Payment Page
Integration Guide

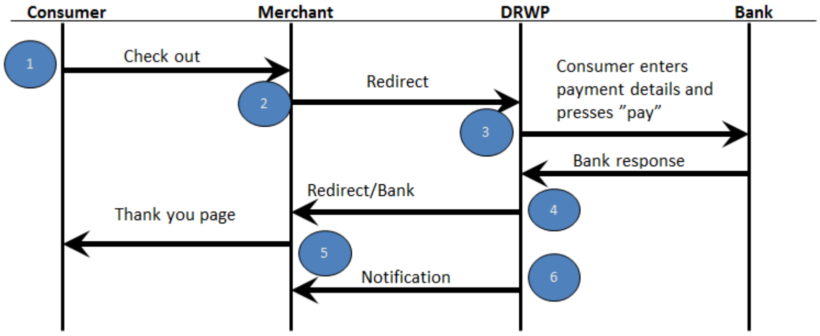
Contents

Revision History

Revision	Date	Detailed Description
3.3.0	2014-02-25	SEPA Direct Debit release
3.3.1	2014-04-09	Added request parameters to appendix A
3.3.2	2014-10-15	Changed recommendations regarding iframes.
3.3.3	2015-02-24	Added information about Client Side Encryption and REST API.
3.3.4	2015-03-17	Available markets from Digital River World Payment: Added Dotpay payment method for Poland Updated payment methods for Finland
3.3.5	2015-03-27	Added all available response parameters from the Payment Page Server.
3.3.6	2015-04-30	Updated the RSA keypair command in chapter “Step 5 – Moving to Production” to support 2048 bit keys.
4.0.0	2015-05-22	Added command to generate 2048 size RSA keypair in chapter “Java Example” to support 2048 bit keys.
4.1.0	2015-06-08	Added command to generate 2048 size RSA keypair in chapter “.NET API Example” to support 2048 bit keys. Added initialization instruction for 2048 size RSA keypair in .NET reference implementation.
4.1.1	2015-09-29	Updated 3D Secure related documentation.
4.1.2	2015-10-28	Updated the names of some Response Parameters in Appendix A to align them with the naming in the Payment Page Java and .Net APIs.
4.1.3	2015-12-07	Some corrections done to the request and response parameters in Appendix A.
4.1.4	2016-02-23	Added clarifications about the test keys and certificates included with the Start-up kit and clarified that 1024 bit keys are deprecated from now on.
4.1.5	2016-03-10	Clarified how to use the Java API in the Java Reference Implementation chapter.
4.1.6	2016-04-11	Updated the .Net API integration example.
4.1.7	2017-02-16	Added information for request parameter authorizationType in Appendix A
4.1.8	2017-06-08	Added information for request parameter authenticationRedirect in Appendix A

Introduction

Digital River World Payments' Payment Page is a web-based solution for accepting cards and alternative payment methods for e-commerce businesses. When checking out from your web store, the consumer is redirected to a branded payment page hosted on one of Digital River's World Payments' servers. The customer enters the payment details and completes the payment. The consumer is then returned to your web store with the payment transaction status. The financial credentials never need to touch your site.



Behind the scenes, when redirected to the Payment Page server, the consumer is presented with a payment selection web page. This can be customized through Page Builder, a content management wizard optimized for payment pages available via Digital River's self-serve remote control Digital River Command Console (DRCC) web interface.

To assist the consumer in completing a payment there is also built-in support for server side retry-handling, where the payment page is displayed again for declined transactions to allow the consumer to correct their details or select another payment method instead of being redirected back to the merchant.

If redirecting to a 3rd party payment method during a session in the Payment Page flow, the contact with the consumers may be lost (e.g. they close their browser). Therefore, an outgoing host-to-host notification when an order status has changed state is available. Depending on the nature of the payment method, the final status of a payment may not be known when the consumer comes back to the merchant's site. Digital River World Payments can poll 3rd party payment method providers for updates and send outgoing notifications to the merchant. Digital River World Payment s also provides an alternative where a merchant may pull order information using a Web Service Interface. The information regarding successful or failed payments is also available in Digital River World Payments standard reports and in the Digital River Command Console web interface.

Before you start

Recommended skillset

In order to successfully implement Payment Page, you need the following skill set:

- Basic programming skills in Java or other selected programming language
- Client certificate management
- Style sheet (CSS) skills for advanced branding of a page

Make sure you have all required information at hand

- The Payment Page Start Kit. The Start Kit is a collection of things needed to get started with the Payment Page integration. Currently it contains the following items:
    - PaymentPage_Integration_Manual.pdf (this document)
    - test_keys.zip (contains all keys/certificates needed for testing)
    - PaymentPageCertificationDocument.pdf (test specification for the certification)
    - TestData.xls (A customized version will be provided to you before certification)
    - PaymentPageAPI.jar (for Java) or NetgiroClient.dll (for .NET); A reference implementation used for creating a correctly formatted URL to Payment Page and unpacking the response from Payment Page. The source code is also available for merchants that which to create their own implementation.
    - Payment_Page_Security_Features_Digital_River.pdf (Describes the security features applied on sensitive data managed and transmitted by the Payment Page. This is of special interest for merchants that do not wish to use the reference implementation)
- A test Merchant ID which has been configured based on Digital River World Payments' set-up form.
- The URL for Digital River World Payments' Payment Page test environment (see Appendix).
- A user and URL to Digital River World Payments' online Command Console (DRCC) with access to Page Builder and transaction searches.

Integrate with Payment Page in 5 steps

The following two pages serve as a quick guide of how to integrate with Payment Page. Further details of each step are provided in the following sections of this document.

1. Review configuration and solution design options with Digital River World Payment

The Payment Page product allows certain flexibility in terms of how the system handles 3D Secure, timeouts, stored tokens etc. Get familiar with the different payment method families and how that may impact your client-side implementation. If your solution needs to interface with Digital River World Payments outside of Payment Page, you should also review those needs with Digital River World Payments.

2. Client side implementation

1. **Get familiar with the Payment Page Java or .NET reference implementation.** This will help you build and encrypt the URL used for payment initialization and unpack the response.
2. **Install the key store (and trust store if using .Net)** provided in the Payment Page start kit in your test environment
3. **Code your checkout process** to integrate with the Payment Pages hosted on Digital River World Payments' servers
   i. Create the redirect URL using the CreateRedirectUrl method in the Java reference implementation or createPaymentPageUrl method in the .Net reference implementation to specify which template you want to display, the order value to be paid and other required parameters.

   ii. Using your web server, redirect the consumer's browser to Digital River World Payments' servers.

   iii. Interpret the response coming back from Digital River World Payments using the `UnpackResponse` method and update your order status accordingly.

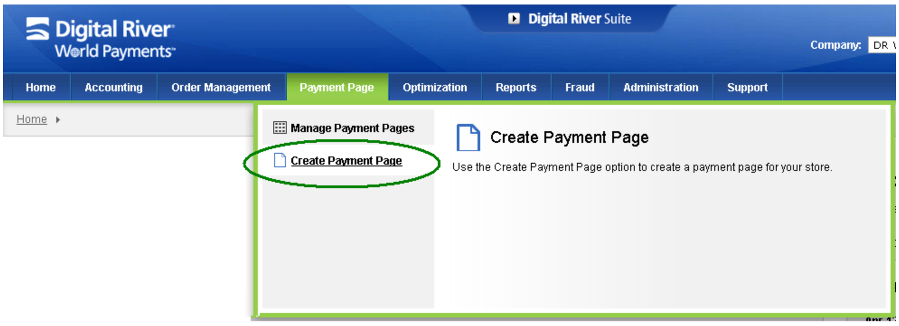
iv. Design and display a result page to be displayed on your server with relevant consumer message.

4. **Optionally implement support for receiving notifications**

    i. Check with your integration manager for specifications.

3. Customize the payment pages to fit your needs

1. Log in to DRCC and go to the Payment Page menu to create a page via Page Builder



Create Payment Page

Use the Create Payment Page option to create a payment page for your store.

2. Start testing the payment page integration via your site by redirecting a browser to the newly created page, specified via the template reference parameter.

4. Testing and Certification

1. Once you are satisfied with the result of your own tests, notify Digital River World Payments that you are ready for certification.

2. Digital River World Payments certifies the solution using relevant test cases that will be provided to you.

5. Moving to Production

1. **Create a private and public key pair and a certificate signing request**

2. Digital River World Payments will provide you with a **production package** including a new Production Merchant ID (MID), the signed certificate and Digital River World Payments' public certificate

3. **Move your code to your production systems, including change of parameters such as Merchant ID, URL and keys.**

4. **Install the production certificates from Digital River World Payments in your production environment**

5. Verify that **everything works by performing one or more test transactions**

6. Go live!

Reference Guide

Step 1 - Configuration and solution design

The focus of this section is to introduce you to the various payment methods and how they behave in Payment Page. This will give you an idea of how we should set up the system and will also give you some implementation hints.

