest object

Configure a [PaymentDataRequest](#) object specifying your payment gateway and accepted methods of payment. We recommend you obtain shipping address, email address, and phone number through the browser's `PaymentRequest options` argument, not the Google Pay API, for consistency across payment methods. Use the Google Pay API to receive only the payment credentials and optional billing address.

```
var googlePaymentDataRequest = {  
  apiVersion: 1,  
  // set a TEST environment for initial testing  
  environment: 'TEST',  
  // a merchant ID is available after approval by Google  
  merchantId: '01234567890123456789',  
  paymentMethodTokenizationParameters: {  
    tokenizationType: 'PAYMENT_GATEWAY',  
    // tokenization parameters are provided by your gateway  
    parameters: {}  
  },  
  allowedPaymentMethods: ['CARD', 'TOKENIZED_CARD'],  
  cardRequirements: {  
    allowedCardNetworks: ['AMEX', 'DISCOVER', 'MASTERCARD', 'VISA']  
  }  
};
```



Step 2: Declare Google Pay API support

Add the Google Pay API payment method identifier and its configuration to the `methodData` parameter passed to `PaymentRequest`.

```
var methodData = [
  {supportedMethods: 'https://google.com/pay', data: googlePaymentDataRequest},
  {supportedMethods: 'basic-card'}
];
```



Putting it all together

The following example shows an implementation of the Google Pay API with other supported payment methods in `PaymentRequest`:

```
<div id="checkout">
  <button id="buyButton">Checkout</button>
</div>

<script>
var allowedCardNetworks = ['AMEX', 'DISCOVER', 'MASTERCARD', 'VISA'];

if (window.PaymentRequest) {
  var request = createPaymentRequest();

  request.canMakePayment()
    .then(function(result) {
      if (result) {
        // Display PaymentRequest dialog on interaction with the existing check
        document.getElementById('buyButton')
          .addEventListener('click', onBuyClicked);
      }
    })
    .catch(function(err) {
      showErrorForDebugging(
        'canMakePayment() error! ' + err.name + ' error: ' + err.message);
    });
} else {
  showErrorForDebugging('PaymentRequest API not available.');
```



```

        handlePaymentResponse(response);
    })
    .catch(function(err) {
        showErrorForDebugging(
            'show() error! ' + err.name + ' error: ' + err.message);
    });
}

/**
 * Define your unique Google Pay API configuration
 *
 * @returns {object} data attribute suitable for PaymentMethodData
 */
function getGooglePaymentsConfiguration() {
    return {
        // A merchant ID is available after approval by Google.
        // 'merchantId': '01234567890123456789',
        environment: 'TEST',
        apiVersion: 1,
        allowedPaymentMethods: ['CARD', 'TOKENIZED_CARD'],
        paymentMethodTokenizationParameters: {
            tokenizationType: 'PAYMENT_GATEWAY',
            // Check with your payment gateway on the parameters to pass.
            // @see {@link https://developers.google.com/pay/api/web/reference/object#G
            parameters: {}
        },
        cardRequirements: {allowedCardNetworks: allowedCardNetworks}
    };
}

/**
 * Create a PaymentRequest
 *
 * @returns {PaymentRequest}
 */
function createPaymentRequest() {
    // Add support for the Google Pay API.
    var methodData = [{
        supportedMethods: 'https://google.com/pay',
        data: getGooglePaymentsConfiguration()
    }];
    // Add other supported payment methods.
    methodData.push({
        supportedMethods: 'basic-card',
        data: {
            supportedNetworks:
                Array.from(allowedCardNetworks, (network) => network.toLowerCase())
        }
    });
}

```

```

    });

    var details = {
        total: {label: 'Test Purchase', amount: {currency: 'USD', value: '1.00'}}
    };

    var options = {
        requestPayerEmail: true,
        requestPayerName: true
    };

    return new PaymentRequest(methodData, details, options);
}

/**
 * Process a PaymentResponse
 *
 * @param {PaymentResponse} response returned when a user approves the payment re
 */
function handlePaymentResponse(response) {
    var formattedResponse = document.createElement('pre');
    formattedResponse.appendChild(
        document.createTextNode(JSON.stringify(response.toJSON(), null, 2)));
    document.getElementById('checkout')
        .insertAdjacentElement('afterend', formattedResponse);
}

/**
 * Display an error message for debugging
 *
 * @param {string} text message to display
 */
function showErrorForDebugging(text) {
    var errorDisplay = document.createElement('code');
    errorDisplay.style.color = 'red';
    errorDisplay.appendChild(document.createTextNode(text));
    var p = document.createElement('p');
    p.appendChild(errorDisplay);
    document.getElementById('checkout').insertAdjacentElement('afterend', p);
}

</script>

```

Payment method parameter list

The Google Pay API configuration passed in `PaymentRequest` is similar to the `PaymentDataRequest` object used by the Google Pay API JavaScript client library with a few exceptions:

- Set an `apiVersion` property value of 1 to associate your data format with a known API version
- An `environment` property should be set on the object matching a string value described in `PaymentOptions`
- The `transactionInfo` property should be omitted. Total price and currency should instead be specified in the `details` argument passed to `PaymentRequest`

Additional object properties specified for the Google Pay API payment method in a `PaymentRequest` payment method data property:

Property	Type	Necessity	Description
<code>apiVersion</code>	number	required	Specifies the data format version. Set to 1 for the current Google Pay API.
<code>environment</code>	string	optional	If set to 'TEST', returns dummy, non-chargeable credentials.

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

Last updated May 16, 2018.