A Guide to the Development of NAVER Login

1. Outline

2.2 Functions provided

NAVER Login provides the following services.

2.2.1 Social login

The fast and secure NAVER Login allows you to use its services more conveniently without the need to go through complicated and cumbersome procedures for membership registration.
Users do not have to worry about entering a variety of their personal information, the inconvenience of their personnel authentication, and their own IDs and passwords that are easily forgotten.

The 42 million users being registered on NAVER are customers who will be with you.
Please give the convenience to users through the easy and convenient login method.

2.2.2 Searching member profiles linked to login

If NAVER Login is applied, you can obtain information about the profile of users, linked to NAVER Login, with their consent and use it for your services.
Profile information includes users' names, email addresses, nicknames, profile pictures, birthdays, age groups, and genders that you can check simply with the API.

2.2.3 Provision of statistical data on the use of services

If NAVER Login is applied, you can easily check statistical data on the use of the service through the menu of Login Statistics in the Developer Center, even if statistical information is not separately established.

2.3 Service environment

NAVER Login supports a variety of service environments.
You can provide users with a more effective environment by applying NAVER Login according to the service environment.

2.3.1 Web/mobile web service

NAVER Login is available in both web/mobile web service environments.
The service for NAVER Login is being provided through the oAuth 2.0 Open Protocol.
The oAUTH2.0 Open-Source Library allows it to be easily applied in various development environments.
In addition, it can be applied more easily and conveniently through the provision of javascript plugin.

2.3.2 Mobile web

NAVER Login provides you with an SDK dedicated to mobile apps to support various service environments.
If NAVER Login is applied using the SDK dedicated to mobile apps, it can be of great help in the provision of services because not only can development tasks be carried out in an easier and simpler way, but also the service users can easily use NAVER Login.
Currently, NAVER provides you with SDKs for each of iOS and Android.

Go to the guide for iOS development >

Go to the guide for Andorid development >

2.3.3 Compatibility between the browser and mobile OS

The SDK provided by NAVER for the convenience of development operates smoothly in the following versions.

Javascript SDK

- Required library
  - jQuery 1.10.0 or higher
- IE 7 or higher
- Other modern browsers (Safari, Firefox, Chrome, etc.)

Android SDK

- Supported Android OS version
    - Android 2.2 or higher

iOS SDK

- Supported iOS version
    - iOS 7 or higher

3. Application of NAVER Login

3.1 Checklists before the application of NAVER Login

3.1.1 Confirmation of the service environment

You must first confirm your service environment to apply NAVER Login to that service.
The environment that supports NAVER Login can be divided into the following four environments.

- PCWEB / MOBILEWEB: Web service environment
- Android: Android application environment
- iOS: iOS application environment
- Windows: Windows desktop program environment

Application registration and development methods may vary slightly depending on each of service environments.

3.1.2 Application registration

In order to apply NAVER Login, you must first register the application via the 'Application-Application Registration' menu.

3.1.3 Confirmation of required items

When the application is fully registered in the NAVER Developer Center, you can confirm the registered application in the 'Application-My Applications' menu in the Developer Center.
The following items must be confirmed in order for NAVER Login to be applied correctly.

1. The name of the application must be written clearly and concisely.
2. Users can see the name of the application when it is linked with NAVER Login, so that you must not use "unknown characters" or "names not related to the service.”
3. The logo image must be set as an image that can represent the service while conforming to the specifications.
   Users can see the logo image when it is linked with NAVER Login, so that it must be either an icon or image that can represent the service.
4. You must select "NAVER Login" for the API you are using to enter additional information.
   For an open API service environment at login, you must select more than one service environment.
5. The representative URL (homepage URL) of the service must be accurately entered in the web service environment.
6. You must make sure that the AppScheme and package name are entered correctly among the information about the basic settings of the application.

3.1.4 Request for pre-inspection

You must register a request for the inspection of the application before its application to the actual service after it has been fully developed. Only when it is approved after being inspected can you use NAVER Login normally without any limitation on IDs that can be used to log in

If you would like to know the standards for inspection, how to apply for inspection, tips for submitting inspection data, etc., please refer to the NAVER Login guide for inspection.

It is recommended that you familiarize yourself with the standards for inspection introduced in the guide for inspection from the design stage so that the NAVER Login can be applied according to the standards for inspection.

Go to the guide for inspection >

3.1.5 Develop your application using NAVER Login's SDK.

NAVER Login provides you with an SDK that supports JavaScript, Android, and iOS. The SDK includes sample applications, allowing you to implement authentication for login and API calls very easily.

Download NAVER Login's SDK >

In addition, If the SDK is used to develop mobile applications, users can use simple login via the NAVER app, suggesting that they can log in much more conveniently.

3.1.6 Regarding the image of the NAVER Login button

NAVER Login provides a default image for the NAVER Login button available in applications.
The design of the button image may be changed according to the situation of the application, but it is recommended that you keep the following guidelines as much as possible in order to maintain NAVER's own identity.

Go to the guide for the use of the login button >

If NAVER Login is applied using the iOS SDK or Android SDK, button resources are provided by the SDK itself.

3.2 Confirmation of member information and linkage with the existing members

3.2.1 Profile information of members using NAVER Login

Members can obtain access tokens upon successful completion of linking with NAVER Login.
It is possible to search the profile information of users through the access tokens.

Available profile information

- Unique identifier (not NAVER ID but unique identifier)
- Name
- Nickname (nickname on NAVER)
- Profile image (profile image URL in My Info on NAVER)
- E-mail address (e-mail in My Info on NAVER)
- Birthday
- Age group
- Gender
- Year of birth
- Cell phone number

The specifications of each piece of profile information are as follows.

- Unique identifier: BASE64 strings consisting of up to 64 characters
    - (Applied to applications created after May 1, 2021. Figures from the existing INT64 standard)
- Name: Strings consisting of 10 characters or less
- Nickname: Strings consisting of 20 characters or less
- Profile image: URL strings consisting of 255 characters or less
- E-mail address: Strings in image specifications
- Birthday: Strings like month-day(MM-DD)
- Age group: Strings represented as 0-9 / 10-19 / 20-29 / 30-39 / 40-49 / 50-59 / 60- according to the age range
- Gender: M/F Strings expressed as (male/female)
- Year of birth: Strings in the format of year (YYYY)
- Cell phone number: Strings of mobile phone numbers containing dashes (-)

Examples of profile information

- Unique identifier: abcdefgABCDEFG1234567
- Name: NAVER
- Nickname: NAVER nickname
- Profile image: https://phinf.pstatic.net/.../image.jpg
- E-mail address: naveridlogin@naver.com
- Birthday: 08-15
- Age group: 20-29
- Gender: F
- Date of birth: 1900
- Cell phone number: 010-0000-0000

The profile information provided by NAVER, except for the "unique identifier," may not be provided depending on the users' choice.

3.2.2 Whether logged-in members use NAVER Login

Users who have linked to login via NAVER Login have a unique user ID for distinguishing each of them.
The unique identifier is information that can be searched through an API for searching profile information.

Unique identifier

- BASE64 strings consisting of up to 64 characters
  - o (Applied to applications created after May 1, 2021. The existing applications consist of numbers in the INT64 specification.)
- Unique value by NAVER ID
- Unique identifiers are not shared between applications.

The service uses the unique identifier as the identification value of users who have used NAVER Login, so that it can perform some tasks such as searching the information of the users, handling their subscription, etc.

3.2.3 Linking existing members to NAVER Login

If you already have members in an existing service, you can use the following information to minimize their duplicate subscriptions.
Normally, users linked with NAVER Login can be identified by a unique identifier.
The profile information of users provided by NAVER, as well as unique identifiers, identifies existing registered users.

NAVER's user-based profile information

- E-mail address
- Gender
- Name
- Age group
- Birthday

For users who have performed the first link (if there are no users with the same unique ID in the database), it is possible to “link” an account for NAVER Login to an existing user account by searching user information with a combination of the above-listed user profile information.

In general, it is appropriate to apply in the following situations.