Payment methods: Basic flows

The Payment Page flow is slightly dependent on the selected payment method selected since the underlying payment methods differ in behavior. However, you can build a future-proof implementation of Payment Page if you consider those behaviors from the outset.

Payment Method type	Description
Cards	Real-time response from Payment Page, immediate confirmation of payment
Consumer redirect payment methods, e.g. Internet Bank Payments and e-wallets	Real-time response from Payment Page, usually immediate confirmation of payment but sometimes delayed since the user may close the browser instead of coming back
Offline consumer push payments, such as bank transfers	Response from Payment Page will always indicate delayed confirmation of payment
Payouts	Response from Payment Page will always indicate delayed confirmation of refund.

Our recommendation is to implement the following high level steps for all payment methods

1. Redirect to Payment Page

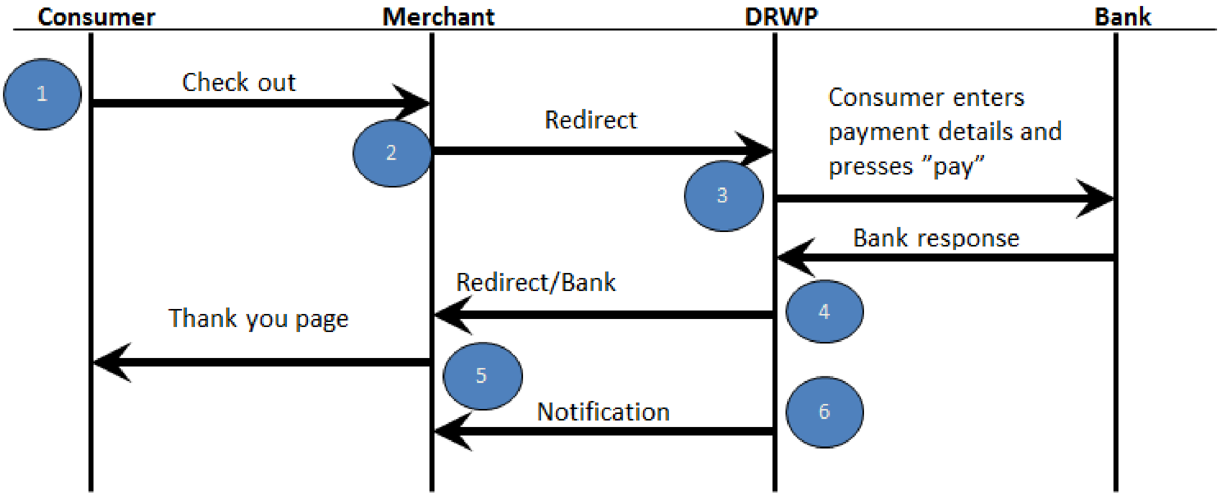
2. Wait for the response with the current status.

3. If current status is not final, wait for the notification, report file or query Digital River World Payments Web Service Interface to get the final confirmation before releasing the product or service they paid for

Below you will find the Payment Page flow illustrated for four alternative payment methods. For a current list of available payment methods within each family, contact Digital River World Payments.

Payment method: Cards

Cards, debit or credit, will be processed in real time. The payment page offers flexibility to do a dual-message authorization/capture or a single-message Debit, configurations for 3D Secure as well as tokenization to store the card details for later usage.



Card Transaction type options - Authorize or Debit

The suitable implementation for a merchant depends primarily on the kind of services or goods that are associated with the transactions processed via Payment Page.

- Debit
  The *Debit* transaction type triggers a withdrawal from the consumer account. This is generally used for services or digital goods (e.g. downloads) that can be transferred to the consumer immediately.
- Authorize and capture
  The *Authorize* transaction type reserves funds on the consumer account but does not withdraw actual funds. This is generally used when physical goods are being exchanged. **If**

you use Authorizations you must upon successful processing of an authorization submit a separate *Capture* transaction after the goods are delivered, either in a batch file or by using the Digital River World Payments' Web Service Interface, to settle the authorized funds into your merchant account.

3D Secure options

3D Secure is an option to enhance security and prevent fraud. This is done by checking 3D Secure enrolment status of card holders and performing 3D Secure authentication of enrolled cards. Merchants must decide on whether they want this option activated, or in some cases this is also required by the acquiring bank.

If 3D Secure is used, the merchant must decide on a configuration profile that specifies how restrictive the liability shift rules are applied. Currently there are two profiles available:

- **Profile "Accept all cards"**
  This means that Digital River World Payments will process all cards. If the card is enrolled in 3D Secure, 3D Secure verification will be performed. If not, the card will be processed in any case.

  Basic flow:
  DRWP checks if 3DS authentication is required for the card. If it is, DRWP redirects the consumer to the ACS (Access Control Server). Else, the payment continues directly with card processing.

  In case the consumer is redirected to the ACS, DRWP awaits the response and depending

  On the response it performs the following actions:

  1) If authentication is successful it will continue with 3DS processing.
  2) If authentication is not required it will continue without3DS.
  3) If the authentication was not successful, the merchant informs the CardHolder that the purchase was declined.

- **Profile "Only accept cards that are enrolled in 3D Secure."**
  This means that when a consumer enters a card number, Digital River World Payments will check if the card is enrolled in 3D Secure. If it is, the consumer will be redirected to the issuing bank and the transaction will proceed as normal. If the consumer is not enrolled in 3D Secure, Digital River World Payments will decline the transaction.

  Basic flow:

  DRWP checks if 3DS authentication is required for the card. If it is, DRWP redirects the consumer to the ACS (Access Control Server). Else, the transaction is declined.

  In case the consumer is redirected to the ACS, DRWP awaits the response and depending

  On the response it performs the following actions:

  1. If authentication is successful it will continue with 3DS processing.
  2. If the authentication was not successful, the merchant informs the CardHolder that the purchase was declined.

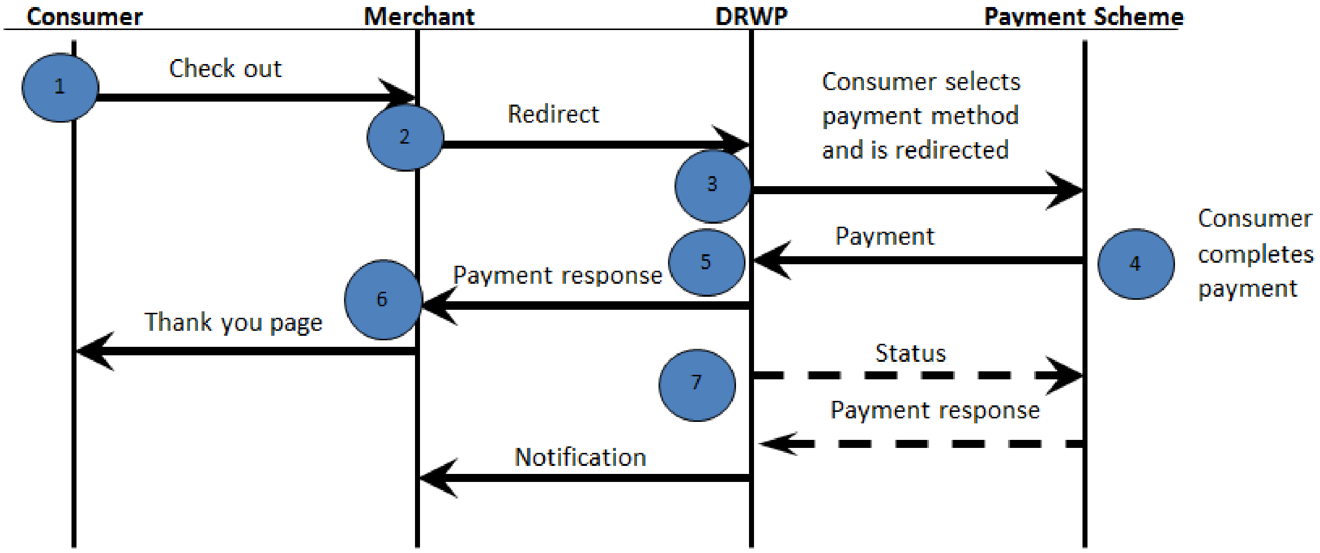
Please note that the configuration for these profiles is done by DRWP on behalf of the merchant.

Tokenization and Store flag

If your business requires that you reuse the same card number multiple times, you may use the Tokenization feature in Payment Page. Payment page provides a flag called `Store` that is used to save credit card information at Digital River World Payments for later use. This is managed by setting a store flag in the redirect URL before sending the consumer to the Payment Page. For SEPA Direct Debit tokenization is by default enabled as it stores Mandate Information. Storing payment information can either be done together with an actual transaction, e.g. a card debit transaction, or by itself which only returns a reference number that can be used for later transactions. This option is for instance often used for recurring or subscription-like business models.

