Guardian

Guardian Connect (Developer Portal)

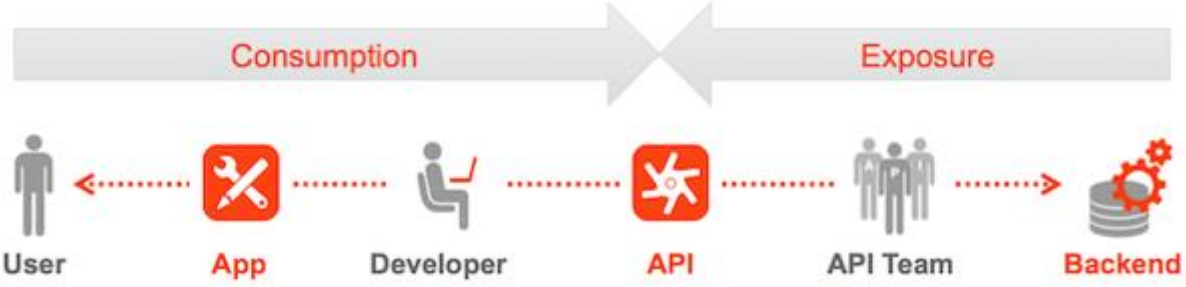
Version 2 last updated March 25, 2021

Guardian

Table of Contents

Guardian Connect

Guardian Connect is an API Developer portal, which is a user friendly, self-service portal that enables API owners to publish their API documentation for application developers. Application developers from external partners can leverage Guardian APIs within their applications they are planning to build for their customers. Application developers can browse through the APIs published on the portal, if the APIs meet their requirements, request access to try them in test environment.



Registration

Developers on the partner side to get access to APIs, the person(administrator) who is responsible to manage carrier connections, should email their company name, administrator first name, last name and contact email address to API_Admin@glic.com. Once we received required details, an account will be created in Guardian Connect and email will be shared to partners to register in Guardian Connect. If you do not receive the email, please check your spam folder. If still not found contact API_Admin@glic.com.

Sample Email:

From: noreply-guardian-connect@glic.com <noreply-guardian-connect@glic.com>
Sent: Saturday, March 7, 2020 8:40 AM
To: <PARTNER EMAIL ADDRESS>
Subject: Welcome to Guardian Connect

Dear <Partner Name>,

Guardian

Welcome to Guardian Connect!

To activate your new Guardian Connect Account, you will need to verify a few pieces of information along with your one time use registration code:

Registration Code : 61bb569b-359a-475e-8417-658e1c8e4228

For account verification and activation please navigate to the following link:

Click here to continue your PartnerAdmin Registration

Happy Building!

