

Reseller onboarding process

1. **Reseller** redirects merchant to the platform.

The merchant from **Reseller** desires to use WD as payment processor for online payment of the orders on his shop. **Reseller** will gather a part of the information from the merchant and will redirect the merchant to the platform's registration page. The page URL needs to be constructed using the platform's branch (UK, Ireland, Austria or Germany), the language code (*en_GB*, *fr_FR*, *it_IT* and *de_DE* for now) and the path to the page. For example, the full URL for an English-speaking merchant assigned to the German branch of the platform based on country of business would be https://www.checkoutportal.de/en_GB/signup. The following information needs to be POSTed to this URL (JSON given here for ease, normal POST expected):

```
#!/javascript
{
  "merchant_id": "<id>", //to match in Reseller system
  "merchant_country": "<country code>", // 3-letter ISO
  "merchant_mcc": "<MCC>", //VISA MCC codes, to be provided by Wirecard
  "reseller_id": "<hash>" //pre-agreed reseller hash
}
```

The fields above are absolutely necessary for the onboarding process, but the reseller can send along extra information (if available) to shorten the registration form fill-in time for the merchant. Should the MCC and country of the merchant not be provided, an intermediary form will appear in the process asking for this information. We strongly recommend that, where available, this information be provided, for usability purposes. The other available fields are listed bellow as an HTML form:

```
#!/html
<form method="POST" action="https://www.checkoutportal.com/en_GB/signup/" target="_blank">
  <label>Merchant ID</label>
  <input type="text" name="merchant_id" placeholder="merchant_id"><br>
  <label>Merchant country</label>
  <input type="text" name="merchant_country" placeholder="eg. DEU - ISO alpha3 code"><br>
  <label>Merchant MCC</label>
  <input type="text" name="merchant_mcc" placeholder="eg. 5511"><br>
  <label>Package ID</label>
  <input type="text" name="package_id" placeholder="eg. RESELL1"><br>
  <label>Reseller ID</label>
  <input type="text" name="reseller_id" placeholder="eg. yw3664rwlkjh"><br>
  <label>Username</label>
  <input type="text" name="username" placeholder="username"><br>
  <label>Email</label>
  <input type="text" name="email" placeholder="email"><br>
  <label>Representative phone</label>
  <input type="text" name="representative_phone"
    placeholder="representative phone"><br>
  <label>Representative mobile</label>
  <input type="text" name="representative_mobile"
    placeholder="representative mobile"><br>
  <label>Representative salutation</label>
  <input type="text" name="representative_salutation"
    placeholder="m or f, one character"><br>
  <label>Representative first name</label>
  <input type="text" name="representative_first_name"
    placeholder="representative first name"><br>
  <label>Representative last name</label>
  <input type="text" name="representative_last_name"
    placeholder="representative last name"><br>
  <label>Representative address</label>
  <input type="text" name="representative_address"
    placeholder="representative address"><br>
  <label>Representative ZIP</label>
  <input type="text" name="representative_zip"
    placeholder="representative zip"><br>
```

```

<label>Representative city</label>
<input type="text" name="representative_city"
    placeholder="representative city"><br>
<label>Representative fax</label>
<input type="text" name="representative_fax"
    placeholder="representative fax"><br>
<label>Representative email</label>
<input type="text" name="representative_email"
    placeholder="representative email - if not provided, it will be user email"><br>
<label>Company name</label>
<input type="text" name="company_name" placeholder="company name"><br>
<label>Company address</label>
<input type="text" name="company_address" placeholder="company address"><br>
<label>Company city</label>
<input type="text" name="company_city" placeholder="company city"><br>
<label>Company ZIP</label>
<input type="text" name="company_zip" placeholder="company zip"><br>
<label>Company state</label>
<input type="text" name="company_state" placeholder="company state"><br>
<label>Company URL</label>
<input type="text" name="company_url" placeholder="company url - website of the company"><br>
<label>Shop URL</label>
<input type="text" name="shop_url" placeholder="shop url - the webshop address"><br>
<label>VAT number</label>
<input type="text" name="vat_number" placeholder="VAT number"><br>
<label>Bank name</label>
<input type="text" name="bank_name" placeholder="bank name"><br>
<label>Bank city</label>
<input type="text" name="bank_city" placeholder="bank city"><br>
<label>Bank IBAN</label>
<input type="text" name="bank_iban" placeholder="bank IBAN"><br>
<label>Bank SWIFT</label>
<input type="text" name="bank_swift" placeholder="bank SWIFT"><br>
<label>Bank code</label>
<input type="text" name="bank_code" placeholder="bank code"><br>
<label>Bank account number</label>
<input type="text" name="bank_account_number" placeholder="bank account number"><br>
<label>Bank owner</label>
<input type="text" name="bank_owner" placeholder="bank account owner full name"><br>
<input type="submit" value="Register merchant">
</form>