Payment method: IBP and other redirects to 3rd parties

IBP, Internet Bank Payments, is a payment method designed specifically for e-commerce where consumers complete payments in real time on their banks' internet sites. Consumers are redirected from the merchant site to their banks' internet sites where they log on, complete payment and are redirected back to the merchant with confirmation. A similar flow is used for other so called redirect payment methods such as e-wallets and others.



Automatic Query time configuration

If the consumer does not return to Payment Page after they have been redirected to the bank to complete the payment, there is a procedure for how to find the status of the transaction.

If the consumer has not returned in X minutes, Digital River World Payments will request the bank for the status of the transaction. If the payment is accepted or declined a notification will be sent to the merchant with the status.

Now the auto query is not configurable for merchants, rather it is mandatory for all merchants with same configurations.

Automatic Query will only be sent for Processed InitDebits without a Query/Debit in an end state, i.e. processed or declined, (called Orphan Inits) and will stop as soon as an end state has been reached.

In new automatic query we will have three separate jobs processing orphan transactions with different time span (time range in which auto query will pick up orphan transactions) and retry interval.

1) **Most Frequent Job:** Time span of this job is 2hrs. And this job will execute in every 10 min. It will take orphan transactions in the span of 2hrs from the current time.

2) **Average Frequent Job:** Time span of this job is 24 hrs. And this job will execute in every 120 min. It will take orphan transactions in the span of 24hrs from the current time. If any orphan transaction is not processed in time span of **Most Frequent Job** then this job will try to process it.

3) **Least Frequent Job:** Time span of this job is 1 week and retry interval for this job is 24 hrs. it will take orphan transactions in the span of 1 week from the current time. Now as this is the last job to execute, we have some transactions which will get queried for the last time i.e. they will not get processed any more. This is called as final auto query.

To achieve final auto query in third job we will divide this job into two parts-

i) **Part 1 –** This will act as a Final Auto Query for orphan transactions because after this time span no other job will process those orphan transactions which will fall into this time span. In this case, if the status of query transaction is not Processed/Declined, then "PENDING" notification is sent. Also we send an alarm in case of Final Auto Query with all orphan transactions list.

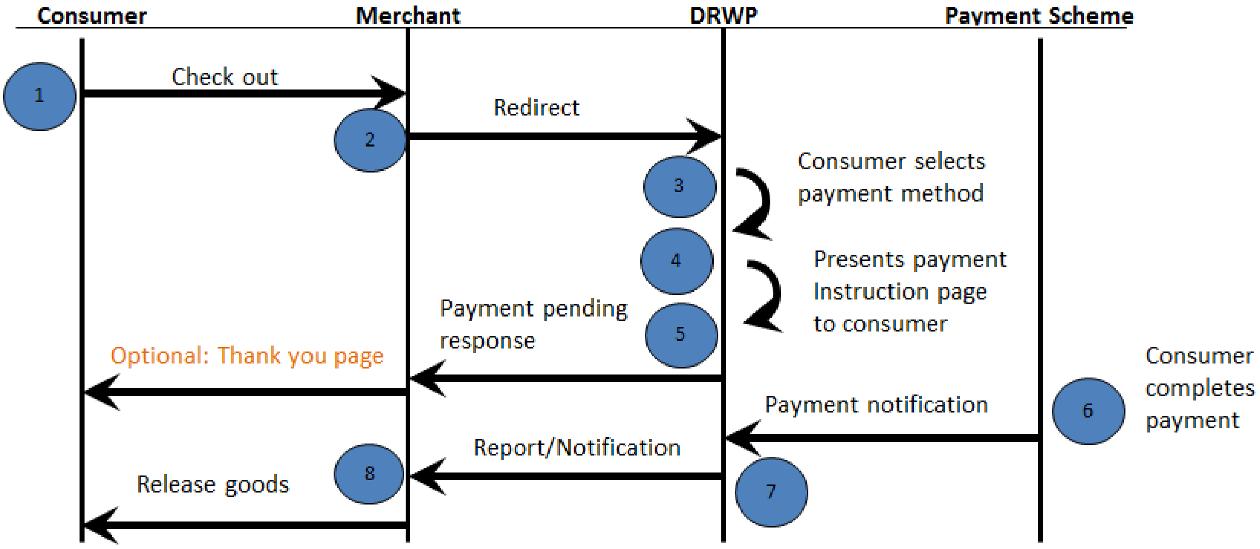
ii) **Part 2 -** Rest of the orphan transactions will be queried under this time span.

Note: The values for time span and retry interval are configurable, which will be common for all merchants.

Payment method: EFT

Offline payment methods like Electronic Fund Transfers (a.k.a. “Wire transfers”) are the family of payment methods where the consumer does not complete their payment during the order session. Instead, the consumer is instructed online to complete the payment via another mean, e.g. by going to a local convenience store and paying it there using a payment reference. Payment page will return a reference, a PENDING status and in some cases a link to a payment slip to the client system.

Digital River World Payments will then check with the bank for status updates and notify the merchant in a report and/or notification once the money has been deposited. Note that it is possible that the consumer never completes the payment.



EFT: Hosted result page

Digital River World Payments can also host the result page (5) in which case the customer is not directed back to merchant’s site once order has been placed. Payment Page will instead redirect the customer to a locally hosted result page. This page is built and selected the same way as the payment page. The result template is selected the same way as the payment template.

There is of course a link back to the merchant’s site at the bottom of the result page, containing the same data as if the result where to be presented on the merchant’s site.

The result page, EFT specific, contains information on how the customer completes the payment. I.e. it presents account information for the account to which the consumer should deposit his/her payment.

EFT: Present result page again

In case the consumer needs to see the payment instruction page again the merchant has the option to redirect the consumer to Payment Page once again using the same order ID and setting the correct payment method ID. Payment Page will then present the original payment instruction page without initiating a new payment.

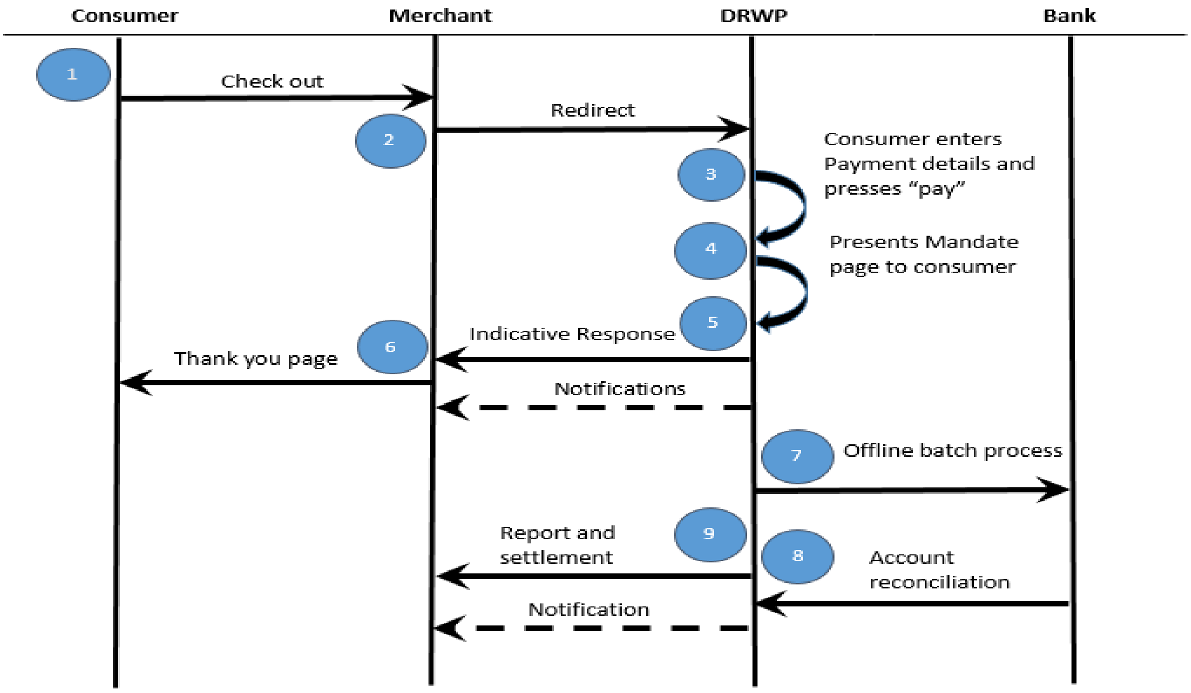
Payment method: SEPA Direct Debit

SEPA Direct Debit is an "offline" payment method offered by DRWP.

The money will be withdrawn from the consumer's account at a later date and Payment Page will respond PENDING since the funds haven't actually changed hands yet.

When funds have been transferred then Digital River World Payments will notify the merchant by a report and/or a notification.

More information on SEPA Direct Debit can be found in "Direct Debit Integration Guide".



Direct Debit: Hosted Payment Page

On the main payment page the consumer enters their bank account details and has option to change billing information. On the same page consumer must agree to SEPA Mandate Terms and conditions to submit the details.

