# API Gateway HMAC Authentication Guide (for JAVA)

## Overview

The API Gateway is a platform that supports authentication, quota limit, statistics and format conversion to provide NAVER contents data through a single channel.

This document describes how to call NAVER service APIs using HMAC authentication, which is one of the authentication methods of API Gateway.

## How to Authenticate

To call NAVER service APIs using HMAC authentication, you should get an encrypted value for the URL to be called. The EncryptedUrl is created with the combination of a developer's key that is issued for each developer, API URL and call time.

* Developer's key: A unique key that is provided as a file for each developer, by NAVER.
* API URL: HTTP URL to be called for executing an API
* Call time: Time serial to call an API

The API Gateway uses the received URL's hash value to check whether the request is valid.

## Setup

* Save the setup files, which are listed in "6. Setup Files," in your local folder.
* Add the ApiGateway-MAC.jar to the dependency library of your application.
* Save the NHNAPIGatewayKey.properties in the classpath.

## How to Use

* Set a URL to be called.

String url = "http://dev.apis.naver.com/Developer\_ID/hmac/hmactest.xml";

* Call the getEncryptUrl of MACManager to get an encrypted URL. You should get a new encryptedUrl every time you call a service API.

String encryptedUrl = MACManager.getEncryptUrl(url);

* Request data with the created encryptedUrl.

## Result

The following results are returned depending on whether HMAC authentication succeeds or fails.

* When failed: An error code and message that reads authentication failure is returned.

e.g.) The result of authentication failure is returned in the form of XML.

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<result>

<message>HMAC Authentication failed</message>

<error\_code>022</error\_code>

</result>

* When successful: A result of API call is returned.

e.g.) The result when HMAC authentication test is successful is returned.

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<result>

<authentication>true</authentication>

</result>

## Setup Files

|  |  |
| --- | --- |
| File | Description |
| ApiGateway-MAC.jar | A library file for authentication, provided by NAVER |
| NHNAPIGatewayKey.properties | A property file that contains a key issued for each game publisher |

## Example

The following example checks whether HMAC authentication is applied. Replace "Developer\_ID" with your ID in the example below, and test the code. In this example, we used the apache-common project to handle the http request.

/\*

\* @(#)sample.java $version 2010. 3. 17

\*

\* Copyright 2010 NAVER Corp. All rights Reserved.

\* NAVER PROPRIETARY. Use is subject to license terms.

\*/

**package** com.nhn.api;

**import** org.apache.commons.httpclient.HttpClient;

**import** org.apache.commons.httpclient.HttpMethod;

**import** org.apache.commons.httpclient.methods.GetMethod;

**import** com.naver.api.security.client.MACManager;

/\*\*

\* The HMAC Authentication sample.

\*/

**public** **class** HmacSample {

/\*\*

\* The main method.

\*

\* **@param** args the arguments

\*/

**public** **static** **void** main(String[] args) {

String url = "http://dev.apis.naver.com/Developer\_ID/hmac/hmactest.xml?param=paramValue";

// initialize MACManager

MACManager.*initialize*();

// Generate Encrypted url

String encryptedUrl = MACManager.*getEncryptUrl*(url);

// for Apache HTTP client.

HttpClient client = **new** HttpClient();

HttpMethod method = **new** GetMethod(encryptedUrl);

// call api of NAVER api-gateway

client.executeMethod(method);

/\* ============== success return ============================

\* <result>

\* <authentication>true</authentication>

\* </result>

\* ============== fail return ============================

\* <result>

\* <message>Authentication Exception</message>

\* <error\_code>022</error\_code>

\* <result>

\* ==========================================================

\* \*/

System.*out*.println(method.getResponseBodyAsString());

}

## Reference URL

* Apache commons project : <http://commons.apache.org/>
* HMAC Overview: <http://en.wikipedia.org/wiki/HMAC>