

Card Payment

To process card transactions through a Checkout page, set the configuration

Card configuration

```
{
  ...
  "configurations": {
    "card": {
      "capture_now": Boolean,
      "dynamic_descriptor": String,
      "threed_secure": {
        "enabled": Boolean,
        "description": String,
        "redirect_url": String,
      }
    }
  }
  ...
}
```

- **capture_now** - If set to `true`, the transaction will be capture immediately. Else only an authorization will be done.
- **dynamic_descriptor** - This string will be used to describe the transaction on the customer's bank statement. Formatting requirements might vary per processor and can be seen at their [respective connection pages](#).
- **[threed_secure]**
- **[threed_secure.enabled]** - Whether 3-D Secure flow should take place for the transaction
- **[threed_secure.description]** - Short message to display on the page of the MPI, once the customer is redirected to authorize the 3-D Secure step
- **[threed_secure.redirect_url]** - An optional value

Account Verification

Checkout pages can be setup for processing Account Verification transactions. This is done as simply as providing the amount of the Checkout as 0. Same business logic applies to Account Verification transactions, as explained in the [card integration guide](#) - they cannot be captured, and 3DS data cannot be attached to them.

However, the Checkout allows for a **special way** of handling 3DS flow. It is still possible to set `threed_secure.enabled` to `true` and configure 3DS flow to be executed as additional step. But in this case, the 3DS flow will be performed for a different amount, for a future non-zero transaction, in which the Merchant can use the 3DS data resulting from this Checkout. This allows a Merchant the flexibility to verify the account of the shopper, and also perform 3DS ensuring that they are the cardholder, but only use the 3DS data for a transaction initiated later on. In order to do that, the `threed_secure` field has to be configured, with an additional field `amount`, which amount would be only used for the 3DS operation, and should match the amount of the transaction to be performed in the future.

Example payload for Checkout, performing Account Verification and 3DS operation for a future transaction with amount 33.15.

```
{
  "amount": 0,
  ...
  "configurations": {
    "card": {
      "capture_now": false,
      "threed_secure": {
        "enabled": true,
        "amount": 3315,
        ...
      }
    }
  }
  ...
}
```

If such Checkout is processed, the resulting redirection will contain both `transaction=` & `authentication_id` as part of the query parameters.

Credentials on File

It is possible to perform an initial Credentials on File transaction, where the shopper credentials get stored for future use. The same business rules and schema for the `recurring` parameter apply, as explained in the [card integration guide](#). Furthermore, those transactions can be configured as Account Verification, and/or to apply 3DS on top of them.

It is not possible to use Checkout for recurring one-click transactions, only for the initial tokenization. Follow up one-click transactions are done through the [transaction](#) endpoint.

Example schema for initial COF transaction via Checkout:

```
{
  ...
  "configurations": {
    "card": {
      ...
      "recurring": {
        "contract": String,
        "processing_model": String
      }
    }
  }
}
```

}
...
}
}