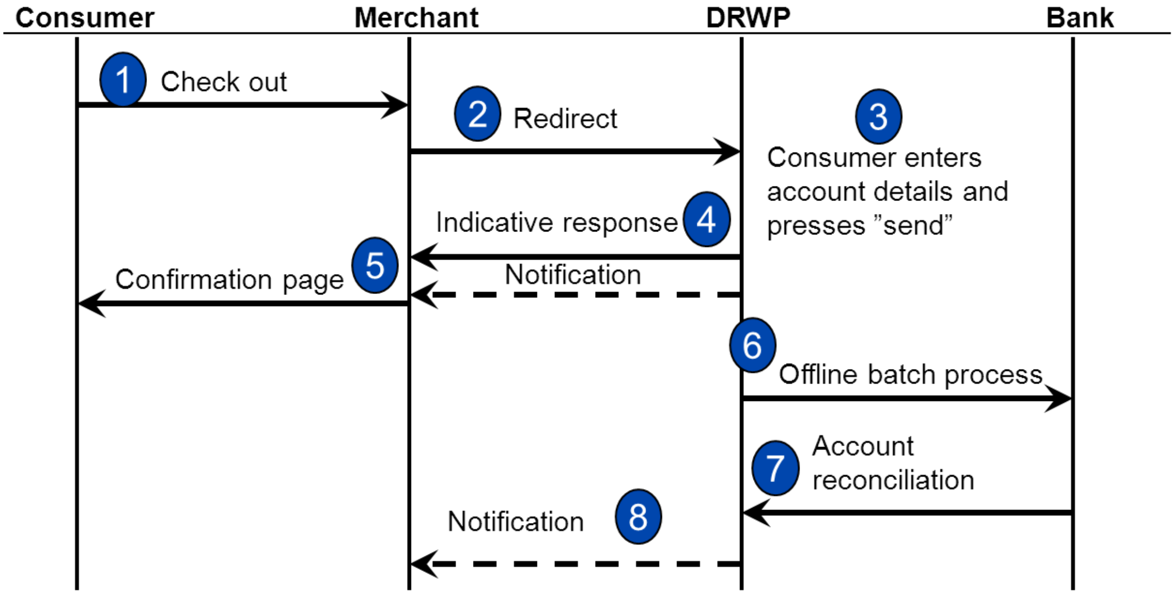
Once consumer agrees to SEPA Mandate Terms & Conditions he/she will be redirected to intermediate Mandate Page where consumer have option to print the Mandate and a link to go back to merchant. Once the Mandate is stored a notification will be sent to merchant which will include Token (Mandate Id) and merchant can refer to that Token for future payments.

Payment method: Payout

The Payout payment method is primarily intended to be used in combination with other (non-card) payment methods to refund consumers. Payout is an account-to-account payment method that uses domestic local clearing systems in order to perform payments back to consumers. All that is needed in order to make a Payout to a consumer is that the consumer has a local bank account in one of the countries where this service is supported.

The merchant redirects the consumer to Payment Page setting the transaction type to REFUND. The order ID should be the same as for a previous successful payment. Just as for SEPA Direct Debit the consumer enters the account information on the main payment page and the money will be funded to the consumer's account later on. Payment Page will respond PENDING since the funds haven't actually changed hands yet.

When funds have been transferred then Digital River World Payments will notify the merchant in a report and or/notification.


Consumer
Merchant
DRWP
Bank
1 Check out
2 Redirect
3 Consumer enters account details and presses ”send”
Indicative response 4
Notification
Confirmation page 5
6 Offline batch process
7 Account reconciliation
Notification 8

Step 2 - Client side Implementation

Select your preferred way to create URL and unpack response

When the consumer is redirected from the merchant to the Payment Page, the merchant has to pass on a few data elements, such as amount, currency etc., to Payment Page. Since some of the data elements contain sensitive information, Payment Page implements a secure protocol to protect the data. To assist the merchant in packing the data in a correct manner, a Java reference implementation is supplied in the start kit. A merchant may use the Java implementation as is or implement the needed functionality by using the reference implementation for guidance. The same thing can be done using .Net reference implementation if using .Net.

Install certificates for authentication

The security solution in Payment page consists of host authentication and message encryption. Certificates are used to authenticate the other part when the merchant communicates with Digital River World Payments. The public certificate is signed by *Digital River World Payments* and is used continuously until replaced by Digital River World Payments.

During the implementation of Payment Page, you need to install the key store (and trust store if using .Net) in the environment your application is calling from **twice**:

1. The first set is used during the development of the Payment Page solution at the merchant site. For your benefit, the key store and trust store used during development and test in the Acceptance Test Environment are provided as part of the Start-up kit.
2. The second set is implemented at the time of launching the Payment Page solution in production for live traffic. Production certificates will be issued by Digital River World Payments as part of moving the solution to production, and **you must request the certificate to be signed by Digital River World Payments** (see the section about going live for more information about this process).

For more detailed information about how the message exchange is secured, please see the Payment Page Security Features Specification (included in the Start-up kit).

Certificates included in the Start-up kit

The file `test_keys.zip` in the Start-up kit includes a number of certificate- and key files. They will be explained in detail in this chapter.

For a merchant using the Java API, the only file really needed is the Java key store (see the chapter **Java reference implementation** for implementation details):

- `java/merchant.jks`

For a merchant using the .Net API, two files are needed (see the chapter **.NET reference implementation** for implementation details):

- The PKCS12 container (combined private key and merchant certificate file):
    - `pkcs12/merchant.pfx`
- The trust store (DRWP public certificate). It's available in three different formats:
    - `pkcs12/trust.pfx`
    - `pem/drwp_cert.pem`
    - `der/drwp_cert.cer`

If a merchant doesn't use neither the Java nor the .Net API, the provided files can be used to build key- and trust stores as needed by the implementation based on the following files:

- Merchant private key. Available in the following formats:
    - `pem/merchant_key.pem`
    - `der/merchant_key.der`
- Merchant public certificate. Available in the following formats:
    - `pem/merchant_cert.pem`
    - `der/merchant_cert.cer`
- DRWP public certificate, to be added to the trust store. Available in the following formats:
    - `pem/drwp_cert.pem`
    - `der/drwp_cert.cer`
    - `pkcs12/trust.pfx`

Please see the file `info.txt` included in the `test_keys.zip` for information about the passwords of the different key stores.

Java reference implementation

If the merchant is using the Digital River World Payments Java reference implementation, the *PaymentPageAPI-<version>.jar* file needs to be placed on the classpath along with the *merchant.jks* keystore provided by Digital River World Payments (the keystore can also be pointed out with an absolute path, see below).

Example:

- Initialize the API with a 2048 bit keypair as:

```
PaymentPageHandler paymentPageHandler = new PaymentPageHandler(
    PaymentPageHandler.DEFAULT_PRODUCTION_BASE_URL,
    new JKSKeyHandlerV6("merchant.jks", "merchant”, “merchant",
    "drwp_cert"));
```

Where the first argument is the name of the keystore file. The second argument is the key store password. The third argument is the alias (name) of the merchant private/public key pair. The fourth argument is the alias of the DRWP certificate (public key).

.NET reference implementation

When using the .NET API, the NetgiroClient.dll needs to be included in your project. Either x86 or x64 version depending on your operating system. Initialize the API with the `PPDOTNETPaymentPageHandler.init()` method.

Example:

- Initialize the API with a 2048 bit keypair as:

```
private PPDOTNETPaymentPageHandler ppHandler = new PPDOTNETPaymentPageHandler();

X509Certificate2 merchantCert;
X509Certificate2 drwpCert;

merchantCert = new X509Certificate2(PPDOTNETSecurityHandlerV6.getCert
              ("test_keys\pkcs12\merchant.pfx", "password", True));

drwpCert = new X509Certificate2(PPDOTNETSecurityHandlerV6.getCert
           ("test_keys\der\drwp_cert.cer", null, False));

ppHandler.init(merchantCert,
               drwpCert,
               “https://testpage.payments.digitalriver.com/pay/?creq=”,
               new PPDOTNETSecurityHandlerV6());
```

Redirecting the consumer to Payment Page

The consumer shall now be redirected to Payment Page to complete their payment

1. Call the `createRedirectURL` method `for Java reference/ createPaymentPageUrl` method `for .Net reference` with the desired order parameters from your checkout application. Several parameters are (R) required when initiating this call, while others are (C) conditional or (O) optional depending on merchant configuration. For example, you will need to provide

    1. Your Merchant ID
    2. Unique order parameters, such as Order ID, Amount, Currency
    3. Which payment page to present via the template reference parameter. The page is created in Page Builder (described separately). Each page will be allocated a reference ID by the system. It is recommended to take this into account when building the integration on the client side since you may want to be flexible for easily accessing newly added pages when entering new markets, adding brands or other situations.
    4. Which locale (country + language) to use
    5. Which return URL to use for the payment response
    6. Whether to store the card details at Digital River World Payments for future usage (tokenization)
    7. etc. (see appendix for a complete list of parameters)

    Keep in mind that some web browsers have a maximum limit of allowed characters in the URL and we recommend that the created URL doesn´t exceed 1500 characters. To prevent this you should try to not pass unnecessary information e.g. in the additional parameters field.

