

Process…?

* Simple “can I make a payment please?” request made **first** to Service to open the conversation. This in a way checks the Service is running before we go any further.
* If it is, the Service creates a record in Token table & returns unique @TokenID to be used by system in question in all future conversations around this transaction.
* Details of payment records are then sent back to Service, with @TokenID for validation.
* If @TokenID exists in Token table and date stamp is less than 1 minute ago (?) then add records received to Payment Order table
  + Data expected and entered into new record will be: Booking ID, Amount, Cost Centre, Account Code, Project Code.
  + Remaining fields remain NULL and will be updated after Transaction is attempted.
* Delete @TokenID record from Token table?? – its only meant to be used once.
* Call Payment Gateway asking for sum (Amount) from Payment Order table where TokenID = @TokenID to be taken. Include @TokenID in call to them as our unique identifier.   
    
  **Note:** Currently I think we are doing this but use **VendorTxCode**. *Maybe we should re-badge TokenID to VendorTxCode…*
* Gateway will send a URL with payment fields to complete from Payment Gateway
* Show this URL in new Browser Tab / Pop- Out Window as with Portal.
* Wait for Customer to complete Checkout Screen with card details and press Pay Now.
* Receive Reply for this @TokenID
* Populate new record in Payment Transaction Log whether it’s a successful transaction or not.
  + The Payment Gateway reply will include all data to fill new record in the table
* If successful payment taken, update records in Payment Order table with Success=1, processed datetime =getDate() and VPSTxID where TokenID = @TokenID
  + Show Success Message
* If successful payment not taken, update records in Payment Order table with Success=0 only where TokenID = @TokenID
  + Show Error Message
* Service replies to calling system with @TokenID, single Success Status (and possibly single Registration Status Detail so they can store and report locally on what the issue was if needed).  
    
  **FYI -** We require to give the calling application a single Success Status because there is no situation where some of the courses will be successfully paid and some not. The Transaction is a single record. Its all or nothing / all get paid or none get paid.
  + The only reason we need the Payment order table is for refunds, as these are done on a course by course basis and the service needs to know how much to refund an individual at that granularity.

SagePay in Detail

<https://www.sagepay.co.uk/support/15/36/sage-pay-server-understanding-the-process>

<http://integrations.sagepay.co.uk/content/getting-started-overview>