```

The reseller needs to provide WD two logos with transparent background, one for placing on top of a light-colored background, the other one for dark backgrounds. The logos must be suitable for display in a horizontal rectangular area. The reseller must also provide an email address for technical contact. The platform makes available notifications of all the stages of the onboarding process to the reseller via a secure (HTTPS) POST callback url - if the feature is desired, the reseller needs to provide this URL as well.

1. Merchant is presented with the signup page

The signup page contains the **Reseller**, the platform's and Wirecard logos and is translated to the language specified in the merchant_country POST field. The merchant fills in basic company and user information, necessary for the automatic compliance verification by Webshield. Upon completion, he is redirected to the merchant dashboard. At the same time, **Reseller** receives a POST at the designated end-point (**important!** the request is not a REST POST, but a form POST - for legacy reasons - with a JSON payload inside the postback variable), with the following info (given here in JSON):

```

#!javascript
{
    "merchant_id": "<id>",
    "merchant_status": "<status>", //PROCESSING
    "reseller_id": "<hash>"
}

```

2. Account activation

On the merchant dashboard, the user is presented with the additional information

request form. This information is necessary for the semi-automatic account setup using the SFTP upload process. The merchant dashboard form contains the fields for the information that needs to be acquired, the overview of the chosen payment methods and services (pre-selected, not editable) and the confirmation checkbox for the Terms and Conditions and Privacy Policy. The merchant is also required to upload documents for the company through a notification. Upon the verification of the documents, a new reseller notification is issued, with the status `DOCUMENTS_COMPLETE`, like this:

```
#!/javascript
{
  "merchant_id": "<id>",
  "merchant_status": "<status>", //DOCUMENTS_COMPLETE
  "reseller_id": "<hash>"
}
```

After the merchant fills in the information, depending on the Webshield compliance response, the merchant account is activated, either automatically or manually from the admin back-end. At this point, the Wirecard Checkout Page tokens should be ready to be sent to **Reseller**.

Should the merchant be rejected for any reason, the reseller will receive a notification with the new status:

```
#!/javascript
{
  "merchant_id": "<id>",
  "merchant_status": "<status>", //DECLINED or TERMINATED, if the merchant closed the shop
  "reseller_id": "<hash>"
}
```

3. Token transmission

For security concerns, the platform doesn't store Wirecard Checkout Page token information in the database. However, a method to securely transmit this information to **Reseller** is necessary to automate the process on the reseller side. We achieve this by adding a form for each merchant on the admin side that posts this information to the reseller provided end-point, using the reseller hash as identifier. The end-point should accept a JSON encoded object like:

```
#!/javascript
{
  "merchant_id": "<id>",
  "merchant_status": "<status>", //ACTIVE,
  "wirecard_credentials": {
    "<key>": "<value>"
    // key-value pairs with Wirecard Checkout Page tokens
    // usually 'customer_id' and 'secret'
  },
  "reseller_id": "<a hash>"
}
```

At this point the merchant should be set up and able to process transactions. **Reseller** needs to setup and enable the merchant payment integration using the received tokens.

If additional information is required (copy of passport, registration certificate), it will happen at step #3 and will be handled through the platform's interface.