2. The secure URL created by the API will look something like this:

    `https://secure.payments.digitalriver.com/pay/?creq=AbCd123456`...

3. Redirect the consumer's browser to the secure URL. This must be performed with a HTTP GET Redirect, and sends the consumer to the Payment Page.

Handle the response from Digital River World Payment

Once the consumer has completed their interaction with Payment Page, the result is returned.

1. The result of the consumer's interaction with Payment Page is sent via HTTP GET to the "return URL" that was provided in the request. It will look something like this:

   https://["return url"]/?response=BEii9iVIovcIfS_zH...

   The payment result is encrypted in the `response` parameter.

2. Use the `UnpackResponse` method to decode and verify the contents of `response`. See Appendix for a complete list of parameters found in the response.

3. Take action - there are six statuses that can appear in the response, described in the table below, indicating what kind of message to present to the consumer or what action to take. Note that Digital River World Payments does not present the result of a payment to the consumer.

Status	Description
OK	The payment has been successfully processed by Payment Page!
NOK	The payment was not completed. The user has used several attempts to submit a payment and ultimately been redirected back to the merchant due to exceeding the (configurable) maximum number of retries. NOK may also be returned if Digital River World Payments detects a possible fraud attempt. Please contact your integration manager about Digital River World Payments fraud protection offerings.
USERCANCEL	The consumer pressed the cancel button. You may redirect the consumer back to the payment page.
TIMEOUT	The consumer tried to initiate a payment after the timeout set by the merchant has expired.
PENDING	The final status of the payment is not yet determined. The final status of the payment will be notified via a notification, in a report or pulled from Digital River World Payment Web Interface depending on your setup. Note that this is an expected result for many payment methods.
ERROR	Payment Page detects that there is something wrong in the URL content or the configuration of the merchant. If this result status is encountered, the merchant should review their parameters and contact Digital River World Payments if the problem cannot be identified.

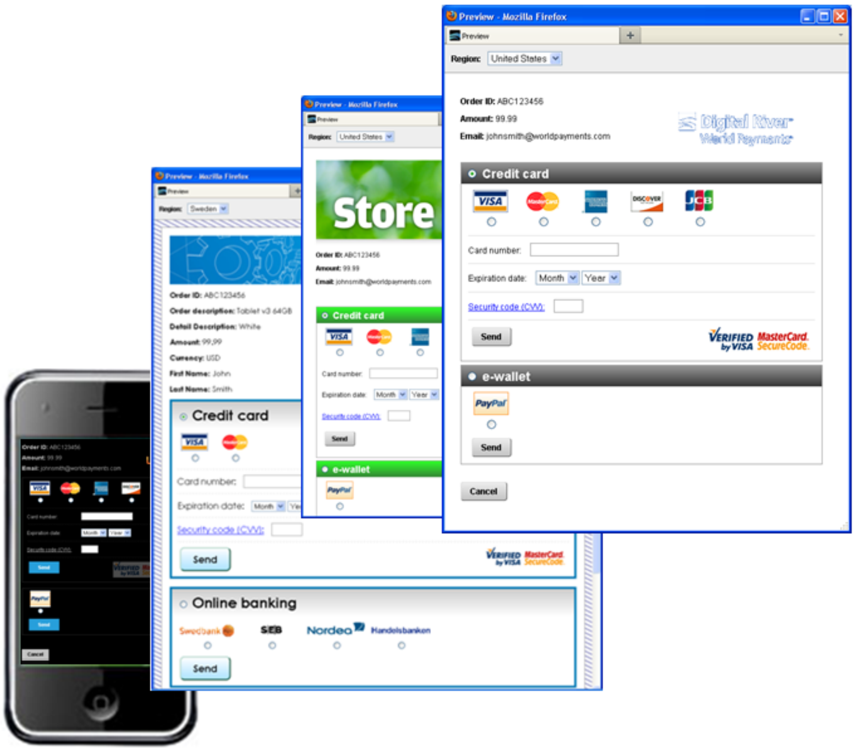
Multiple responses and no response

One very important aspect to remember when implementing response handling is the fact that the consumer is in control of the information flow. The merchant should expect that there will occasionally be multiple responses for a single transaction, due to reloads and navigating via the history. The best way to handle this is to make sure that if the merchant receives a response with status *OK*, this must always take precedence over any previous or later responses with another status.

In certain cases it is also possible that no response will sent at all. For example, if the consumer is redirected to the bank for 3D Secure verification and never comes back, Payment Page has no way of returning the consumer to the merchant.

Step 3 – Customize the Payment Pages

Via the Page Builder content management wizard in DRCC you can create a set of localized payment pages.



Preview - Mozilla Firefox

Region: United States

Order ID: ABC123456
Amount: 99.99
Email: johnsmith@worldpayments.com

○ Credit card

Card number:
Expiration date: Month Year
Security code (CVV):

Send

● e-wallet

Send

Cancel

Figure 1. Examples of customized pages created in Page Builder

The generated payment page is presented in run-time as a combination of:

- The web page template created via Page Builder
- Text in local language based on the locale parameters from the redirect URL
- Content and data from the redirect URL

Creating the pages

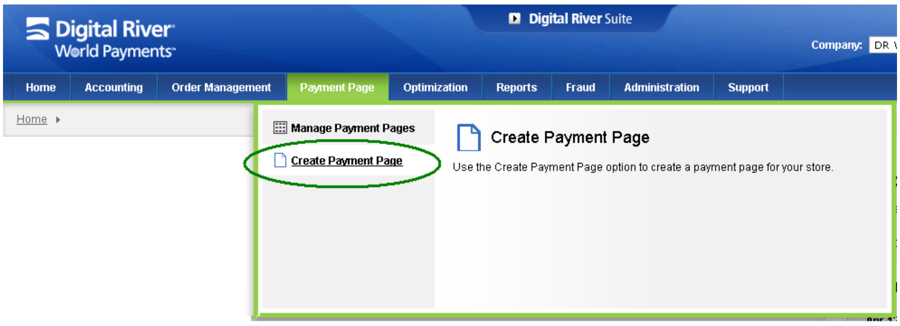
To be able to create and manage the payment pages you will need access to the Digital River Command Console (DRCC) and a DRCC-user account with the permission to access Payment Page (Page Builder) pages. There are two environments that you will use. During testing, development and certification you will use the Acceptance environment and when you go-live you will use the Production environment. The two environments will require two different users. Request your DRCC user account credentials from your integration manager.

Use the following URLs to access DRCC:

Environment	URL
Acceptance (test) environment	
Production environment	https://www.payments.digitalriver.com

Go through the following steps to get started:

1. Log in to the DRCC
2. Go to [Payment Page] menu and click on [Create payment page]



Use the Create Payment Page option to create a payment page for your store.

3. The Page Builder wizard will help you create pages for the Merchant ID(s) that you are allowed to manage. Help pages are available in the tool if needed.
4. You must actively deploy the page you want to use. If a page is not deployed, it is only available in the Page Builder tool and not on the Payment Page servers. A deployed page is usually activated within a minute or a few minutes, but it can sometimes take up to 10-15 minutes until the page is fully activated on all our servers.
5. Each page will be allocated with a reference ID. This is the ID you will need to pass in the template reference parameter in I in order to present the page. If you create a new page, a new reference ID will be generated by the system for this page.

Available markets from Digital River World Payment

The following markets are provided as default, but may change over time so please check with your account manager

Market	Payment methods available	Description	Recommended language
International	Visa, MasterCard, American Express, PayPal, UnionPay	Standard used for internationally accepted payment methods	Any*
USA	"International" + Discover, JCB, PayPal		US English*
United Kingdom	"International" + Maestro UK, Solo, Visa Delta, Visa Electron		UK English*
Ireland	"International" + Laser		UK English*
France	"International" + Carte Bleue, Carte Aurore		French*
Germany	"International" + Maestro International, Giropay, Sofortüberweisung, SEPA Direct Debit (ELV), EFT		German*
Netherlands	"International" + Maestro International, iDEAL, SEPA Direct Debit, EFT		Dutch*
Austria	"International" + Maestro International, EPS, SEPA Direct Debit		Austrian German*
Sweden	"International" + IBP Swedbank, IBP SEB, IBP Nordea, IBP SHB/ Handelsbanken, Maestro International, Visa Electron		Swedish*
Finland	"International" + IBP Säästöpankki, IBP S-Pankki, IBP OP-Pohjola, IBP Aktia, IBP Danske Bank, IBP Nordea, Maestro International, Visa Electron		Finnish*, Swedish

Market	Payment methods available	Description	Recommended language
Denmark	"International" + IBP Nordea, Dankort, Dankort Visa		Danish*
Norway	"International"	Use International template and local language	Norwegian*
Spain	"International"	Use International template and local language	Spanish*
Italy	"International"	Use International template and local language	Italian*
China	"International" + Alipay, IPS (IBP and debit cards), UnionPay		Simplified Chinese*
South Korea	"International" + Local payments (requires Internet Explorer)		Korean*
Brazil	"International" +¨ Boleto, Hiper, Diners, Aura, Elo		Brazilian Portuguese
Australia	"International" + BPay, UnionPay		English
Lithuania	"International"+ EFT		Lithuanian
Poland	"International"+ Dotpay		Polish

Notes:

- These pages are tested with JavaScript and cookies activated.

