

## Checkout.com .NET Challenge

### Building a Payment Gateway

#### Background

E-Commerce is experiencing exponential growth and merchants who sell their goods or services online need a way to easily collect money from their customers.

We would like to build a payment gateway, an API based application that will allow a merchant to offer away for their shoppers to pay for their product.

Processing a payment online involves multiple steps and entities:



1. Shopper: Individual who is buying the product online.
2. Merchant: The seller of the product. For example, Apple or Amazon.
3. Payment Gateway: Responsible for validating requests, storing card information and forwarding payment requests and accepting payment responses to and from the acquiring bank.
4. Acquiring Bank: Allows us to do the actual retrieval of money from the shopper's card and payout to the merchant. It also performs some validation of the card information and then sends the payment details to the appropriate 3rd party organization for processing.

We will be building the payment gateway only and simulating the acquiring bank component in order to allow us to fully test the payment flow.

#### Requirements

The product requirements for this initial phase are the following:

1. A merchant should be able to process a payment through the payment gateway and receive either a successful or unsuccessful response
2. A merchant should be able to retrieve the details of a previously made payment

The next section will discuss each of these in more detail.

#### Process a payment

The payment gateway will need to provide merchants with a way to process a payment. To do this, the merchant should be able to submit a request to the payment gateway. A

payment request should include appropriate fields such as the card number, expiry month/date, amount, currency, and cvv.

**Note: Simulating the bank**

In your solution you should simulate or otherwise mock out the Bank part of processing flow (see diagram above). This component should be able to be switched out for a real bank once we move into production. We should assume that a bank response returns a unique identifier and a status that indicates whether the payment was successful.

**Retrieving a payment's details**

The second requirement for the payment gateway is to allow a merchant to retrieve details of a previously made payment using its identifier. Doing this will help the merchant with their reconciliation and reporting needs. The response should include a masked card number and card details along with a status code which indicates the result of the payment.

**Deliverables**

1. Build an API that allows a merchant
  - a. To process a payment through your payment gateway.
  - b. To retrieve details of a previously made payment.
2. Build a simulator to mock the responses from the bank to test the API from your first deliverable.

**Considerations**

Include whatever documentation/notes you feel is appropriate, this should include some details of assumptions made, areas you feel could be improved etc.

**Extra mile bonus points (not a requirement)**

In addition to the above, time permitting, consider the following suggestions for taking your implementation a step further:

- Application logging
- Application metrics
- Containerization
- Authentication
- API client
- Build script / CI
- Performance testing
- Encryption
- Data storage
- Anything else you feel may benefit your solution from a technical perspective.