# API Gateway HMAC Authentication Guide (for C++)

## 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, for UNIX/Linux and Windows.

## 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, and API URL.

* 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

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

## Setup

### UNIX/Linux

1. Unzip the file you received. The list of setup files are shown in "6. Setup Files."
2. Save the libApiGateway-MAC.so in the path that your application can access.   
   e.g.) $PROGRAM\_ROOT/lib/libApiGateway-MAC.so
3. Save the MACManager.h in the path that your application can access.  
   e.g.) $PROGRAM\_ROOT/lib/include/MACManager.h
4. Save the NHNAPIGatewayKey.properties in the path that your application can access. The MACManager::getKeyFromFile() basically reads the NHNAPIGatewayKey.properties from $PROGRAM\_ROOT. If the NHNAPIGatewayKey.properties is in another path, you should use the MACManager::getKeyFromFile(const std::string& filePath) method.
5. Link the libApiGateway-MAC.so with your application build.  
   e.g.) When there is $PROGRAM\_ROOT/lib/libApiGateway-MAC.so,   
    **g++ $(OPTIONS) –Wl,-rpath=lib –Llib –lApiGateway-MAC $(SOURCES) –o $(TARGET)**

### Windows

1. Unzip the file you received. The list of setup files are shown in "6. Setup Files."
2. Save the MACManager.dll in the path that your application can access.   
   e.g.) **$(SolutionDir)\DLL\Debug\libApiGateway-MAC.dll   
    $(SolutionDir)\DLL\Release\libApiGateway-MAC.dll**
3. Click Project **Properties > Build Events > Post-Build Event > Command Line** and specify the following setting, so that the .dll file is copied to the path where the executable file is located when you build the program.   
   e.g.) **copy /Y $(SolutionDir)DLL\$(Configuration)\libApiGateway-MAC.dll $(SolutionDir)$(Configuration)\**
4. Save the MACManager.h in the path where the project's header files are located, click **Project Properties > C/C++ > General > Additional Include Directories** and set "$(SolutionDir)include."
5. Save the libGApiGateway-MAC.lib in the project's library folder, and set the project as follows:

* Click **Project Properties > Linker > General > Additional Library Directories** and set "$(SolutionDir)lib."
* Click **Project Properties > Linker > Input > Additional Dependencies** and set "libApiGateway-MAC.lib."

## How to Use

How to use libraries is the same for both \*NIX system and Windows system. The following example code is written for \*NIX system, and basically the same code can be used for Windows system as well. However, you may need to change part of it depending on your development environment, because include path or commands such as pragma comment may differ.

1. Set a URL to be called. It is the API URL provided for each game publisher.

#include “lib/include/MACManager.h”

const std::string URL = “http://dev.apis.naver.com/Developer\_ID/hmac/hmactest.xml”;

1. Specify a key value that is defined in the NHNAPIGatewayKey.properties provided for each game publisher.

// When you directly specify a key value in the source code

const std::string KEY = “xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx”;

// When you get a key value from NHNAPIGatewayKey.properties of $PROGRAM\_ROOT, the default path

const std::string KEY = MACManager::getKeyFromFile();

// When you get a key value from the resources/NHNAPIGatewayKey.properties, a custom path

const std::string filePath = “resources/NHNAPIGatewayKey.properties”;

const std::string KEY = MACManager::getKeyFromFile(filePath);

1. Call the getEncryptUrl(const std::string& url) of MACManager to get an encrypted URL. The url parameter of getEncryptUrl should include the URL to be called and all the parameters to be passed with GET method. The encryptedUrl created by the getEncryptUrl includes additional parameters, msgpad and md.

// Create an object in Stack

MACManager manager(KEY);

std::string encryptedUrl = manager.getEncryptUrl(URL);

// Create an object in Heap

MACManager\* pManager = new MACManager(KEY);

std::string encryptedUrl = pManager->getEncryptUrl(URL);

delete pManager;

1. Request data from the APIGW server, with the encryptedUrl created in step 3.

## Result

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

### When successful

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

<result>

<authentication>true</authentication>

</result>

### When failed (example)

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

<result>

<message>HMAC Authentication failed</message>

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

</result>

### Error Code

| Error Code | Error Message | Description |
| --- | --- | --- |
| 000 | System error | System error |
| 010 | Your query request count is over the limit | The number of query requests exceeds the limit. Generally, this error occurs when the number of requests exceeds 25,000, which can be changed. |
| 011 | Incorrect query request | This error occurs when a query (query=) field does not exist. The query entered in the form of "...&query=&" is perceived as a valid query request. |
| 020 | Unregistered key | Unregistered key |
| 021 | Your key is temporarily unavailable | This error occurs when a query is requested with a key that is registered in OpenAPI but temporarily unavailable. |
| 022 | Authentication Fail | Authentication failed. |
| 030 | UnsupportedFormatException | Unsupported return format. |
| 040 | NoSuchApiException | The API does not exist. |
| 100 | Invalid target value | This error occurs when you use a target name that is not included in the field description, or there is no target field. |
| 101 | Invalid display value | This error occurs when the display value is over 100, less than or equal to 0, or an empty value such as "...&display=&" is entered. When there is no display field, it is perceived as a valid query request and "display=10" is set by default. |
| 102 | Invalid start value | This error occurs when the start value is over 1000, less than or equal to 0, or an empty value such as "...&start=&" is entered. When there is no start field, it is perceived as a valid query request and "start=1" is set by default. |
| 110 | Undefined sort value | This error occurs when the sort value is not one of the sort options supported by the corresponding service, or an empty value such as "...&sort=&" is entered (for example, when "sort=asc", one of the sort options of Shopping search, is set for Knowledge iN search query request).  When there is no sort field, it is perceived as a valid query request. For the service API with a sort option, "sort=sim" is set by default. |
| 200 | Reserved | Reserved |
| 900 | Undefined error occurred | Undefined error has occurred. |

## Setup Files

### UNIX/Linux

|  |  |
| --- | --- |
| File | Description |
| libApiGateway-MAC.so | A library file for authentication, provided by NAVER |
| MACManager.h | A header file provided by the authentication library |
| NHNAPIGatewayKey.properties | A property file that contains a key issued for each game publisher |

### Windows

|  |  |
| --- | --- |
| File | Description |
| libApiGateway-MAC.dll | A library file for authentication, provided by NAVER |
| libApiGateway-MAC.lib | A link library file for load-time linking |
| MACManager.h | A header file provided by the authentication library |
| NHNAPIGatewayKey.properties | A property file that contains a key issued for each game publisher |

## Other Library

### UNIX/Linux

OpenSSL

Many \*NIX systems come with pre-compiled OpenSSL packages. If you do not have OpenSSL, you can use yum install openssl.

Reference: <http://www.openssl.org/>