Languages

We have listed the recommended language for each market, although any language may be used for any market, and this is specified in each redirect request using standard ISO codes. Digital River World Payments provides localized language properties for a number of markets.

Note: The pages require local language settings in the Operating System of the client computer to display correctly (e.g. Simplified Chinese may not look right on Western locale settings).

Timeout

A timeout can be set in the redirect to Payment Page and is configured in seconds. If a consumer attempts to initiate a payment after the timeout is reached it will be rejected and return status *TIMEOUT*. If this parameter is not set, or set to zero, it means that the request will never time out.

Tokenization

If tokenization is used, this is specified as part of the call before sending the consumer to Payment Page. For SEPA Direct Debit tokenization is by default enabled as it stores Mandate Information.

Retries

The number of retries allowed for each consumer is configured on the server side and cannot be changed in the templates. Default is three times.

Error handling on the page

Within the page an error text may be displayed to indicate that the consumer has performed a failed payment attempt before the current page hit. The various errors that may occur have been grouped by the appropriate action for the consumer, based on the list below, and are localized:

- Incorrect payment information. The consumer should check details and try again.
- Temporary error. The consumer should try again in a short while.
- Payment declined. The consumer should check details or try a different bank/card

Security

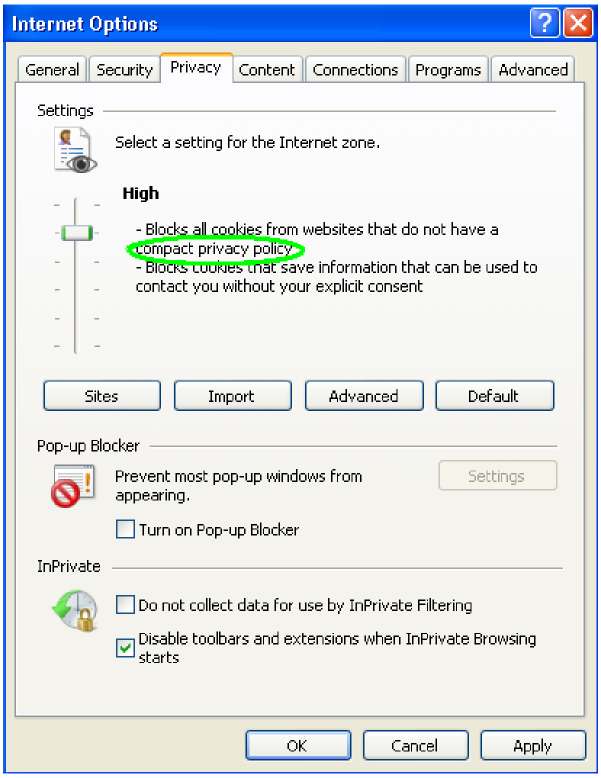
Since Digital River World Payments are hosting the servers where Payment Page templates are processed and displayed, the content must meet both Digital River World Payments and PCI-DSS (Payment Card Industry Data Security Standard) requirements. **The merchant is responsible** for any submitted files or scripts or material that is made available through Payment Page and must be cautious not to introduce vulnerabilities or exploits, such as cross site scripting (XSS).

Digital River World Payments reserves the right to remove the content from its servers.

For more information on web security, Digital River World Payments recommends visiting http://www.owasp.org/.

Cookies

Payment Page uses cookies, and has implemented the industry standard compact privacy policy. This policy will be accepted for all Privacy settings in Internet Explorer except the highest one, "Block All Cookies". In other words, the following setting in Internet Explorer, or less strict, will be compatible with the Payment page cookie policy:



Select a setting for the Internet zone.

High

- Blocks all cookies from websites that do not have a compact privacy policy
- Blocks cookies that save information that can be used to contact you without your explicit consent

Prevent most pop-up windows from appearing.

□ Turn on Pop-up Blocker

□ Do not collect data for use by InPrivate Filtering

☑ Disable toolbars and extensions when InPrivate Browsing starts

Please be aware that globally, many banks and 3rd party payment methods (e.g. e-wallets and Internet banks) use cookies in their interaction with the consumer. Therefore, a user will not be able to pay unless they accept cookies as part of the checkout process.

Step 4 – Testing and Certification

Acceptance Test Environment

To simplify and speed up the integration process for the merchant, Digital River World Payments provides an integration environment; the Acceptance Test Environment. The content can be tested by the merchant without the need for lengthy interaction with Digital River World Payments.

See *Appendix F – URLs* for details.

Certification

When the merchant has performed testing in the acceptance test environment the merchant is ready to begin the certification process. The merchant will execute transactions according to certification protocols provided by Digital River World Payments and the result will be verified. Exactly what needs to be certified depends on the payment types and methods in use. A complete specification of all available certification test cases can be found in the start kit. Note that the merchant will receive a customized version of the TestData.xls document which will specify all the test cases that the merchant needs to perform to become certified.

Step 5 – Moving to Production

Now that you have built and certified your solution, you need to request production certificates to ensure that you follow the security standards employed within Payment Page.

Request and install certificates for production

Note that the merchant needs to work with separate certificates for the production environment. A high level procedure for generating new certificates is described here:

1. The merchant generates an RSA 2048 key and sends a Certification Signing Request to Digital River World Payments.
2. Digital River World Payments responds by issuing 2 certificates
   - The requested merchant certificate, signed by Digital River World Payment, used by Payment Page to encrypt the responses to the merchant.
   - Digital River World Payments' public certificate, used by the merchant to encrypt the requests to Payment Page.
3. The merchant switches certificates.

How-to guides for the different APIs are described in the following sections

Java Example

1. Generate a new RSA keypair for production use.
   - For generating 2048 size RSA keypair:

```
>keytool -genkeypair -keystore merchant.jks –sigalg SHA256withRSA -
alias merchant -keyalg RSA -keysize 2048 -dname
"C=SE,O=Organisation,OU=OrganisationUnit,CN=CertificateHolder"
```

A few notes:

- This command will create a new `merchant.jks` keystore, so make sure that there is no file named merchant.jks in the folder where the command is executed.
- Before running the command, replace the distinguished name "C=SE,O=Organisation,OU=OrganisationUnit,CN=CertificateHolder" with information that applies to your company. For instance, "C=US,O=Digital River World Payments Systems AB,OU=Digital River World Payments ITO,CN=payments.digitalriver.com".

- The password that keytool prompts for will be the password used for accessing the keystore. It is important that the same password is used for the key or the Digital River World Payments API will not be able to access it.

2. Create the certificate request that will be used by Digital River World Payments to issue a production certificate.

```
>keytool -certreq -alias merchant -file request.csr -keystore
merchant.jks
```

3. Send the file *request.csr* to your integration manager at Digital River World Payments.

4. Digital River World Payments will respond by sending back:

    - The merchant certificate, signed by Digital River World Payments. Two different formats are available. Choose any of them:

        i. *merchant_cert.pem*

        ii. *merchant_cert.p7c*

    - The Digital River World Payments public certificate. Two different formats are available. Choose any of them:

        i. *drwp_cert.cer*

        ii. *drwp_cert.pem*

5. Import the issued merchant certificate in the production keystore. This will associate the certificate with the key.

```
>keytool -importcert -trustcacerts -alias merchant -file
merchant_cert.p7c -keystore merchant.jks
```

6. Import the Digital River World Payments certificate. This will associate the certificate with the key.

```
>keytool -importcert -alias ngcert -file drwp_cert.pem -keystore
merchant.jks
```

7. Replace the acceptance test keystore with the new keystore in production. It must be named *merchant.jks*.

The private key generated by the keytool command in step 1 can be exported to PEM format using the tool in ***Appendix B – Export Private Key to PEM format****.*

.NET API Example

1. Generate a new RSA keypair for production use. In this example openssl is used for key generation, but other tools can be used.

   - For generating 2048 size RSA keypair:

   ```
   >openssl req -new -newkey rsa:2048 -nodes -out request.csr -keyout
   merchant_key.pem -keyform PEM -subj
   "/C=SE/O=Organisation/OU=OrganisationUnit/CN=CertificateHolder"
   ```

   Note:

   - Before running the command, replace the distinguished name "C=SE,O=Organisation,OU=OrganisationUnit,CN=CertificateHolder" with information that applies to your company. Since this certificate will also be used for receiving notifications the CN will usually be the hostname of the receiving server. For instance, "C=US,O=Digital River World Payments Systems AB,OU=Digital River World Payments ITO,CN=payments.digitalriver.com".