- If there is an existing membership system of the service,
- If existing service subscribers request to maintain unique information about the service (points, posts, etc.),
- If the login process is simplified while maintaining the service's unique information about the existing service subscribers (login via NAVER Login)

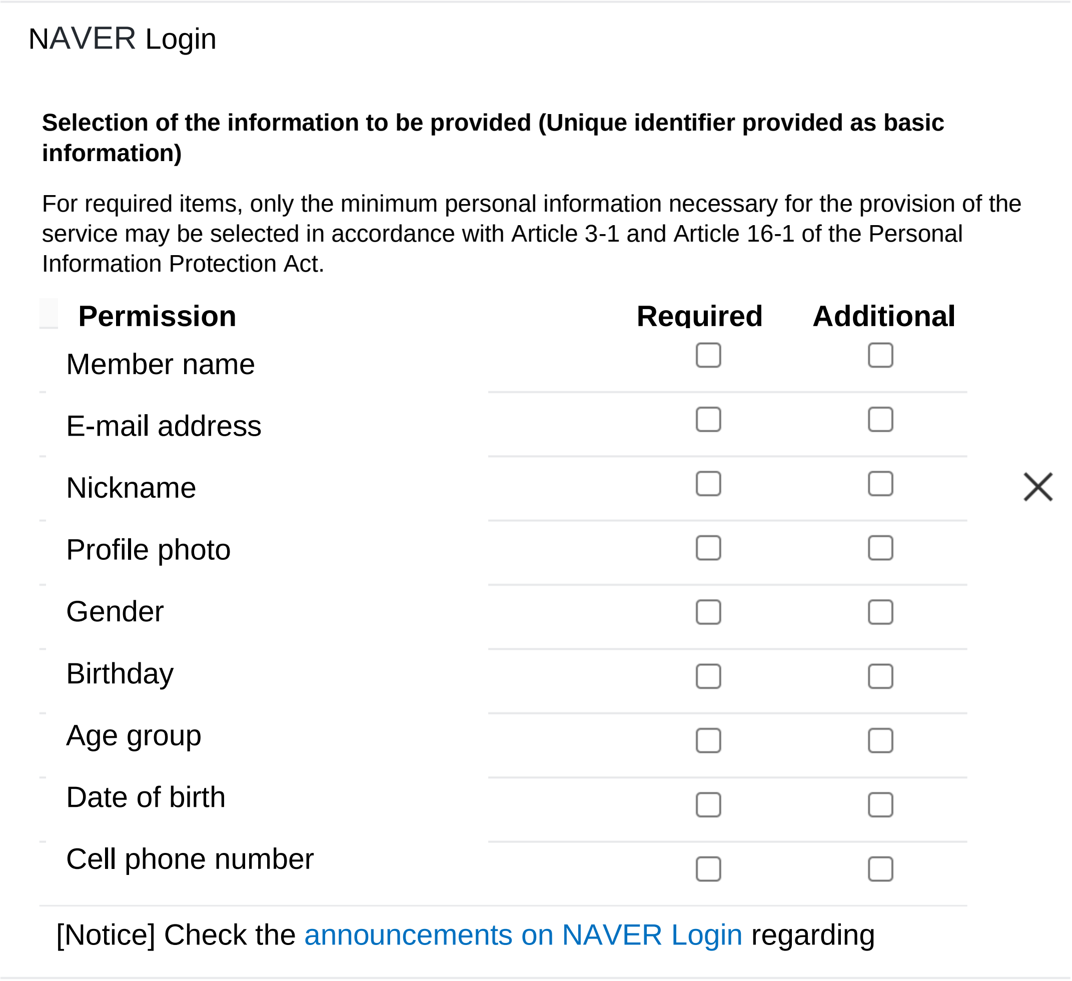
3.2.4 Regarding the optional provision of profile items by users

Users linked to login with NAVER Login can choose ***not to provide*** certain profile items to others at the first linkage.

The profile item provided by NAVER by default is a ***unique identifier***, and whether or not all profile items, except for items provided by default, are delivered is determined by a response from the API that searches profile information according to the user's selection.

Profile items to be provided in the application may be selected from the "My Application" menu. Profile items that you do not want to receive may be selected as ***not being provided*** through the corresponding menu.

Each profile item may be selected as required to be provided or additionally provided. The ***required items*** are the information that must be provided for users to use "My Service." In NAVER Login, all profile items, except ***unique identifiers***, are set to be selectively provided by users. The required items are shown to users so that they are at least not rejected. However, even in this case, users can select whether or not each item is to be provided, and each item is shown by default as ***being provided***. The ***items additionally provided*** are the information that users want to receive additionally, and each of them are shown by default as ***not being provided***.



NAVER Login

Selection of the information to be provided (Unique identifier provided as basic information)

For required items, only the minimum personal information necessary for the provision of the service may be selected in accordance with Article 3-1 and Article 16-1 of the Personal Information Protection Act.

Permission	Required	Additional
Member name	□	□
E-mail address	□	□
Nickname	□	□
Profile photo	□	□
Gender	□	□
Birthday	□	□
Age group	□	□
Date of birth	□	□
Cell phone number	□	□

[Notice] Check the announcements on NAVER Login regarding

3.2.5 Handling of permission for profiles users have rejected

Users can choose not to provide certain profile items according to their decision when they agree to link with NAVER login. In this case, the service cannot use profile search to obtain information about profile items rejected by users. In the service, it is possible to handle the rejected permission by selecting as follows.

1. The service may continue without obtaining the rejected profile items without further processing.
2. The service may continue by requesting that users manually enter the rejected profile items.
3. The service may obtain consent from users once again after sufficiently notifying them of the reasons why the rejected profile items must be delivered. Please refer to the following items for the related matters. (Item 5.15 of the guide)

3.3 Development of linkage with NAVER Login

3.3.1 Prior to the development of linkage with NAVER Login

There is a need to first register the application through NAVER Developer Center to apply NAVER Login.
You can see the Client ID and Client Secrets values of the applications you have registered in the 'My Applications' menu of the Developer Center.

About Client ID and Client Secrets

Client IDs and Client Secrets are important pieces of information that distinguish your application from others. Be sure to keep them safe.
In addition, they are information used in the process of linking with NAVER Login. Incorrect Client ID / Client Secrets information may lead to the linkage failure.
Once issued, the Client ID can't be changed, but the Client Secrets can be reissued through the Developer Center. If leakage of the Client Secrets is suspected, it may be reissued to prevent theft from occurring.

Client ID and Client Secrets specifications

- Client ID: A string of 40 characters or less with a combination of Alphabet uppercase and lowercase letters and numbers
- Client Secrets: A string of 40 characters or less with a combination of Alphabet uppercase and lowercase letters and numbers

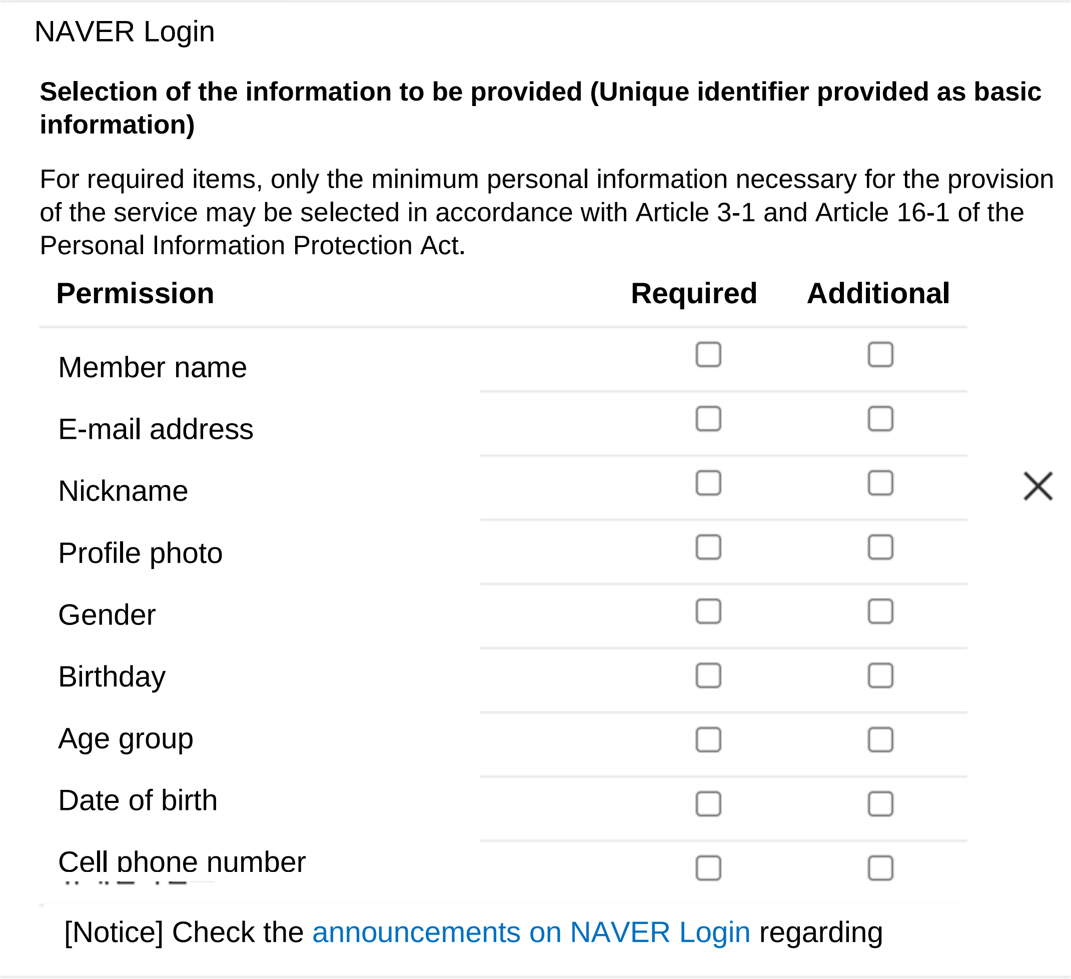
About API permissions

You can develop the service by using the login open API provided when using NAVER Login. You must set the permissions so that the API calls can be made from the application to use the open API.