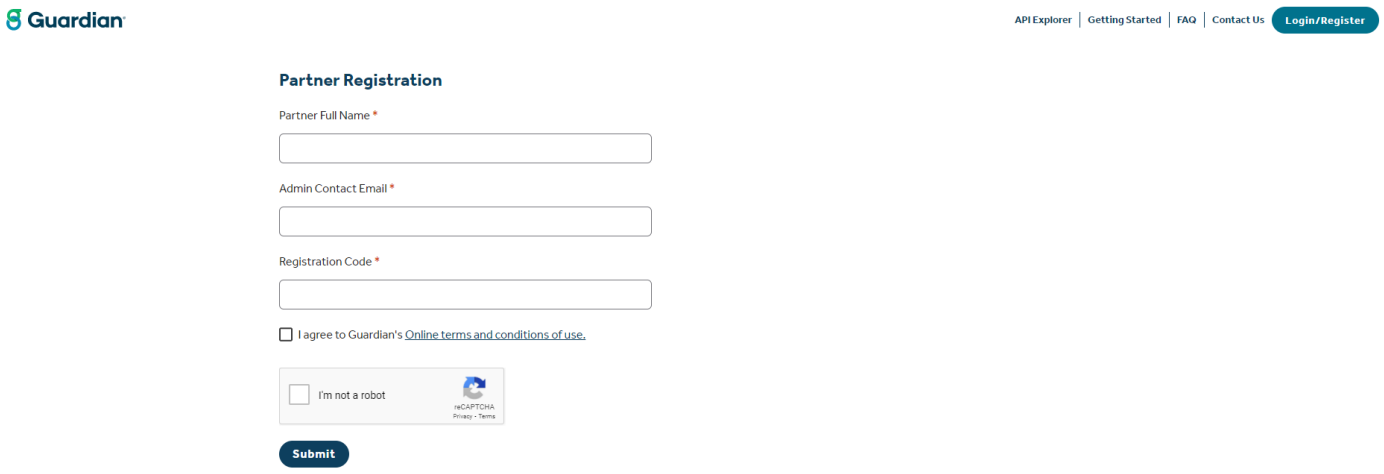
Sincerely,

The Guardian Connect Team

**********PLEASE DO NOT REPLY TO THIS EMAIL**********

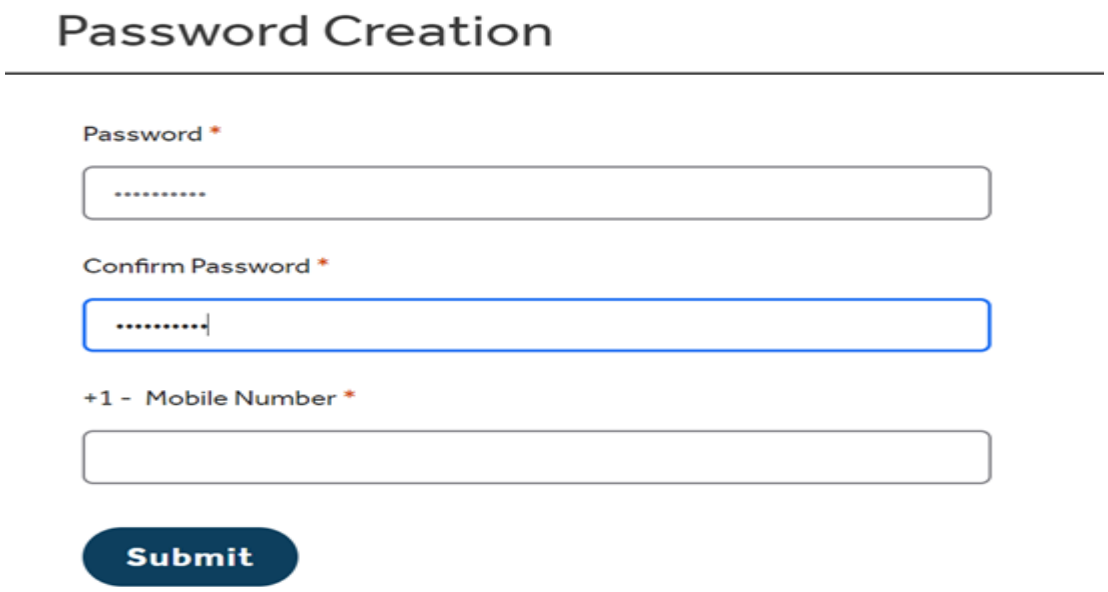
Clicking the above link will route to Guardian Connect registration page. Follow instruction from below table to fill the registration details.

Field Name	
Partner Full Name	Enter Partner full Company's name
Admin Contact Email	Enter the email address to be used to contact to validate for security issues
Registration code	Enter code received from email



Once above details are validated, you will have option to create password as below. After entering password and mobile number, click submit. Once account is created in Guardian Connect, the system will route to the sign in page.

