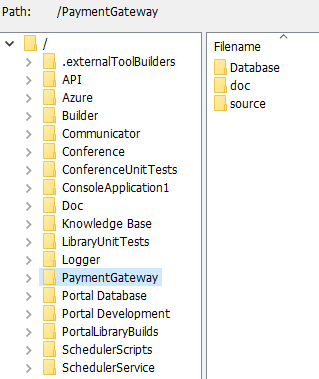
# Introduction

Payment Gateway is a single point of recording/reporting transactions from multiple applications to SagePay.

The Gateway is implemented as a MVC Web API. The requests from calling applications is to the API service. An administration web site is attached to the Web API for registering client applications & admin users to the site.

## Repository

The code currently resides on GIT in the Portal repository.



## Test Site

There is a trial installation of the application on the Portal UAT server.

<http://uat.paymentgateway.scot.nhs.uk/>

To access this site you need to add the following entry to your local hosts file:

51.141.226.164 uat.paymentgateway.scot.nhs.uk

The site requires login details.

#### Login Details

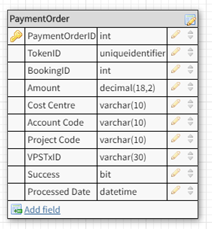
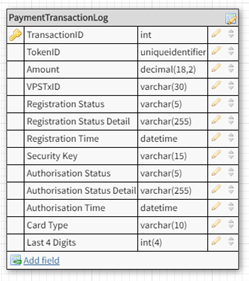
UName: NesPGAdmin

Pwd: s6aNeSpuZi

## Database

The database for the site is also on the Portal UAT server. Scripts for creating database tables are in GIT.

Database User: NESAdmin, Password: myPassword@123

**NOTE – This is initial structure. Some fields may have changed.**

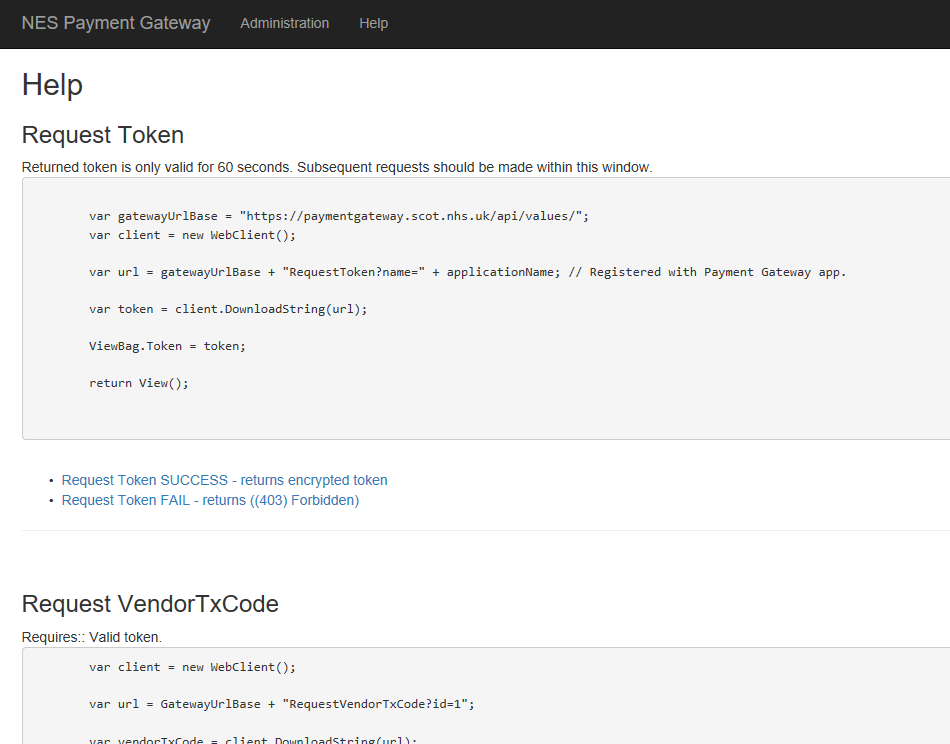
## Token

The token is not stored in the database (as indicated). It is generated upon request and attached to the HEADER of subsequent REQUESTS. The token consist of the Application name + Timestamp – encrypted.

Upon use, both the decrypted token is checked for valid Application name and timestamp within 60 seconds of initial request.

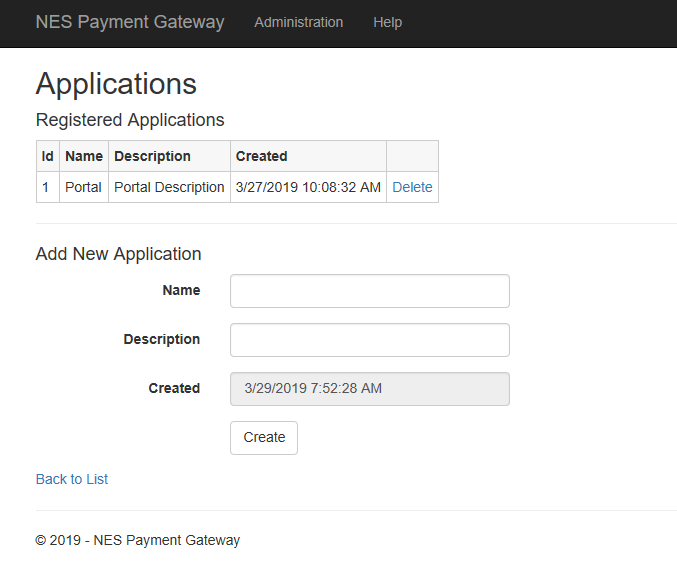
## Example Code for integration into client application

The test site has page with example code which is a basis for integration into any calling application.



## Application Registration

Any application calling the Web API must be registered in the database – Admin functionality in the web site.



# Notes

## Code Structure

Dependency Injection implemented using Ninject.

Security implemented using custom ActionFilters

The following areas of the code still need completing:

#### Error Logging

Error handling is only implemented by adding a logging filter at base level for all controller actions. No custom exception handling has been implemented.

The filter based logging still requires configuration.

#### Routing

The route mapping for actions currently is not working correctly for Actions without parameters. If you examine the code you will notice that some calls use a dummy parameter so that the routing picks up the Action.

#### VendorTxCode

The VendorTxCode is a randomly generated string, following a format expected by SagePay. This has been implemented as an Action in the Web API, however the returned string includes a further set of escaped quotes which are messing up the generated code. Time constraints prevented the investigation into this and the quickest solution was to move the block of code generating the VendorTxCode to the calling application.

#### PaymentTransactionLog

This is a very simple implementation for proof of concept only. For a fuller example of returned message handling look at the Portal code: -

PaymentHelper.PaymentNotification(HttpRequestBase(Request, bool IsMobile)