2. Send the file *request.csr* to your integration manager at Digital River World Payments.

3. Digital River World Payments will respond by sending back:

   i. The merchant certificate, signed by Digital River World Payments. Two different formats are available. Choose any of them:

      1. *merchant_cert.pem*
      2. *merchant_cert.p7c*

   ii. The Digital River World Payments public certificate. Two different formats are available. Choose any of them:

      1. *drwp_cert.cer*
      2. *drwp_cert.pem*

4. Create a pfx file by combining the key generated in step 1 with the merchant certificate received from Digital River World Payments. In this example openssl is used, but other tools can be used.

   ```
   >openssl pkcs12 -export -out merchant.pfx -inkey merchant_key.pem -
   in merchant_cert.pem
   ```

5. Replace the *merchant.pfx* and *drwp_cert.cer* (alt. *drwp_cert.pem*) from acceptance test with the new files for production.

Moving to production

- Digital River World Payments will set up the merchant in the production environment and distribute required information for accessing Payment Page. For details about URLs in production see Appendix F – URLs
- You will also be issued a Production Merchant ID, so you will need to change this parameter in your code.
- Lastly, Digital River World Payment needs to move your payment page files to the production site. Once there, you have control over them again.
- Once the certificates are installed and all address and ID modifications are done on the client side, contact Digital River World Payments to set up a first live test transaction to verify the setup.
- Then, go live!

Appendices

Appendix A – Parameters

Request Parameters

The following parameters may be used when initiating in a Payment Page request. The name column is only of interest for merchants that not wish to use the Java reference implementation.

Parameter	**Required?**	**Name**	Comment	Valid values	Max length
MID	**Yes**	A	Merchant ID		
Sub-merchant ID	No	B	Sub-merchant ID		
POS ID	Check with integration manager	C	Point of sale ID		
Transaction channel	**Yes**	D	Transaction channel. Typically “Web Online”.	Web Online Mail Telephone Fax FaceToFace Cash register	
Transaction type	No	E	Transaction type	debit authorize refund	
Token	No	F	Previously stored token		
Order ID	**Yes**	G	Order ID		50
Order description	No	H	Order description		
Order detail description	No	I	Order detailed description		
Amount	**Yes**	J	Order amount.		
Currency	**Yes**	K	Currency. Three alphabetic letter, ISO-4217 code		3
VAT amount	No	L	VAT (Value Added Tax) amount.		

Parameter	**Required?**	**Name**	Comment	Valid values	Max length
VAT rate	No	M	VAT rate		
Country	**Yes**	T	Country code. ISO-3166, e.g. US. Used together with language to form the locale being used by Payment Page		2
Language	**Yes**	U	Language. ISO 639-1, e.g. en. Used together with country to form the locale being used by Payment Page		2
Return URL	**Yes**	V	URL for consumer redirect back to merchant.		
Time limit	No	W	Maximum time in seconds for request to be valid or zero if not used.		
Additional parameters	No	Y	Additional parameters. Used for ${param.*} in template. Name-value pair, separated by '#'. E.g. "PARAM1=VALUE1#PARAM2=VALUE2#"		
Payment method ID	No	Z	Used when the consumer selects the payment method at the merchant site.		
Store flag	No	AA	Indicates that a token should be stored.	"0" = Store not used "1" = Store and Debit/Authorize "2" = Store only	1
Template reference	No	AB	Template reference. Specifies which page to present.		

Parameter	**Required?**	**Name**	Comment	Valid values	Max length
Billing address line 1	No	AG	Billing address line 1		
Billing address line 2	No	AH	Billing address line 2		
Billing city	No	AI	Billing city		
Billing state province	No	AJ	Billing state province		
Billing zip code	No	AK	Billing zip code		
Billing country	No	AL	Billing country code. ISO-3166, e.g. US.		2
Billing e-mail address	No	AM	Billing e-mail address		
Billing phone	No	AN	Billing phone number		
Billing mobile phone	No	AO	Billing mobile phone number		
Billing last name	Check with integration manager	AP	Billing last name. Will be concatenated with Billing first name to form a Billing full name.		
Billing first name	Check with integration manager	AQ	Billing first name. Will be concatenated with Billing last name to form a Billing full name.		
Billing full name	Check with integration manager	AR	Billing full name. Instead of using first+last name the full name can be submitted directly.		
Shipping address line 1	No	AS	Shipping address line 1		
Shipping address line 2	No	AT	Shipping address line 2		
Shipping city	No	AU	Shipping city		

Parameter	Required?	Name	Comment	Valid values	Max length
Shipping state province	No	AV	Shipping state province		
Shipping zip code	No	AW	Shipping zip code		
Shipping country	No	AX	Shipping country code. ISO-3166, e.g. US.		2
Shipping e-mail address	No	AY	Shipping e-mail		
Shipping phone	No	AZ	Shipping phone number		
Due date	No	BA	Due date for payment. If not set, then max configured is used instead.		
Payment plan code	No	BB	Payment plan code is used to break up a payment into multiple payments (paid over time, usually monthly). The payment plan code describes the length and type of installment that should be used.		
Billing company name	No	BC	Billing company name		
Billing buyer VAT number	No	BD	Billing buyer VAT number		25
Billing buyer type	No	BE	Billing buyer type	individual business	
Shipping company name	No	BG	Shipping company name		
Shipping address line 3	No	BH	Shipping address line 3		
Billing address line 3	No	BJ	Billing address line 3		

Parameter	**Required?**	**Name**	Comment	Valid values	Max length
Birth date	No	BT	Birth date		
Company responsible birth date	No	CB	Birth date of the responsible person at the company		
Company responsible full name	No	CN	Full name of the person responsible at the company		50
Company responsible VAT number	No	CV	VAT (Value Added Tax) number for the responsible person at the company		25
Recurring Type	No	EA	Recurring type. Mandatory for SEPA Direct Debit.	NOT_RECURRING FIRST_RECURRING SUBSEQUENT_RECURRING LAST_RECURRING	
POS description	No	PD	Point of sale description		
Shipping mobile phone	No	AAA	Shipping mobile phone number		
Shipping last name	No	AAB	Shipping last name. Will be concatenated with Shipping first name to form a Shipping full name.		
Shipping first name	No	AAC	Shipping first name. Will be concatenated with Shipping last name to form a Shipping full name.		
Shipping full name	No	AAD	Shipping full name. Instead of using first+last name the full name can be submitted		

Parameter	**Required?**	**Name**	Comment	Valid values	Max length
			directly.		
Billing SSN	No	AAE	Billing social security number		30
Company tax ID	No	AAF	Company tax ID.		50
Gender	No	AAG	Gender, used for Klarna in certain countries.		
Billing Street Name	No	AAH	Street name, used together with House Number instead of “Address Line 1” for Klarna in certain countries.		
Billing House Number	No	AAI	House Number, used together with Street name instead of “Address Line 1” for Klarna in certain countries.		
Billing House Extension	No	AAJ	House extensiona (for example “B”) used for Klarna in certain countries.		
Shipping Street Name	No	AAK	See Billing Street Name		
Shipping House Number	No	AAL	See Billing House Number		
Shipping House Extension	No	AAM	See Billing House Number		
Shipping CareOf	No	AAN	C/O address (if used by the consumer).		
Authorization Type	No	AAO	Authorization type: Mastercard now require merchants to define authorization attempts as either a pre-authorization or a final-authorization. Finalauthorizations that meet	PRE_AUTHORIZATION FINAL_AUTHORIZATION UNDEFINED	

Parameter	Required?	Name	Comment	Valid values	Max length
			Mastercard’s criteria will be free of scheme fee impact but pre-authorizations & undefined authorization attempts will be subject to additional scheme fees.		
Authentication Redirect	No	AAP	It specifies whether the transaction would go through 2-step authentication flow or not	NOREDIRECT, REDIRECT, REDIRECTONLY	
Line item ID	**Yes (if line items are sent)**	LIA For each line item a suffix is needed e.g. LIA_1	Line item ID.		16
Line item description	No	LIB For each line item a suffix is needed e.g. LIB_1	Line item description		
Line item amount	**Yes (if line items are sent)**	LIC For each line item a	Line item amount		

Parameter	**Required?**	**Name**	Comment	Valid values	Max length
		suffix is needed e.g. LIC_1			
Line item quantity	**Yes (if line items are sent)**	LID For each line item a suffix is needed e.g. LID_1	Line item quantity		
Line tax amount	No	LIE For each line item a suffix is needed e.g. LIE_1	Line item tax amount		
Line tax rate	No	LIF For each line item a suffix is needed e.g. LIF_1	Line item tax rate		

For additional details, contact your integration manager.

