



eNETS Transaction Query Implementation Guide

Version 1.1

Transaction Query Implementation Guide

Contents

1	Introduction	3
1.1	Objective.....	3
1.2	Audience.....	3
2	Functionality and Features	4
2.1	Overview	4
2.2	Transaction Query URL	5
3	Sample Codes.....	Error! Bookmark not defined.
3.1	Transaction Query Request/Response – UMAPI - Java.....	Error! Bookmark not defined.
3.2	Transaction Query Request/Response – UMAPI .Net... ..	Error! Bookmark not defined.
3.3	Transaction Query Request/Response – UMAPI Lite	Error! Bookmark not defined.
	Appendix A.....	7

Transaction Query Implementation Guide

1 Introduction

1.1 Objective

The purpose of this guide is to provide eNETS merchants who wish to use transaction query function with necessary technical information to implement their applications with eNETS transaction query service.

1.2 Audience

This guide is intended for developers and system integrators of eNETS merchants who are integrating merchants' applications with eNETS transaction query service.

Transaction Query Implementation Guide

2 Functionality and Features

2.1 Overview

eNETS query server provides the facility to merchants to query the status of a transaction previously carried out. Merchant will provide the required input parameters to query the transaction status.

There are two approaches to query the status of transaction from merchant end.

- Passing "**nets_txn_ref**" parameter to query with the unique eNETS transaction reference code.
- Passing "**mid**" and "**merch_txn_ref**" 2 parameters to query with both merchant id and merchant transaction reference.

Example for the two approaches:

No	Portal	URL
1	Type 1	https://admin.enets.sg/txnquery/TxnStatus?nets_txn_ref=20140925000006618
2	Type 2	https://admin.enets.sg/txnquery/TxnStatus?mid=917369000&merch_txn_ref=47895188

Upon submission of the query transaction request, eNETS will process the request based on the merchant provided parameters.

The enquiry result is returned as **<stage>_<responsecode>**

The stage code and response code can be found at Appendix A in the complete merchant integration manual for Java & .NET v2.8

Please take note only the full value of the following stage code_ response code will be interpreted as a successful and settled with bank.

- "1003_00" (credit)
- "000008_000000" (debit)
- "000004_000000" (debit)

Transaction Query Implementation Guide

Additional Note:

Multiple <stage>_<responsecode> will also be returned to upon query if the same transaction reference have being transact more than 1 time.

For example:

- Transaction reference ABC123 is used to transact the first time, it encountered 000001_040011
- ABC123 is again used to transact for the second time, it will be rejected by our gateway with -1223 error "Duplicate Transaction"
- If ABC123 is again used to transact for the third time, it will again be rejected by our gateway -1223 error "Duplicate Transaction"
- If ABC123 is continue to be used to transact for "x" number of time the outcome is as follows

000001_040011;2004_-1223;2004_-1223;<follow by the same error for "x" number of time>

Each transaction result tries is segregated by ";"

2.2 Transaction Query URL

- Production URL → <https://admin.enets.sg/txnquery/TxnStatus>
- UAT URL → <https://test2.enets.sg/txnquery/TxnStatus>

Transaction Query Implementation Guide

3 Sample Code – Query Transaction Status

```
package sg.enets.example;

import java.io.BufferedReader;
import java.io.DataOutputStream;
import java.io.InputStreamReader;
import java.net.URL;
import javax.net.ssl.HttpURLConnection;

public class QueryTxnStatus
{
    public static void main(String[] args) throws Exception
    {
        URL url = null;
        HttpURLConnection httpsURLConnection = null;
        url = new URL("https://test2.enets.sg/txnquery/TxnStatus");
        httpsURLConnection = (HttpURLConnection) url.openConnection();
        httpsURLConnection.setRequestMethod("POST");
        httpsURLConnection.setRequestProperty("User-Agent", "Mozilla/5.0");
        httpsURLConnection.setRequestProperty("Accept-Language", "en-US,en;q=0.5");
        httpsURLConnection.setDoOutput(true);
        DataOutputStream wr = new DataOutputStream(httpsURLConnection.getOutputStream());

        //Remark: There are 2 methods to invoke the Query Transaction Status function and merchants may use either one.
        //      Method (1) uses nets_txn_ref
        //      Method (2) uses mid & merch_txn_ref
        // Method (1) uses nets_txn_ref
        String nets_txn_ref = "20160908143139561";
        String urlParameters = "nets_txn_ref=" + nets_txn_ref;

        // Method (2) uses mid & merch_txn_ref
        //String mid = "888454000";
        //String merch_txn_ref = "20160908143138225562";
        //String urlParameters = "mid=" + mid + "&merch_txn_ref=" + merch_txn_ref;

        wr.writeBytes(urlParameters);
        wr.flush();
        wr.close();
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(httpsURLConnection.getInputStream()));
        System.out.println("Stage Response Code = " + bufferedReader.readLine());
        bufferedReader.close();
    }
}
```

Transaction Query Implementation Guide

Appendix A

Change History

Version No.	Revision No.	Description	Approval Date
1	0	Created document Transaction Query Implementation Guide	8 Sep 2016
	1	Section 2 : Update test URL Section 3 : Update Sample Code to use HTTPS POST	3 July 2017

Documents and Reference Materials

Reference Number	Document Name	Remarks
1	UMAPI Merchant Integration Manual – Download Java Manual	Version 2.8
2	UMAPI Merchant Integration Manual – Download .NET Manual	Version 2.8

Terminology and Convention

Term / Abbreviation	Meaning	Remarks
UMAPI	Unified Merchant API	UMAPI