- Setting API permissions: You can set API permissions you want to use in the 'Manage API Permissions' tab of 'My Applications'.

If there are no permissions to call the API

If you have not set the API permissions or users have not agreed to the permissions when logging in to NAVER, the API calls may result in the failures.
Accordingly, you must check the permission settings in order to use the API smoothly.



NAVER Login

Selection of the information to be provided (Unique identifier provided as basic information)

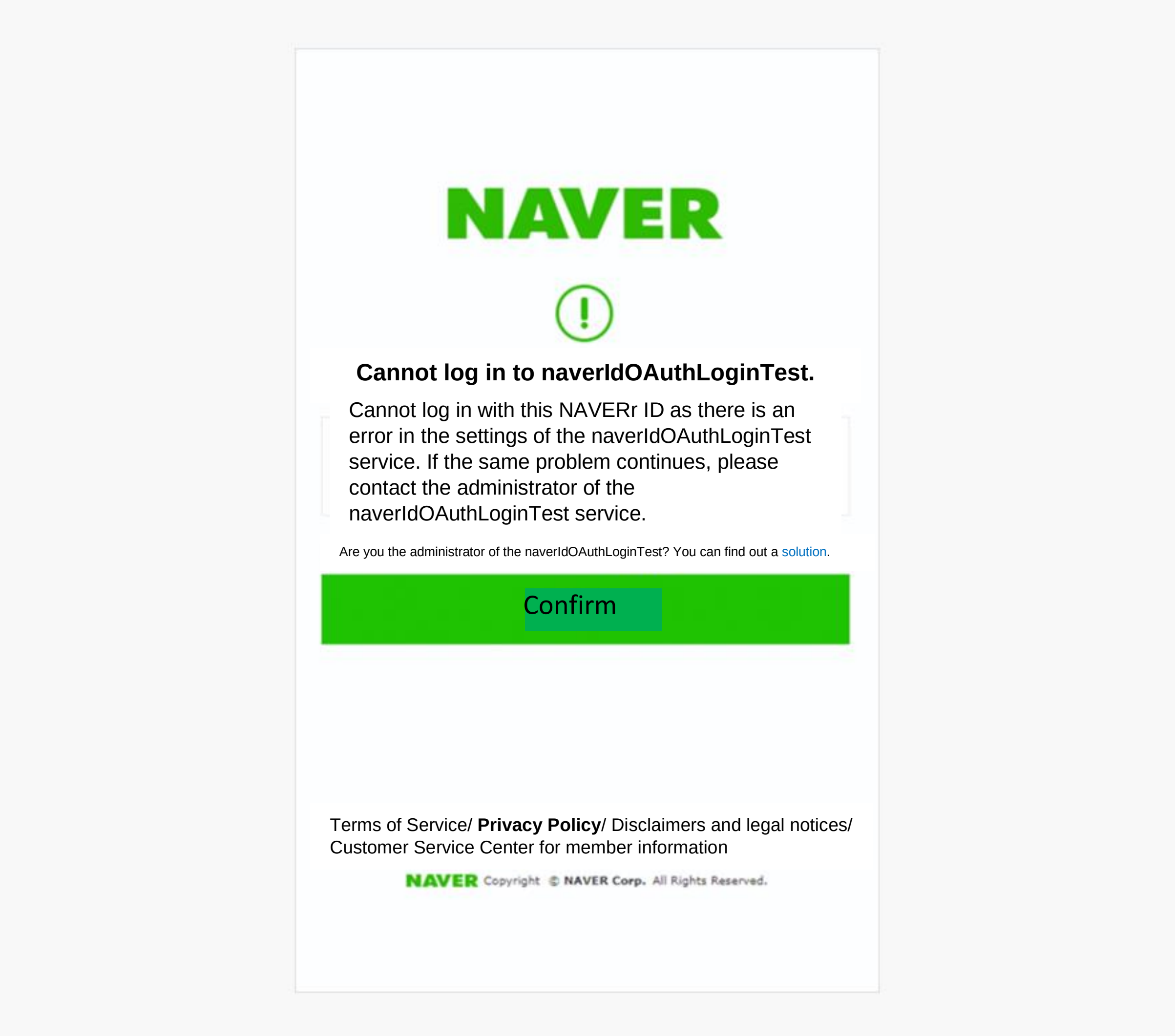
For required items, only the minimum personal information necessary for the provision of the service may be selected in accordance with Article 3-1 and Article 16-1 of the Personal Information Protection Act.

Permission	Required	Additional
Member name	□	□
E-mail address	□	□
Nickname	□	□
Profile photo	□	□
Gender	□	□
Birthday	□	□
Age group	□	□
Date of birth	□	□
Cell phone number	□	□

[Notice] Check the announcements on NAVER Login regarding

If there is incorrect information registered

If you try to log in to NAVER in a situation that the service URL or Callback URL information is not correctly entered or omitted in the application information registered in the Developer Center, or if it does not match the registered information, the following errors may occur during the login process. Therefore, you must conduct a test to confirm that the registered information is applied in the correct way before its application to the service.



Cannot log in to naverIdOAuthLoginTest.

Cannot log in with this NAVERr ID as there is an error in the settings of the naverIdOAuthLoginTest service. If the same problem continues, please contact the administrator of the naverIdOAuthLoginTest service.

Are you the administrator of the naverIdOAuthLoginTest? You can find out a solution.

Terms of Service/ **Privacy Policy**/ Disclaimers and legal notices/ Customer Service Center for member information

NAVER Copyright © NAVER Corp. All Rights Reserved.

3.3.2 Creating a URL for linking with NAVER Login

In order to link with NAVER Login, you must first create a URL for 'NAVER Login' that will be moved when clicking the NAVER Login button.
In that process, users can implement the authentication for NAVER Login and agree to the linkage with NAVER login.
When users agree to the linkage with login, their consent information is sent to the Callback URL.

Information about the request URL

Method	Request URL	Output format	Description

GET / POST	https://nid.naver.com/oauth2.0/authorize	URL redirect	Requesting authentication for NAVER Login

Information on variables requested

Name of variables requested	Type	Required or not	Default value	Description
response_type	string	Y	code	A value for the internal division of the authentication process and must be sent as a 'code'.
client_id	string	Y	-	Client ID issued at the time of registering applications
redirect_uri	string	Y	-	Callback URL entered at the time of registering applications, which is applied with URL encoding
state	string	Y	-	State token value generated by the application to prevent cross-site request forgery attacks from occurring, which is applied with URL encoding

Sample of requests

```
https://nid.naver.com/oauth2.0/authorize?response_type=code&client_id=CLIENT_ID&state=STATE_STRING&redirect_uri=CALLBACK_URL
```

3.3.3 Information about the Callback by linking with NAVER Login

If users are not logged in to NAVER when calling the API requesting authentication for NAVER Login, the NAVER Login screen appears. However, if users are logged in to NAVER, a screen to confirm consent to the provision of basic information appears.

When the required process for NAVER login and consent to the provision of basic information, the

code and state values are sent to the callback URL as a URL string. The code value is used in a request to issue access tokens.
When a request to the API fails, an error code and error message are sent.

Information about the response from the Callback

- When a request to the API is successful: http://callback URL/redirect?code={code value}&state={state value}
- When a request to the API fails: http://callback URL/redirect?state={state value}&error={error code value }&error_description={error message}

Field	Type	Description
code	string	An authentication code that is returned when authentication for NAVER Login is successful. Used to issue an access token.
state	string	State tokens generated by the application to prevent cross-site request forgery attacks from occurring, which is applied with URL encoding
error	string	An error code returned when authentication for NAVER Login fails
error_description	string	An error message returned when authentication for NAVER Login fails

3.3.4 Request for the issuance of an access token

You can get an access token using the information sent by the Callback. An access token is authentication information, ensuring that users have completed their authentication.
If you use this access token, you can call either a profile API or open API.

You can get information about an access token with a response from the API when you call the 'API that issues the access token' using the code value' sent by the Callback.
You can call the API using that 'code value' only once. Once the access token is issued, the used 'code value' can't be reused.

Information about the request URL

Method	Request URL	Output format	Description
GET / POST	https://nid.naver.com/oauth2.0/token	json	Request the issuance of an access token

Information on variables requested

Name of variables requested	Type	Required or not	Default value	Description
grant_type	string	Y	-	A value for the classification of the authentication process 1) Issue:'authorization_code' 2) Refresh:'refresh_token' 3) Delete: 'delete'
client_id	string	Y	-	Client ID issued at the time of registering applications
client_secret	string	Y	-	Client Secret issued at the time of registering applications
code	string	Required for “issue”	-	Authorization code returned after successful calls to APIs requesting authentication for login
state	string	Required for “issue”	-	State tokens generated by the application to prevent cross-site request forgery attacks from occurring, which is applied with URL encoding
refresh_token	string	Required for “refresh”	-	Refresh tokens issued after successful user authentication on NAVER

access_token	string	Required for “delete”	-	Previously issued access tokens using a value with URL encoding applied.
service_provider	string	Required for “delete”	'NAVER'	Sending the name of the authentication provider set to 'NAVER'

Sample of requests

https://nid.naver.com/oauth2.0/token?grant_type=authorization_code&client_id=jyvqXeaVOVmV&client_secret=527300A0_COq1_XV33cf&code=EIc5bFrl4RibFls1&state=9kgsGTfH4j7IyAkg

Response information

Field	Type	Description
access_token	string	The period of the validity of access tokens expires when the time (seconds) set in the expires_in parameter has elapsed after it has been issued.
refresh_token	string	The refresh token is used when the access token is required to be reissued when the period of its validity has expired.
token_type	string	Two types of access tokens, including Bearer and MAC, are supported.
expires_in	integer	Expiry date of the access token (in seconds)
error	string	Error code
error_description	string	Error message

3.3.5 Calling the profile API using the access token

If you use an access token, you can call either APIs that search profile information or open APIs.
You must first call the API to search profile information for obtaining users' login information.

