



GIG Mobile App Integration Registration Guide





Table of Content

Revision History	3
Introduction:	4
1. Endpoint: Get Token(GIG)	5
2. Endpoint: Get Token(Notch)	6
3. Endpoint: Check User	7
Request Details	7
Response Details	8
4. Endpoint: Registration	11
Request Details	11
Response Details	12



Revision History

Name	Date	Description
Mohamed Kamal	23 July 2025	Initiation
Mohamed El Abacy	27 July 2025	Review



Introduction:

This document outlines the technical integration between the mobile application developed by Notchnco and the middleware solution implemented by GIG, which serves as an interface to GIG's internal CRM system. The objective of this integration is to enable seamless, secure, and efficient data exchange related to policy creation, endorsements, and claims management.

The mobile application will provide a user-friendly interface for end-users to initiate and manage insurance-related transactions. These transactions will be communicated to GIG's CRM system through the middleware layer, which will handle validation, routing, and transformation of data to ensure compatibility and compliance with GIG's backend processes.

This document provides the technical specifications, data flow descriptions, API endpoints, authentication mechanisms, and integration protocols required to establish and maintain a robust communication channel between the two systems. It is intended for software engineers, system integrators, and technical stakeholders involved in the development, deployment, and maintenance of this integration.



1. Endpoint: Get Token(GIG)

Source and destination

- Source: Notchnco
- Destination: GIG Middleware

URL

GIG/identity/api/Authentication/login

Request Method

'POST'

Request Headers

Header	Example	Description
Content-Type	application/JSON	Indicates the media type

Request Parameters

Parameter	Type	Description
grantType	String	A static field should be sent as it is
clientId	String	A static field should be sent as it is
clientSecret	String	A static field should be sent as it is

Prerequisites:

- This request\response is draft as the accurate one will depend on GIG feedback\confirmation
- GIG Should provide Notchnco with grantType, ClientId, Clientsecret

Description:

This request allows Notchnco to securely connect with the GIG Middleware by logging in with provided credentials, so they can start exchanging information.



2. Endpoint: Get Token(Notch)

Source and destination

- Source: GIG Middleware
- Destination: Notch

URL

Notch/identity/api/Authentication/login

Request Method

'POST'

Request Headers

Header	Example	Description
Content-Type	application/JSON	Indicates the media type

Request Parameters

Parameter	Type	Description
grantType	String	A static field should be sent as it is
clientId	String	A static field should be sent as it is
clientSecret	String	A static field should be sent as it is

Prerequisites:

- This request\response is draft as the accurate one will depend on GIG feedback\confirmation
- Notch Should provide Notchnco with grantType, ClientId, Clientsecret

Description:

This request allows GIG Middleware to securely connect with Notch by logging in using fixed credentials to enable data exchange.



3. Endpoint: Check User

Source and Destination

- Source: Notchnco
- Destination: GIG Middleware

Request Details

URL:

Gig/v1/api/user/checkuser

Request Method:

'POST'

Request Headers:

Header	Example	Description
Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cGEiOiJ1cm9kaW50IiwiaWF0IjoxNTE2MzY0MDAifQ..	Bearer token for authentication.
Content-Type	application/JSON	Indicates the media type

Request Parameters:

Parameter	Type	Description
PhoneNumber	String	Mobile number
Email	String	Email address

Request Body Examples:

```
{  
  "phoneNumber": "string",  
  "email": "string"  
}
```



Response Details

Field	Type	Description
Data	object	The data returned by the API, typically empty in this case.
isExisting	boolean	Indicates whether the user already exists.
userInfo	object	User information returned if the user exists.
nuserid	Int	ID will be created from GIG and send to Notchnco accordingly Notchnco can use it in future APIs as mobile app user identifier.
CorrelationId	String	correlation ID
FullNameAr	String	Full Arabic name
FullNameEn	String (200)	Full English name
Birthdate	DATETIME	Birth date
Gender	String	M or F
PhoneNumber	String	Mobile number
Email	String	Email address
address	String	User Address
photo	String	User Photo Url
idNumber	String	ID Number
passportnumber	String	Passport number
status	Int	1 : Accepted 2 : reject
Message	string	A message indicating the success or failure of the operation.
Code	int	The status code indicating the result of the operation. 0 for Success, 1 for Error.