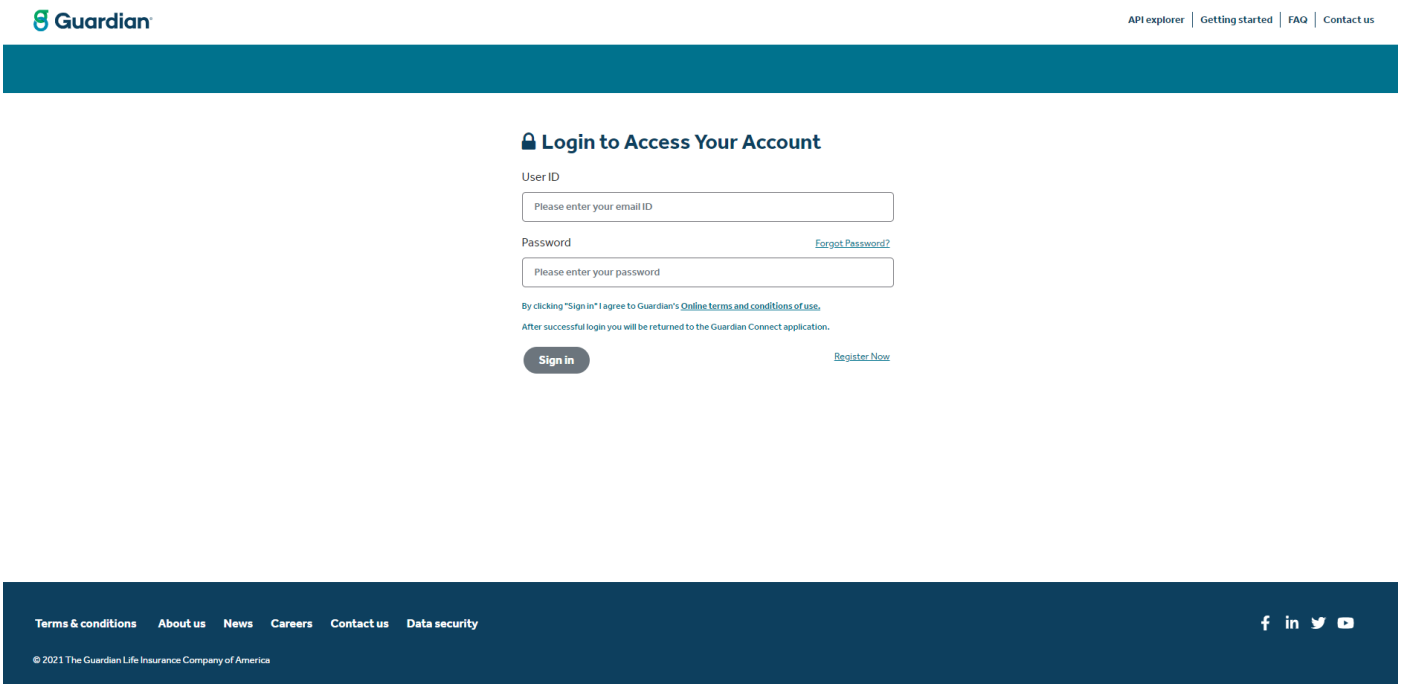
Field Name	
Password	Enter Password (Password requirements are listed on the page)
Confirm Password	Re-Enter Password
+1 – Mobile Number	Enter number with no dashes starting with area code i.e. 2321243234