Information about the request URL

Method	Authentication	Request URL	Output format	Description
GET / POST	OAuth2.0	https://openapi.naver.com/v1/nid/me	JSON	Searching profile information

Information on variables requested

There is no separate variable requested, and when making calls with the request URL, all you have to do is send the access token value in the request header as shown below.

Request header

Request header name	Description
Authorization	A header that delivers an access token that is included in the header value in the following format. The token type is set as "Bearer." Authorization: {token type] {access token]

Sample of requests

```
curl  -XGET "https://openapi.naver.com/v1/nid/me" ₩
     -H "Authorization: Bearer
AAAAPluf0L+qfDkMABQ3lJ8heq2mlw71DojBj3oc2Z6OxMQESVSrtR0dbvsiQbPbP1/cxva23n7mQShtfK
4pchdk/rc="
```

Output results

Field	Type	Required or not	Description
resultcode	String	Y	Result code for API calls

message	String	Y	Message of call results
response/id	String	Y	Information to identify the same person Information to identify the same person is issued for each NAVER ID
response/nickname	String	Y	User nickname
response/name	String	Y	User name
response/email	String	Y	User’s e-mail address
response/gender	String	Y	Gender - F: Female - M: Male - U: Unidentifiable
response/age	String	Y	User's age group
response/birthday	String	Y	User's birthday (MM-DD format)
response/profile_image	String	Y	URL to the user's profile picture
response/birthyear	String	Y	Year of birth
response/mobile	String	Y	Cell phone number

3.3.6 Using an access token to check the permissions on a profile that users allow

The access token can be used to check the profile items that are allowed to be provided by users. If a specific profile item of users is considered essential for the operation of the service, it is recommended that you check the items that can be provided before searching the profile if necessary.

Information about the request URL

Method	Authentication	Request URL	Output format	Description

GET / POST	OAuth2.0	https://openapi.naver.com/v1/nid/verify	JSON	Verification of and confirming permissions on access tokens

Information on variables requested

Name of variables requested	Type	Required or not	Default value	Description
info	boolean	N	false	Response to information of setting the permissions, if it is true

Request header

Request header name	Description
Authorization	A header that delivers an access token that is included in the header value in the following format. The token type is fixed as "Bearer." Authorization: {token type] {access token]

Sample of requests

```
curl  -XGET "https://openapi.naver.com/v1/nid/me" ₩
    -H "Authorization: Bearer
AAAAPIuf0L+qfDkMABQ3IJ8heq2mlw71DojBj3oc2Z6OxMQESVSrtR0dbvsiQbPbP1/cxva23n7mQShtfK
4pchdk/rc="
```

Output results Required or not

Field	Type	Required or not	Description
resultcode	String	Y	Result code for API calls

message	String	Y	Message of call results
response/token	String	Y	Access token
response/expire_date	String	Y	Expiry time of the access token
response/allowed_profile	String	Y	Profile items allowed (comma-separated)

3.4 Linking NAVER Login with OpenID Connect (OIDC)

3.4.1 Prior to the development of linkage between NAVER Login and OpenID Connect (OIDC)

This feature is provided separately from the existing OAuth 2.0 API, as an independent function. Please note that while it is similar to the existing API, it is offered through a different path, and although the request parameters and response values are similar, some aspects are different. When using OpenID Connect (OIDC) to additionally receive an id_token, you must use a different API from the one used previously. From this point onward, OpenID Connect will be referred to as OIDC.

3.4.2 Retrieving OIDC configuration information

An API used to access the API and metadata provided by OIDC. If you are using a client side framework that implements OIDC, you can easily apply OIDC by setting this URL.

Information about the request URL

Method	Request URL	Output format	Description
GET	https://nid.naver.com/.well-known/openid-configuration	JSON	OIDC meta information

Sample of requests

https://nid.naver.com/.well-known/openid-configuration

3.4.3 Issuing a jwk key

An API for issuing the key used for id_token generation and signature verification. If no jwk key for OIDC exists during the initialization process, a request can be made to configure the key.

Information about the requested URL

Method	Request URL	Output format	Description
GET / POST	https://nid.naver.com/oauth2/jwks	json	Requesting Signing keys

Sample of requests

https://nid.naver.com/oauth2/jwks

3.4.4. Creating a URL for linking with NAVER Login

The content is identical to that of section 3.4.2.

Information about the request URL

Method	Request URL	Output format	Description
GET / POST	https://nid.naver.com/oauth2/authorize	URL redirect	Requesting authentication for NAVER Login

Information on variables requested

Name of variables requested	Type	Required or not	Default value	Description
response_type	string	Y	code	A value for the internal division of the authentication process must be sent as a 'code'

Name of variables requested	Type	Required or not	Default value	Description
client_id	string	Y	-	Client ID issued at the time of registering applications
redirect_uri	string	Y	-	Callback URL entered at the time of registering applications, which is applied with URL encoding
state	string	Y	-	State token value generated by the application to prevent cross-site request forgery attacks from occurring, which is applied with URL encoding
scope	string	Y	-	'openid' scope required
code_challenge	string	N	-	Hashed PKCE value
code_challenge_method	string	N	S256	PKCE algorithm

Sample of requests

https://nid.naver.com/oauth2/authorize?response_type=code&client_id=CLIENT_ID&state=STATE_STRING&redirect_uri=CALLBACK_URL&scope=openid&code_challenge=CODE_CHALLENGE&code_challenge_method=S256

3.4.5 Information about the Callback by linking with NAVER Login

If users are not logged in to NAVER when calling the API requesting authentication for NAVER Login, the NAVER Login screen appears. However, if users are logged in to NAVER, a screen to confirm consent to the provision of basic information appears.

When the required process for NAVER login and consent to the provision of basic information is

complete, the code and state values are sent to the callback URL as a URL string. The code value is used in a request to issue access tokens.
When a request to the API fails, an error code and error message are sent.

Callback

Information about the response from the callback

- When a request to the API is successful: http://callbackURL/redirect?code={code value}&state={state value}

- When a request to the API fails: http:/callback URL/redirect?state={state value}&error={errorcode value}&error_description={error message}

Field	Type	Description
code	string	An authentication code that is returned when authentication for NAVER Login is successful. Used to issue an access token.
state	string	State tokens generated by the application to prevent cross-site request forgery attacks from occurring, which is applied with URL encoding
error	string	An error code returned when authentication for NAVER Login fails
error_description	string	An error message returned when authentication for NAVER Login fails

3.4.6 Request for the issuance of an access token

You can get an access token using the information sent by the Callback. An access token is authentication information, ensuring that users have completed their authentication.
If you use this access token, you can call either a profile API or open API.

You can get information about an access token with a response from the API when you call the 'API that issues the access token' using the code value' sent by the Callback.
You can call the API using that 'code value' only once. Once the access token is issued, the used 'code value' can't be reused.

Information about the request URL

Method	Request URL	Output format	Description
POST	https://nid.naver.com/oauth2/token	json	Request the issuance of an access token

Information on variables requested

Name of variables requested	Type	Required or not	Default value	Description
grant_type	string	Y	-	A value for the classification of the authentication process 1) Issue:'authorization_code' 2) Refresh:'refresh_token' 3) Delete: 'delete'
client_id	string	Y	-	Client ID issued at the time of registering applications
client_secret	string	Y	-	Client Secret issued at the time of registering applications
code	string	Required for "issue"	-	Authorization code returned after successful calls to APIs requesting authentication for login
state	string	Required for "issue"	-	State tokens generated by the application to prevent cross-site request forgery attacks from occurring, which is applied with URL encoding
refresh_token	string	Required for "issue	-	Refresh tokens issued after successful user authentication on NAVER
access_token	string	Required for "delete"	-	Previously issued access tokens using a value with URL encoding applied.