Response Example

Example of a Successful Response (existing user)

```
{
  "Data":{
    "isExisting":true,
```



```
"userInfo":{
  "nuserid":5236,
  "CorrelationId":"notchnco_2f5e9f3e-49e8-4d9b-97d4-18a5a8c9c657",
  "fullNameAr":"string",
  "fullNameEn":"string",
  "birthdate":"1997-03-01T00:00:00",
  "gender":"M",
  "phoneNumber":"string",
  "email":"string",
  "address":"string",
  "photo":"string",
  "idNumber":"28703011811215",
  "passportnumber":"",
  "status":1
},
"Message":"Done",
"Code":0
}
```

Example of a Successful Response (not existing user)

```
{
  "Data":{
    "isExisting":false,
    "userInfo":null,
    "Message":"Done",
    "Code":0
  }
}
```

Examples of a Failed Response

```
{
  "Data": null,
  "Message": "error",
  "Code": 1
}
```

Description:

This API allows Notchnco to verify whether a user exists in the GIG system:

- If the user exists in Notchnco database, then OTP will be sent to mobile number and once verified user will be signed in
- If the user doesn't exist in Notchnco database, then notchnco will call this GIG API to check user existence in GIG, if user exist then we will update notchnco user with received data and send OTP to mobile number and once verified user will be signed in



- If user not exist then we will let client to register using eKYC and in case of success will create account at notchnco with status not confirmed by GIG with added mobile number, email and correlation ID.

4. Endpoint: Registration

Source and Destination

- Source: GIG Middleware
- Destination: Notchnc0

Request Details

URL:

GIG/v1/api/user/register

Request Method:

'POST'

Request Headers:

Header	Example	Description
Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cGE6YW50IjoibnR5cGE6YW50In0=	Bearer token for authentication.
Content-Type	application/JSON	Indicates the media type

Request Parameters:

Parameter	Type	Description
nuserid	Int	ID will be created from GIG and send to Notchnco accordingly Notchnco can use it in future APIs as mobile app user identifier.
CorrelationId	String	correlation ID based on
FullNameAr	String	Full Arabic name
FullNameEn	String (200)	Full English name
Birthdate	DATETIME	Birth date
Gender	String	M or F
PhoneNumber	String	Mobile number
Email	String	Email address
address	String	User Address
photo	String	User Photo Url
idNumber	String	ID Number
passportnumber	String	Passport number
status	Int	1 : Accepted 2 : reject

Request Body Examples:

```
{
  "userid":5236,
```



```
"correlationId": " notchnco_ 9f5e9f3e-49e8-4d9b-97d4-18a5a8c9c657",
"fullNameAr": "string",
"fullNameEn": "string",
"birthdate": "1997-03-01T00:00:00",
"gender": "M",
"phoneNumber": "string",
"email": "string",
"address": "string",
"photo": "string",
"phoneNumber": "28703011811215",
"passportnumber": "",
"status": 1
}
```

Response Details

Field	Type	Description
Data	object	The data returned by the API, typically empty in this case.
Message	string	A message indicating the success or failure of the operation.
Code	int	The status code indicating the result of the operation. 0 for Success, 1 for Error.

Response Example

Example of a Successful Response

```
{
  "Data": null,
  "Message": "Done",
  "Code": 0
}
```

Examples of a Failed Response

```
{
  "Data": null,
  "Message": "error",
  "Code": 1
}
```

Description:

- This API allows GIG to call Notchnco to confirm user registration accordingly Notchnco mark this user as confirmed by GIG and update his\her data.
- Client will be notified by SMS and Email that his\her account has been verified by GIG



Thank You