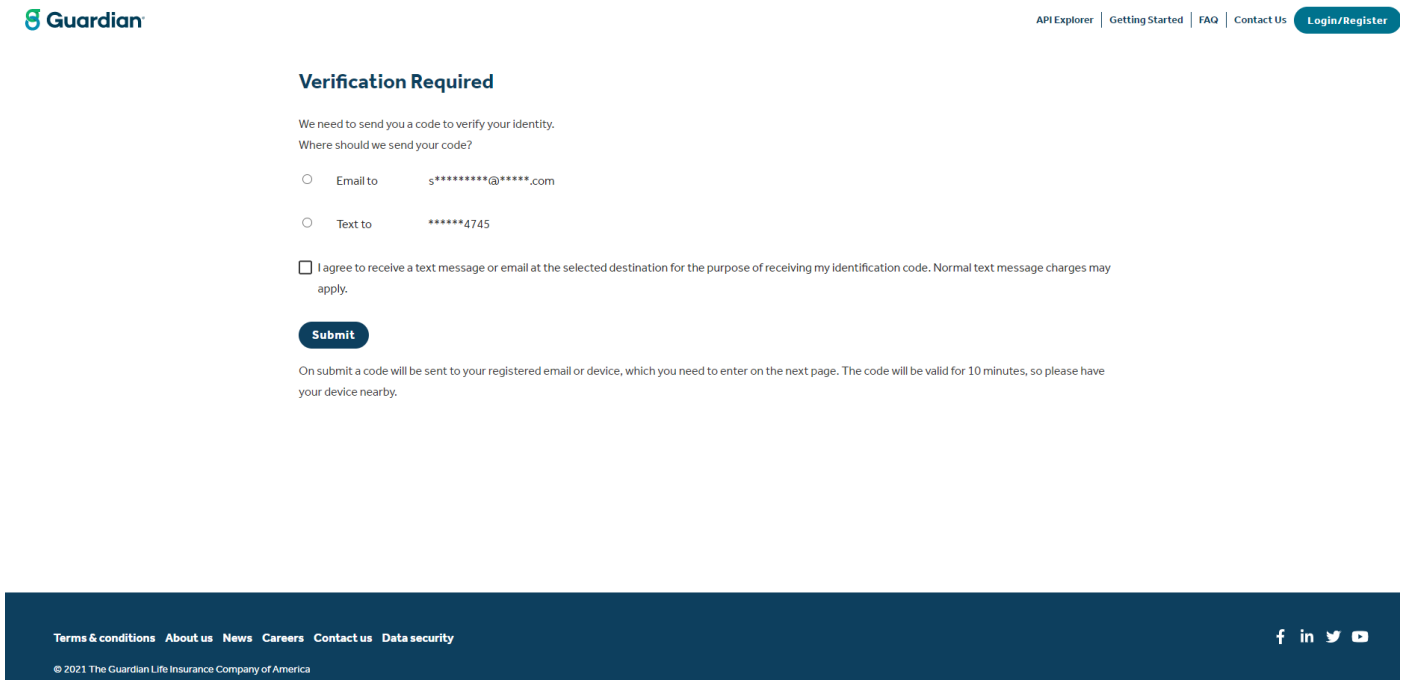
Login to Guardian Connect

Field Name	
UserID	This is the email address used in the first step / Registration
Password	