Name of variables requested	Type	Required or not	Default value	Description
service_provider	string	Required for “delete”	'NAVER'	Sending the name of the authentication provider set to 'NAVER'
code_verifier	string	N	-	Added when operating with PKCE

Sample of requests

https://nid.naver.com/oauth2/token?grant_type=authorization_code&client_id=jyvqXeaVOVmV&client_secret=527300A0_COq1_XV33cf&code=EIc5bFrl4RibFls1&state=9kgsGTfH4j7IyAkg&code_verifier=CODE_VERIFIER

Response information

Field	Type	Description
access_token	string	The period of the validity of access tokens expires when the time (seconds) set in the expires_in parameter has elapsed after it has been issued.
refresh_token	string	The refresh token is used when the access token is required to be reissued when the period of its validity has expired.
token_type	string	Two types of access tokens, including Bearer and MAC, are supported.
expires_in	integer	Expiry date of the access token (in seconds)
id_token	integer	Id token, used for user authentication

Field	Type	Description
error	string	Error code
error_description	string	Error message

4. Profile renewal and re-authentication of users using NAVER Login

4.1 Profile renewal of users using NAVER Login

4.1.1 Regarding the access token

You can get access and refresh tokens through APIs that issue the access tokens.
The format of the access tokens is as follows.

API's response to the access tokens

```
{
 "access_token": "Access Token,”
 "refresh_token": "Refresh Token,”
 "token_type": "Access Token Type (Bearer),”
 "expires_in": "Expiry time (Seconds) "
}
```

Access token specification

The access token specification is as follows.

- access_token: A string of 256 characters or less with a combination of Alphabet uppercase and lowercase letters, numbers, and special characters (+/=)
- refresh_token: A string of 256 characters or less with a combination of Alphabet uppercase and lowercase letters and numbers

- expires_in: Valid from the time the numbers are issued until expires_in (seconds)

Purpose of using an access token

The access token is used as an authentication value of users when calling the API to search a user profile or using the login OpenAPI provided by NAVER.

Method of using an access token

When calling the API using an access token, the access token is included in the request header as follows.

- Request header name: Authorization
- Format of the request header: TOKEN_TYPE ACCESS_TOKEN

Sample of a response header including an access token

```
Authorization: Bearer ACCESS_TOKEN
```

Sample of API calls using an access token

```
curl -XGET "https://openapi.naver.com/v1/nid/me" ₩
    -H "Authorization: Bearer ACCESS_TOKEN"

GET /v1/nid/me HTTP/1.1
Host: openapi.naver.com
User-Agent: curl/7.43.0
Accept: */*
Authorization: Bearer ACCESS_TOKEN
```

4.1.2 Regarding the refresh token

The access token is only valid until expires_in (seconds) from the time it is issued through the API that issues the access token.
In other words, the token can only be used within expires_in (3600 seconds / 1 hour by default) after it has been issued. After that, it can no longer be used.

When the access token has expired, a refresh token issued along with the access token may be used to reissue the valid access token.
Therefore, the refresh token must be stored separately in the database for the expected expiry of the access token and then can be used as circumstances demand.

The method of using the refresh token to reissue an access token is as follows.

Information about the request URL

Method	Request URL	Output format	Description
GET / POST	https://nid.naver.com/oauth2.0/token	JSON	Request for reissuing the access token using the refresh token

Information on variables requested

Name of variables requested	Type	Required or not	Default value	Description
client_id	string	Y	-	Client ID issued at the time of registering applications
client_secret	string	Y	-	Client Secret issued at the time of registering applications
refresh_token	string	Y	-	Refresh token issued through the API that issues the access token
grant_type	string	Y	-	Request type is set to the refresh_token.

Sample of requests

```
https://nid.naver.com/oauth2.0/token?grant_type=refresh_token&client_id=CLIENT_ID&client_secret
=CLIENT_SECRET&refresh_token=REFRESH_TOKEN
```

Output results

Field	Type	Required or not	Description
access_token	String	Y	Reissued access token
token_type	String	Y	Token type (bearer)
expires_in	String	Y	Message of the result of checking the validity of the access token

4.1.3 Expiration and renewal cycle of access tokens. Renewal of profile information

The access token may become invalid depending on its expiry date or actions such as its renewal, deletion, etc.
Invalid access tokens are not qualified to search profile information or call login OpenAPI.
Therefore, if the access token is invalid, a valid access token can be issued either by reissuing a valid access token using a renewal token or by performing authentication for NAVER Login once again.

The following methods may be used to determine the validity of access tokens.

- The access token can be considered valid if the response is normally sent when calling the API that searches profile information.
- It can be determined whether the access token is currently valid by calling an API that checks the validity of the access token.

The API that checks the validity of the access token can be used in the following way.

Information about the request URL

Method	Authentication	Request URL	Output format	Description
GET / POST	OAuth2.0	https://openapi.naver.com/v1/nid/me	JSON	Checking the validity of the access token

Information on variables requested

When you make a call with the request URL without a separate variable being requested, all you have to do is send the access token value in the request header as shown below.

Request header

Request header name	Description
Authorization	A header that delivers an access token that is included in the header value in the following format. The token type is fixed as "Bearer." Authorization: {token type] {access token]

Sample of requests

```
curl  -XGET "https://openapi.naver.com/v1/nid/verify" ₩
   -H "Authorization: Bearer
AAAAPluf0L+qfDkMABQ3lJ8heq2mlw71DojBj3oc2Z6OxMQESVSrtR0dbvsiQbPbP1/cxva23n7mQShtfK
4pchdk/rc="
```

Output results

Field	Type	Required or not	Description
resultcode	String	Y	Result code for API calls
message	String	Y	Message of the result of checking the validity of the access token

4.1.4 Updating profiles

Some of the profile information of users using NAVER login may be changed depending on whether their information is changed.
The following information is available to be changed.

- Name
- Nickname

- Profile image
- E-mail address
- Birthday
- Gender
- Age group
- Date of birth
- Cell phone number

When user information is changed, NAVER does not separately notify the relevant service of the change.
Therefore, it is recommended that you search and refresh the profile information periodically or whenever a user login occurs.

4.1.5 In the event of obtaining consent from users again for the permissions to the profile that the users have rejected

Users can choose *not to provide* certain profile items in the process of agreeing to the initial linkage with NAVER Login. In this case, information on profile items that have been rejected to be provided can't be obtained by searching the profile.
If profile items rejected to be provided are required for the use of the service, it is possible to ask users to provide consent again so that they can once again choose to agree to it.

The specification of APIs for re-consent to NAVER Login is as follows.

Information about the request URL

Method	Request URL	Output format	Description
GET / POST	https://nid.naver.com/oauth2.0/authorize	URL redirect	Requesting authentication for NAVER Login

Information on variables requested

Name of variables requested	Type	Required or not	Default value	Description

response_type	string	Y	code	A value for the internal division of the authentication process and must be sent as a 'code'.
client_id	string	Y	-	Client ID issued at the time of registering applications
redirect_uri	string	Y	-	Callback URL entered at the time of registering applications, which is applied with URL encoding
state	string	Y	-	State token value generated by the application to prevent cross-site request forgery attacks from occurring, which is applied with URL encoding
auth_type	string	Y	-	'Reprompt' that must be sent when making a request for re-consent

Sample of requests

```
https://nid.naver.com/oauth2.0/authorize?response_type=code&client_id=CLIENT_ID&state=STATE_STRING&redirect_uri=CALLBACK_URL&auth_type=reprompt
```

As the above action means asking users for confirmation once again if they reject the relevant permissions, it is recommended that you ask them to go through the authentication process after you have sufficiently explained to them *why the items are necessary* for the service.

4.2 Reauthentication

4.2.1 If the reauthentication of users is required

When trying to increase the level of account security by requiring users to go through the authentication procedures again even if the access token is valid, it is possible to perform authentication by NAVER users through re-authentication for NAVER login.

In general, users may be required to go through authentication procedures in the following situations.

- When searching user's personal information or accessing the change page
- When users are going to cancel or withdraw from the service
- When verification by users is required due to the suspicious theft of user accounts

Re-authentication for NAVER Login is performed as follows.

1. Request re-authentication for NAVER Login
2. Request that users go through the NAVER login procedures, regardless of their current login status
3. Enter ID/PW
4. Authentication completed

The specification of APIs on reauthentication for NAVER Login is as follows.

Information about the request URL

Method	Request URL	Output format	Description
GET / POST	https://nid.naver.com/oauth2.0/authorize	URL redirect	Requesting authentication for NAVER Login

Information on variables requested

Name of variables requested	Type	Required or not	Default value	Description
response_type	string	Y	code	A value for the internal division of the authentication process and must be sent as a 'code'.
client_id	string	Y	-	Client ID issued at the time of registering applications

redirect_uri	string	Y	-	Callback URL entered at the time of registering applications, which is applied with URL encoding
state	string	Y	-	State token value generated by the application to prevent cross-site request forgery attacks from occurring, which is applied with URL encoding
auth_type	string	Y	-	'Reauthenticate' that must be sent when making a request for reauthentication

Sample of requests

https://nid.naver.com/oauth2.0/authorize?response_type=code&client_id=CLIENT_ID&state=STATE_STRING&redirect_uri=CALLBACK_URL&auth_type=reauthenticate

The subsequent use is the same as in the event of authentication for NAVER Login.

4.3 Canceling the linkage with NAVER Login

4.3.1 When the linkage with NAVER Login is required to be canceled,

When users no longer use the service (withdraw from the service) or no longer use the linkage with NAVER Login (cancellation of the linkage),
they can break the connection through the API that cancels the linkage with NAVER Login.