Response Parameters

The following data is returned when the consumer is redirected back to your checkout process after having been on the Payment Page. The name column is only of interest for merchants that not wish to use the Java reference implementation.

Parameter	Name	Comment
MID	A	Merchant ID
Status	B	Status
TransactionId	C	Transaction ID
PaymentMethod	D	Payment method used
Order ID	E	The order ID set by the merchant
Timestamp	F	Timestamp
VEResId	G	VEResId. Only for 3Dsecure
PAResId	H	PAResId. Only for 3Dsecure
DDDSStatus	I	3Dsecure status. Only for 3Dsecure
POSId	J	POSID
CardTxType	K	Card Transaction type.
CardTxId	L	Card transaction ID. Only for card payment
CardType	M	Card type. Only for card payment
Token	N	A reference to a stored payment instrument.
ExpirationDate	P	Card expiration date.
StoreCardType	Q	Store card type for a store transaction.
IbpTxId	R	IBP transaction ID. Only for IBP payment

Parameter	Name	Comment
IbpTxType	S	IBP transaction type. Only for IBP payment
Redirected	T	Redirected status, true if the consumer was redirected to a third party site during the session at Payment Page
MaskedCardNumber	U	Masked card number.
StoreMaskedCardNumber	V	Masked card number for a stored token.
StoreExpirationDate	W	Expiration date for a stored token e.g. card expiration date.
EftReferenceId	X	Reference returned for an EFT payment
EftPaymentSlipUrl	Y	Payment slip URL returned for an EFT payment
EftTxId	Z	EFT transaction ID returned for an EFT payment
DirectDebitTxId	AA	Direct Debit transaction ID.
PayoutTxid	AB	Payout transaction ID
AVS answer code	AC	AVS answer code
AVS response	AD	AVS response
Acquirer answer code	AE	Acquirer answer code
Client answer code	AF	Client answer code
CVV answer code	AG	CVV answer code
CVV response	AH	CVV response
Payment method name	AI	Name of payment
Acquirer authorization code	AJ	Authorization code returned from acquirer
House Extension	AK	As entered by Consumer at the payment page (only certain countries)

Parameter	Name	Comment
House Number	AL	As entered by Consumer at the payment page (only certain countries)
Street Name	AM	As entered by Consumer at the payment page (only certain countries)
Gender	AN	As entered by Consumer at the payment page (only certain countries)
Birth Date	AO	As entered by Consumer at the payment page (only certain countries)
Answer Description	AU	Text description of the Acquirer answer code
Payment Plan Code	AV	As selected by Consumer at the payment page
Social Security Number (or similar like “Personnummer”)	AW	As entered by Consumer at the payment page
First Name	AX	As entered by Consumer at the payment page
Last Name	AY	As entered by Consumer at the payment page
City	AZ	As entered by Consumer at the payment page
Country Code	AAA	As entered by Consumer at the payment page
Zip Code	AAB	As entered by Consumer at the payment page
Address Line 1	AAC	As entered by Consumer at the payment page (only certain countries)

For additional details, contact your integration manager.

Appendix B – Export Private Key to PEM format.

The private key can be exported from a Payment Page keystore to PEM format using keytool.

This can be useful when the procedure described in ***Java Api Example*** is followed since the key will only reside in the keystore. The output is in PEM format which can be used directly or via conversion by most load balancers, proxies etc. KeyTool must be run with merchant.jks located in the same folder.

Command to convert JKS keystore into PKCS#12

```
>keytool -importkeystore -srckeystore merchant.jks -srcstoretype JKS -deststoretype PKCS12 -destkeystore merchant.p12
```

Command to convert PKCS#12 keystore into PEM File

```
> openssl PKCS12 -in merchant.p12 -nocerts -out merchant.pem
```

In this example openssl is used, but other tools can be used.

After the end of above steps you end up with following files:

- merchant.jks
- merchant.p12
- merchant.pem

Appendix C - Advanced customization

Customizing the payment page flow

With Payment Page it is possible to adapt the flow to suit your specific needs. The following sub-sections describe options and requirements that must be considered.

Options for where to host the list of payment options

There are different required templates depending on what payment methods and options that are intend to be used:

Option 1: All payment options displayed on the Payment Page hosted at Digital River World Payments

All requests should be redirected to the relevant payment page, which may be either one International page or multiple different localized pages depending on how many pages you have built in Page Builder. On this page, the consumer makes a payment method selection and enters details (e.g. card number) if needed.

Option 2: (Some) payment options are listed on the merchant site

If the merchant wants to display a list of payment options on their own site, and use Digital River World Payments' Payment Page more as a technical interface to interface with all the banks, the redirect to Digital River World Payments should include the selected payment method ID. See the separate appendix for information about the PaymentMethodID parameter.

Using iframes

It's not recommended to use iframes due to various problems with e.g.:

- Third party cookies. When the browser visits a bank or DRWP, the cookies must be written in "full window". There's currently no mechanism to handle the timing between exploding the iFrame and writing the cookie.
- Cookies are handled in different ways in different web browsers.
- Web browsers are constantly struggling for Internet privacy and at the same time they want to use advertising cookies.

- Implementing a good user experience combining “full frame presentation” with “iFrame presentation” in redirection flows (IBP and 3DS) is challenging.

If a tight interaction between the merchant page and the payment flow is needed, the Client Side Encryption and REST API should be considered instead of using Payment Page with iFrames.

Appendix E – URLs

Environment	Description	Address
Acceptance test	Redirect URL	https://testpage.payments.digitalriver.com/pay/?creq=
Production	Redirect URL	https://secure.payments.digitalriver.com/pay/?creq=

Appendix F – Klarna

Klarna is an "Open Invoice" payment method. When the Consumer chooses to pay using Klarna then Klarna in effect buys the debt from the Merchant. The merchant is guaranteed payment and it is up to Klarna to make sure the order is actually paid.

Klarna does this in several ways; the Consumer can choose to get everything on one invoice to be paid within 14 days, or the Consumer can choose to pay in installments over a period of time.

From a payment flow perspective it looks similar to a Card payment;

The Merchant can choose either Authorize (via PaymentPage) + Capture (via API) flow or roll it all into one Debit (via PaymentPage).

The Merchant gets fulfillment on Authorize and is then free to ship the product. The actual invoice is sent to the Consumer when the Capture/Debit is sent to Klarna.

Since the result is an invoice to the Consumer all items in the order has to be specified. This means that each product in the order has to be represented as a Line Item in the redirect (note that there is a quantity for each line item so in case of several identical products it can be one line item with the appropriate quantity). The sum of all Line Items has to match the sum of the order in the redirect.

The Merchant can then choose to Capture only certain Line Items. Please refer to the API integration guide for more info.

The "payment instrument" in Klarna is a combination of Name, Address, Phone number, email and Social security number (or "Personnummer" in Sweden. Birthday + Gender in Germany etc.).

Since the Address is important Klarna has features for finding out the legally registered address of the Consumer. In the countries where this exists it is considered mandatory, and usually Klarna will not allow orders where the Billing and Shipping address differ (because of the fraud risk).

Attention: This means that if the Merchant passes a Billing/Shipping address to PaymentPage it might get overridden by the Consumer's actual address when selecting the payment method Klarna. This address will then be passed back to the Merchant in the response and this means that the Merchant has to ship there instead (in case of physical goods).

If both Billing and Shipping address is passed, then Shipping address will be used.

Payment Plans

The different variants of Klarna can be dynamically presented to the Consumer at the PaymentPage. Depending on what the Consumer chooses here it can incur a different cost for the Consumer (for example installments over a long time). The order amount is still the same though and the difference is only a matter between the Consumer and Klarna (who now owns the debt).

What options to present and the interest rates has to be negotiated between the Merchant and Klarna separately.

These options are knows as Payment Plans at DRWP, and Klarna calls them Pclasses.

Following are the getPaymentPlanResponse parameters:

Parameter	Comment
PaymentPlanCode	The payment plan code describes the length and type of installment that should be used.
TotalCost	The total amount that is payable for the purchase, including starting fee, administrative fees and interest rates.
MonthlyCost	Calculated monthly cost.
AnnualPercentageRate	Calculated annual percentage rate.
NumberOfPayments	Number of monthly payments.
Interest	Interest rate for payment option.
ArrangementFee	Arrangement fee.
Surcharge	Surcharge.
MinAmount	Minimal amount for the payment plan to be available on template
Type	a) Used to determine whether it is a fixed part payment or flexible part payment.

Parameter	Comment
	b) Type used is 0 or 1. c) 0 is used for Fixed Part Payment and 1 is used for Flexible Part Payment.
Description	Description of payment plans as supplied by Klarna.

Mandatory parameters

There are more mandatory parameters for Klarna than for other payment methods.

Most of these can be entered by the Consumer at the payment page.

But you as a Merchant is expected to send in:

BillingInfo Email

Line item data

But the Consumer is expected to provide much more, either at the Merchant site so that the merchant can send it in the redirect, or at the Payment Page. (see example templates for how to present this).