If the linkage with NAVER Login is successfully canceled through the relevant API, the change is applied as follows.

- The previously issued access and refresh tokens expire as soon as the API is called. (Access and refresh tokens can no longer be used.)
- The items are removed from the list of the linkage with NAVER Login in "My Information > Security Settings > Management of Connections to External Websites" on NAVER.

- Even after the linkage with the NAVER Login is canceled, users can perform the linkage operation again, but they must newly submit their consent during that process.

The linkage with NAVER Login can be canceled as follows.

Information about the request URL

Method	Request URL	Output format	Description
GET / POST	https://nid.naver.com/oauth2.0/token	JSON	Request to cancel the linkage using the access token

Information on variables requested

Name of variables requested	Type	Required or not	Default value	Description
client_id	string	Y	-	Client ID issued at the time of registering applications
client_secret	string	Y	-	Client Secrets issued at the time of registering applications
access_token	string	Y	-	A valid access token
grant_type	string	Y	-	A request type. It is set to delete.

Sample of requests

```
https://nid.naver.com/oauth2.0/token?grant_type=delete&client_id=CLIENT_ID&client_secret=CLIENT_SECRET&access_token=ACCESS_TOKEN
```

Output results

Field	Type	Required or not	Description

access_token	String	Y	Deleted access token
result	String	Y	Processing result (success)

Important!

In the event of an access token used for an API that cancels the linkage with NAVER Login, a valid access token must be used. (The linkage can't be canceled with an expired token or a non-existent token.)
Therefore, it is recommended that you check the validity of the access token and to refresh the access token in accordance with the renewal process of the access token specified in the Item 5.1.

5. Additional features of NAVER Login

5.1 Automatic login of the service in NAVER App

5.1.1 What is the service's automatic login feature?

This is a feature that allows the accounts of users who have used the service through NAVER Login to be automatically logged in to the service for their convenient use when they want to access the service from the NAVER App.

The user's login process can be simplified for their convenient use in the following situations.

- When accessing the service via a search on the NAVER App
- When accessing the service through Favorites on the NAVER App
- When accessing the service through a link delivered by TalkTalk, e-mail, etc.

Automatic login is performed by the following procedure.

1. Check the environment where automatic login is possible.
2. Deal with the URL linked to NAVER Login
3. Perform the linkage operation or handle errors in the callback page
4. Login completed

5.1.2 Restrictions

This feature can be used only when accessing the web page of the service on the "NAVER App."

5.1.3 Conditions for determining the NAVER App

NAVER App has a specific type of User-Agent. So, it can be identified through the User-Agent header of the request header.

Conditions for determining the NAVER App

Check whether the following strings are included in User-Agent.

```
NAVER(inapp; search;
```

Sample of User-Agent on the NAVER App

```
Mozilla/5.0 (iPhone; CPU iPhone OS like Mac OS X) AppleWebKit/605.1.15 Naver(inapp; search; 620; 10.10.2; XR)
```

5.1.4 Specifications for automatically logging in to the service

If the conditions specified in the Item 5.1.3 are met, the service makes users check the authentication page through 302 redirect, javascript location replace, etc.

Information about the request URL

Method	Request URL	Output format	Description
GET / POST	https://nid.naver.com/oauth2.0/authorize	URL redirect	Requesting authentication for NAVER Login

Information on variables requested

Name of variables requested	Type	Required or not	Default value	Description

response_type	string	Y	code	A value for the internal division of the authentication process and must be sent as a 'code'.
client_id	string	Y	-	Client ID issued at the time of registering applications
redirect_uri	string	Y	-	Callback URL entered at the time of registering applications, which is applied with URL encoding
state	string	Y	-	State token value generated by the application to prevent cross-site request forgery attacks from occurring, which is applied with URL encoding
auth_type	string	Y	-	'Autologin' that must be sent when making a request for autologin

Sample of requests

https://nid.naver.com/oauth2.0/authorize?response_type=code&client_id=CLIENT_ID&state=STATE_STRING&redirect_uri=CALLBACK_URL&auth_type=autologin

If the request is processed normally, the page is redirected with the processing result by recirect_uri (callback url). If it is normally processed as a target to be automatically logged in, it must be processed in the same way as in the event of callback for the linkage with NAVER Login. (Login after the issuance of access tokens)

5.1.5 Definition of error conditions and error codes

If it is not possible to proceed with automatic login normally, the corresponding error code is transmitted as a parameter to the callback page.

Parameter	Type	Required or not	Description

error	string	Y	Error codes
error_description	string	Y	Details of error codes

Definitions of error codes and messages

error parameter	error_description parameter	Description
access_denied	user not logged in.	Users are not logged in to NAVER.
access_denied	need user consent.	Users are not linked with the service.
access_denied	unsupported browser environment.	Users have accessed the service in an environment other than the NAVER App.

Remedies to errors

If error codes regarding the failure of automatic login are delivered by the callback, it must be processed so that users can log in through the login button in the same way as before.

5.2 Agree to the Terms of Service by proxy

5.2.1 What is the feature that allows agreeing to the Terms of Service by proxy?

NAVER Login provides users with a feature that performs the "Consent to the Terms of Service" procedure, which is essential for using the service, by proxy in addition to linking users with login. It can provide users with a complicated consent process easily and conveniently through NAVER Login, increasing their convenience.

5.2.2 Checklists before the linkage with consent to the Terms of Service by proxy

- Fully understand the development guide to check the necessary items for the linkage with NAVER Login in advance.
- Register the application through the NAVER Developer Center.
- Register the information necessary for consent to the Terms of Service through the menu of "Information on the Terms of Service" in the Developer Center.

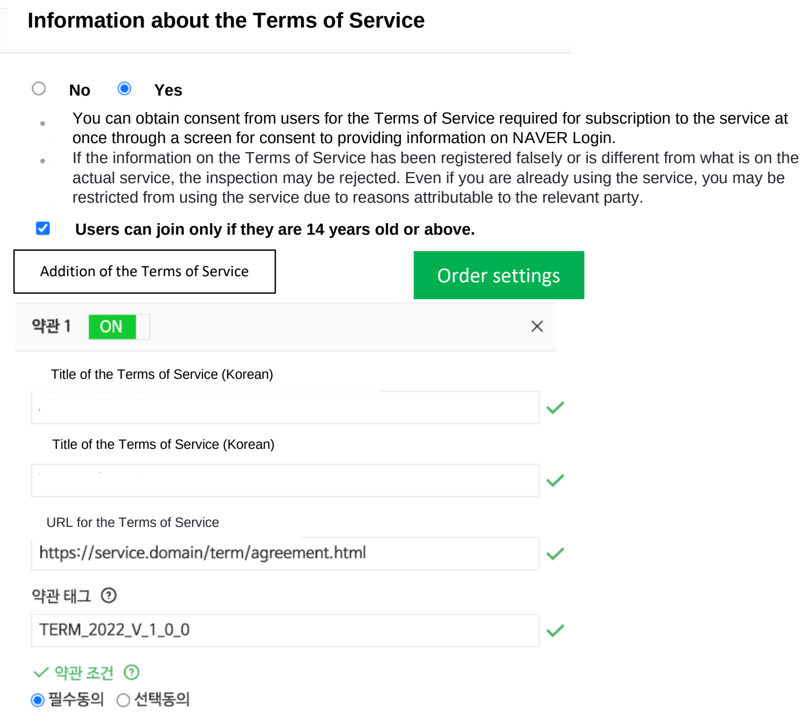
- If the information on the Terms of Service is registered falsely or it is different from those of the actual service, the inspection may be rejected. Even if you are already using the service, you may be restricted from using the service due to reasons attributable to the relevant party.

5.2.3 Main setting information

- Users can join only if they are 14 years old or above.
  - If the service is not provided to users under the age of 14, it is possible to set it up so that only those over the age of 14 can join.
  - If it is set to 'only users over the age of 14 can join', those who are not can't join the service by checking the age information of NAVER users, and users who do not have age information will be shown to items for consent so that they can join the service after clicking [I am over the age of 14] in the consent process.
- Multiple pieces of information can be registered as information about the Terms of Service, and the required items for registration are as follows.
  - Title of the Terms of Service (Korean): Title of the Terms of Service to be displayed on the screen for consent to providing information during the linkage process
  - Title of the Terms of Service (English): Title of the Terms of Service to be displayed on the screen for consent to providing information during the linkage process
  - URL for the Terms of Service: URL on the web page where users can check the details of the Terms of Service
  - Tag for the Terms of Service: Items necessary for the classification of the Terms of Service
  - Required/optional: In the event of 'required consent', if users do not agree, they can't join the service. In the event of 'optional consent', they can join the service whether or not they have consented.

5.2.4 Example of a screen to set up consent to the Terms of Service by proxy

Login open API service environment



Information about the Terms of Service

○ **No** ● **Yes**

- You can obtain consent from users for the Terms of Service required for subscription to the service at once through a screen for consent to providing information on NAVER Login.
- If the information on the Terms of Service has been registered falsely or is different from what is on the actual service, the inspection may be rejected. Even if you are already using the service, you may be restricted from using the service due to reasons attributable to the relevant party.

☑ **Users can join only if they are 14 years old or above.**

Title of the Terms of Service (Korean)

✓

Title of the Terms of Service (Korean)

✓

URL for the Terms of Service

https://service.domain/term/agreement.html ✓

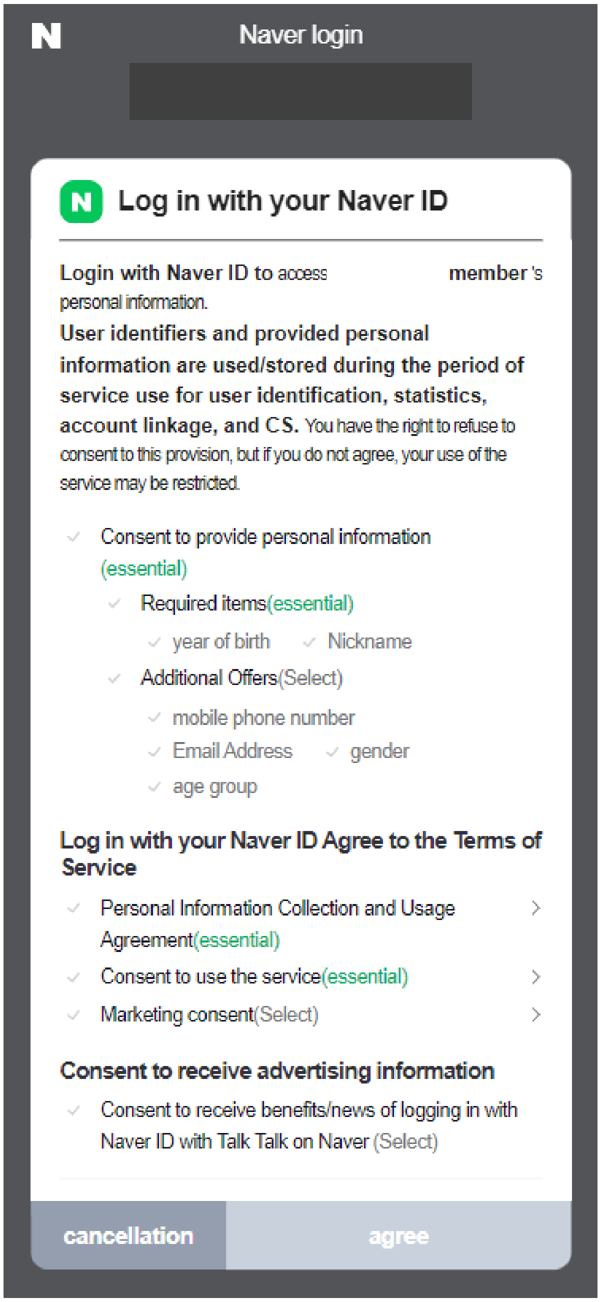
약관 태그 ⓘ

TERM_2022_V_1_0_0 ✓

✓ 약관 조건 ⓘ

● 필수동의 ○ 선택동의

5.2.5 Example of a screen to use consent to the Terms of Service by proxy



Log in with your Naver ID

Login with Naver ID to access **member**'s personal information.

User identifiers and provided personal information are used/stored during the period of service use for user identification, statistics, account linkage, and CS. You have the right to refuse to consent to this provision, but if you do not agree, your use of the service may be restricted.

- ✓ Consent to provide personal information (essential)
  - ✓ Required items(essential)
    - ✓ year of birth ✓ Nickname
  - ✓ Additional Offers(Select)
    - ✓ mobile phone number
    - ✓ Email Address ✓ gender
    - ✓ age group

Log in with your Naver ID Agree to the Terms of Service

- ✓ Personal Information Collection and Usage Agreement(essential) >
- ✓ Consent to use the service(essential) >
- ✓ Marketing consent(Select) >

Consent to receive advertising information

- ✓ Consent to receive benefits/news of logging in with Naver ID with Talk Talk on Naver (Select)

5.2.6 Confirmation of whether users agree to 'consent to the Terms of Service by proxy'

You can search information about the consent status of users, linked to NAVER Login, to the Terms of Service. This feature can be provided only for applications in which consent to the Terms of Service by proxy is set.

Information about the request URL

Method	Authentication	Request URL	Output format	Description
GET / POST	OAuth2.0	https://openapi.naver.com/v1/nid/agreement	JSON	Verification of access tokens and confirmation of consent information for the Terms of Service

Request header

Request header name	Description
Authorization	A header that delivers an access token that is included in the header value in the following format. The token type is fixed as "Bearer." Authorization: {token type] {access token]

Sample of requests

```
curl  -XGET "https://openapi.naver.com/v1/nid/agreement" ₩
    -H "Authorization: Bearer
AAAAPIuf0L+qfDkMABQ3IJ8heq2m...dbvsiQbPbP1/cxva23n7mQShtfK4pchdk/rc="
```

Sample of requests

```
> GET /v1/nid/agreement HTTP/2
> Host: openapi.naver.com
> user-agent: curl/7.79.1
> accept: */*
> authorization: Bearer AAAAPIuf0L+qfDkM...ShtfK4pchdk/rc=

< HTTP/2 200
< date: Tue, 07 Dec 2021 09:52:56 GMT
< content-type: application/json;charset=utf-8
```

Output results

Field	Type	Required or not	Description
result	String	Y	API processing results (success or failure)
accessToken	String	Y	Information about access tokens entered by a header
agreementInfos	List<Object>	Y	List of information about consent to the Terms of Service
agreementInfos[].termCode	String	Y	the Terms of Service code
agreementInfos[].clientId	String	Y	ClientID (Client ID and application identifier)
agreementInfos[]. agreeDate	DateTime	Y	Date of consent (HH:MI:SS.sss AM MM/DD/YYYY)

Note: Please refer to the item below for the details of JSON SCHEMA.

Output results *JSON SCHEMA*

```
{
   "$schema": "http://json-schema.org/draft-07/schema,”
   "type": "object,”
   "description": "The root schema,”
   "properties": {
      "result": {
         "type": "string,”
         "description": "API processing results (success / failure)"
      },
      "accessToken": {
         "type": "string,”
         "description": " Information about access tokens entered by a header "
      },
      "agreementInfos": {
         "type": "array,”
         "description": " List of information about consent to the Terms of Service ,”
         "items": {
           "type": "object,”
           "properties": {
             "termCode": {
               "type": "string,”
               "description": " Terms of Service code (See Appendix 1)"
             },
             "clientId": {
               "type": "string,”
               "description": "ClientID (Client ID and application identifier)"
             },
             "agreeDate": {
               "type": "string,”
               "description": " Date of consent to the Terms of Service "
             }
           }
         }
      }
   }
}
```

Output results *JSON EXAMPLE*

```
{
  "result": "success,”
  "accessToken": "{input accessToken},”
  "agreementInfos": [{
      "termCode": "{TERMCODE},”
      "clientId": "{CLIENTID},”
      "agreeDate": "HH:MI:SS.sss AM MM/DD/YYYY"
  }, {
      "termCode": "{TERMCODE},”
      "clientId": "{CLIENTID},”
      "agreeDate": "HH:MI:SS.sss PM MM/DD/YYYY"
  }]
}
```

5.3 Marketing message consent feature through Talk Talk sync

5.3.1 What is NAVER Talk Talk sync?

When you connect the application login with NAVER Talk Talk account, you can gain consent for Talk Talk notification (marketing message consent) in the same page as the login consent and can send service and marketing messages from Talk Talk Partner center to users that have consented. Use the Login-Talk Talk sync feature to increase your marketing customer base and send messages!

How to send marketing messages in Talk Talk Partner center >

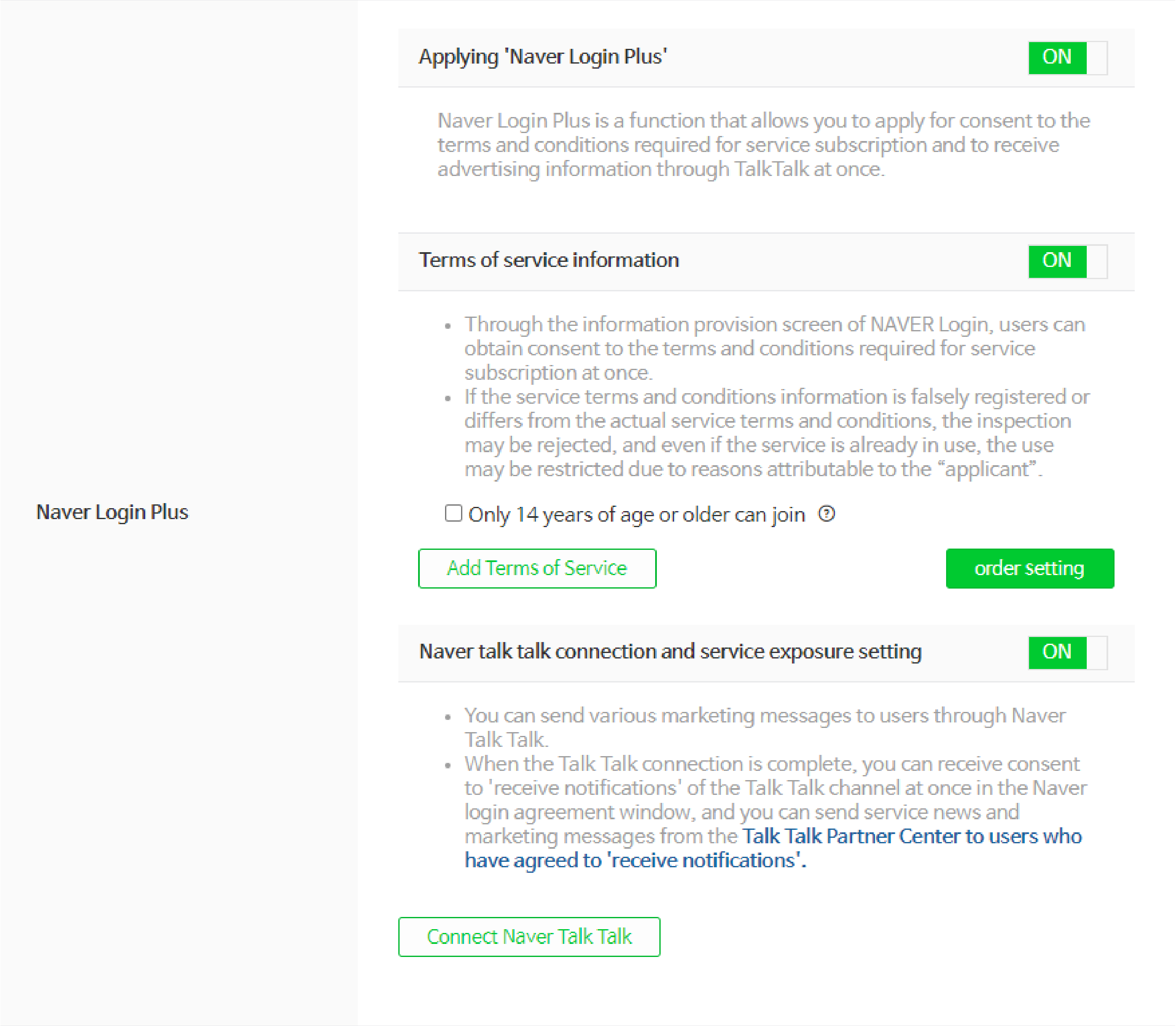
5.3.2 Things to do before Talk Talk sync

- Familiarize yourself with the development guide and the review items that are needed in NAVER login sync.
- Register your application through the NAVER Developers.
- Create a Talk Talk account using NAVER Talk Talk Partner center. If you have a pre-existing Talk Talk account, confirm the account details to prepare for the sync.

5.3.3 Main settings

You can sync login application and Talk Talk account in NAVER Developers and Talk Talk Partner center. You only have to complete the sync process in one for sync information to be displayed.

1) Sync Talk Talk account in NAVER Developers



Naver Login Plus

Applying 'Naver Login Plus' ON

Naver Login Plus is a function that allows you to apply for consent to the terms and conditions required for service subscription and to receive advertising information through TalkTalk at once.

Terms of service information ON

- Through the information provision screen of NAVER Login, users can obtain consent to the terms and conditions required for service subscription at once.
- If the service terms and conditions information is falsely registered or differs from the actual service terms and conditions, the inspection may be rejected, and even if the service is already in use, the use may be restricted due to reasons attributable to the "applicant".

□ Only 14 years of age or older can join ⓘ

Add Terms of Service

order setting

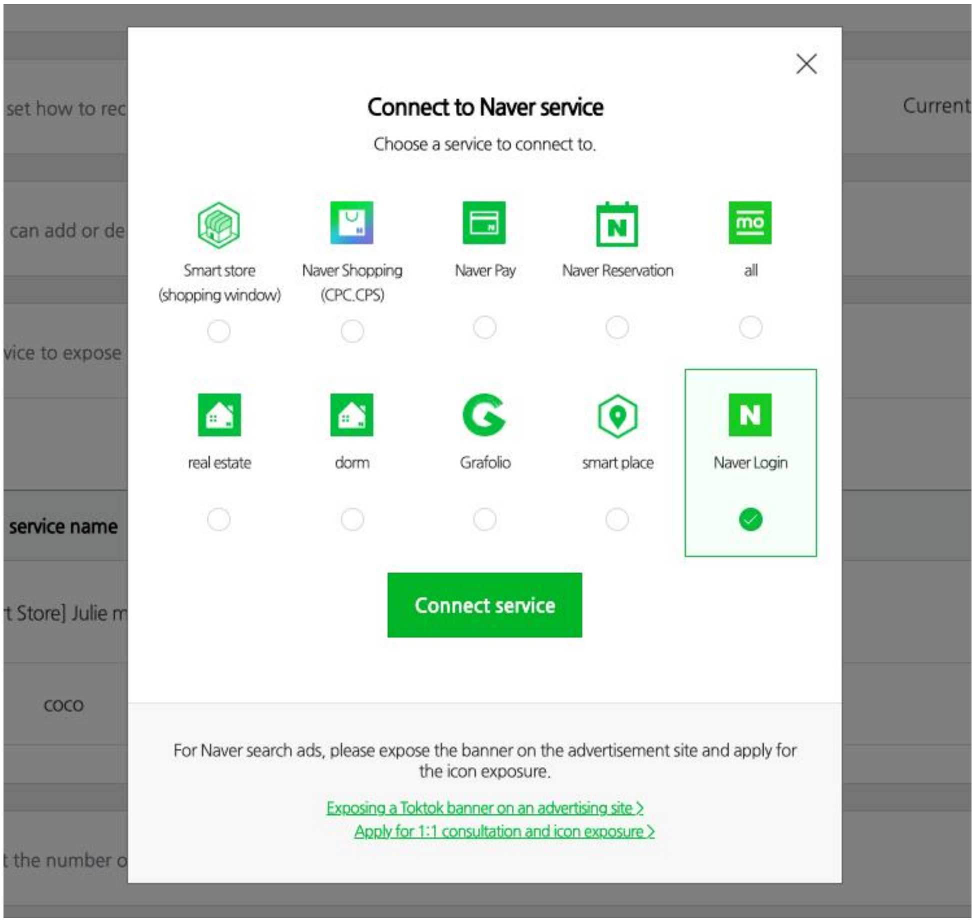
Naver talk talk connection and service exposure setting ON

- You can send various marketing messages to users through Naver Talk Talk.
- When the Talk Talk connection is complete, you can receive consent to 'receive notifications' of the Talk Talk channel at once in the Naver login agreement window, and you can send service news and marketing messages from the **Talk Talk Partner Center to users who have agreed to 'receive notifications'.**

Connect Naver Talk Talk

1. API Settings > NAVER Login Plus > Click NAVER Talk Talk sync in NAVER Talk Talk sync and service display setting area
2. Select the Talk Talk account in the Talk Talk simple sync pop-up.
3. The NAVER login application and Talk Talk account sync has been completed.
4. View the Talk Talk account information.

2) Sync login application in Talk Talk Partner center



Connect to Naver service

Choose a service to connect to.

○ Smart store (shopping window)
○ Naver Shopping (CPC.CPS)
○ Naver Pay
○ Naver Reservation
○ all
○ real estate
○ dorm
○ Grafolio
○ smart place
● Naver Login

For Naver search ads, please expose the banner on the advertisement site and apply for the icon exposure.

Exposing a Toktok banner on an advertising site >

Apply for 1:1 consultation and icon exposure >

1. Connect to Talk Talk Partner center > Settings > Service sync menu.
2. Click service sync button.
3. Select NAVER login and press next.
4. Select login application to connect and click add.
5. NAVER login application and Talk Talk account sync has been completed.

6. Check synced login application information.

5.3.4 Additional Function for “Receive Notifications” in SMART STORE

You can obtain consent for "Receive Notifications" from the SMART STORE you are operating all at once through the login consent window. If your TalkTalk account is a business type, you can set up a SMART STORE with the same business registration number as your TalkTalk account. Here are the steps for setting up a SMART STORE.

- 5.3.3 1) After selecting your TalkTalk account in the "Connect TalkTalk Account" section in NAVER Developers, if there is a SMART STORE that meets the requirements, a screen will appear, allowing you to choose a SMART STORE. On this screen, select the SMART STORE for which you wish to obtain integrated consent in the login consent window, and then click the “Confirm” button to complete the setup.

- 5.3.3 2) After selecting "NAVER Login" in the "Connect Login Application" section in the TalkTalk Partner Center, if there is a SMART STORE that meets the requirements, the area where you can select the SMART STORE will be activated. On this screen, choose the SMART STORE for which you wish to obtain integrated consent in the login consent window, and then click the "Next" button to complete the setup.

If you have already connected the login application to your TalkTalk account, you can select a SMART STORE that meets the requirements by repeating the connection process in section 5.3.3 after disconnecting your TalkTalk account. Even if you disconnect your TalkTalk account, customers who previously consented to "Receive Notifications" will remain connected.