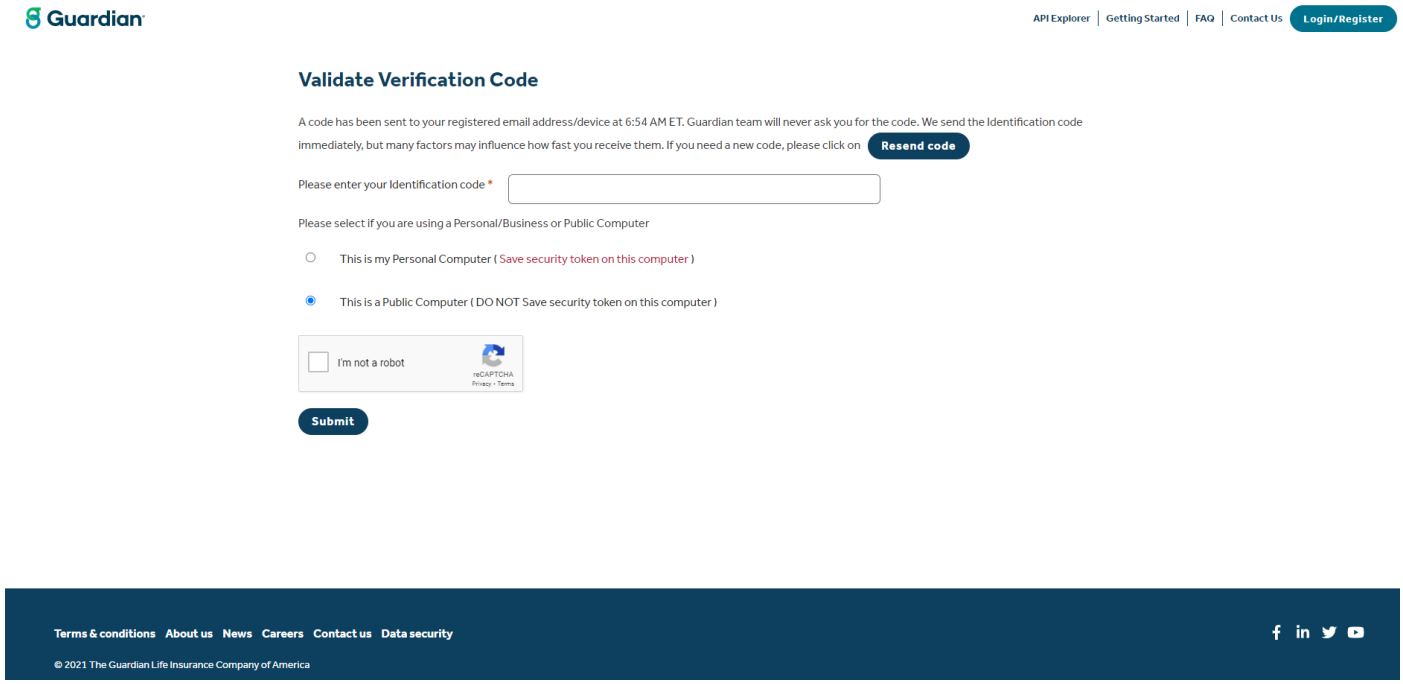
Guardian


Guardian
API explorer | Getting started | FAQ | Contact us
Login to Access Your Account
User ID
Please enter your email ID
Password
Forgot Password?
Please enter your password
By clicking "Sign in" I agree to Guardian's Online terms and conditions of use.
After successful login you will be returned to the Guardian Connect application.
Sign in
Register Now
Terms & conditions
About us
News
Careers
Contact us
Data security
© 2021 The Guardian Life Insurance Company of America

Field Name	
Email	This prefills with the email address utilized for the Admin Contact Email
Text	This prefills with the mobile number utilized in password creation
	The agree button to allow Guardian to send identify questions
	Select the preference of receiving the security code and then submit

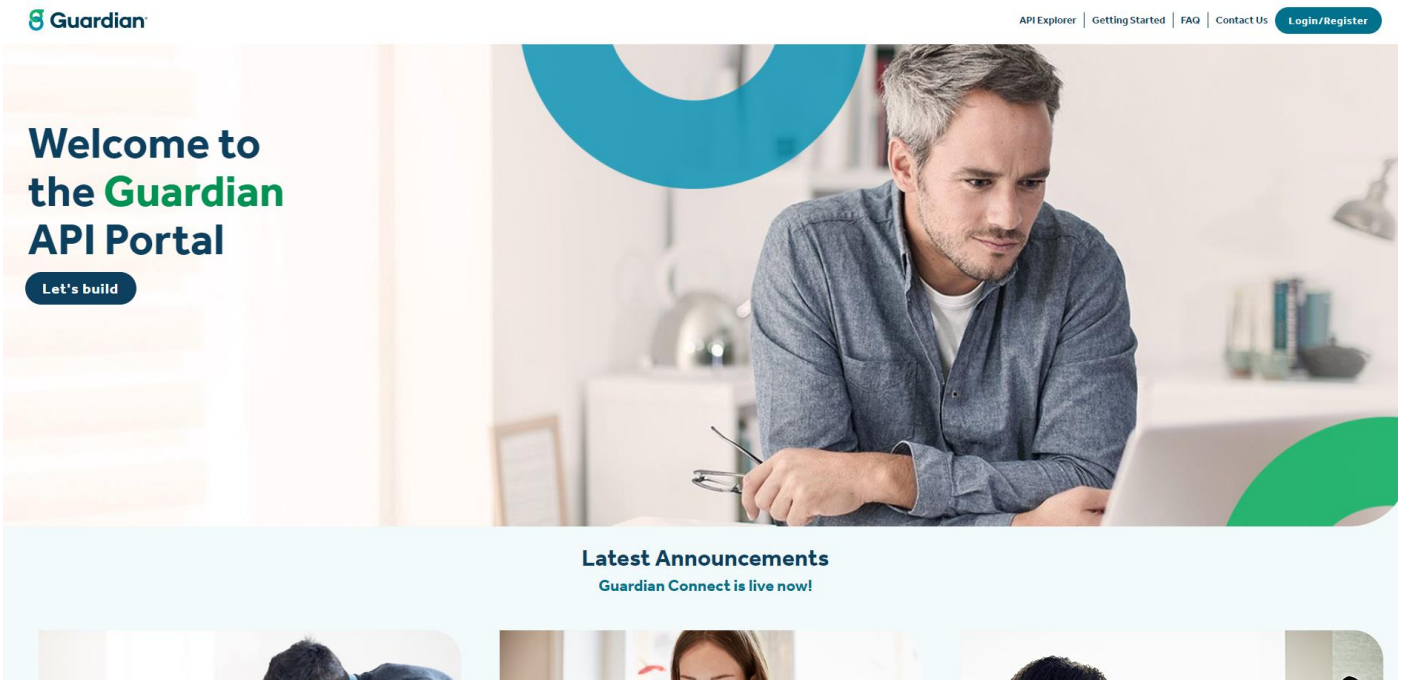


Field Name	
Identification code	Enter the one-time code received in the email
	If public computer is selected identity will need to be verified each time logging in
Code	Enter code in the image



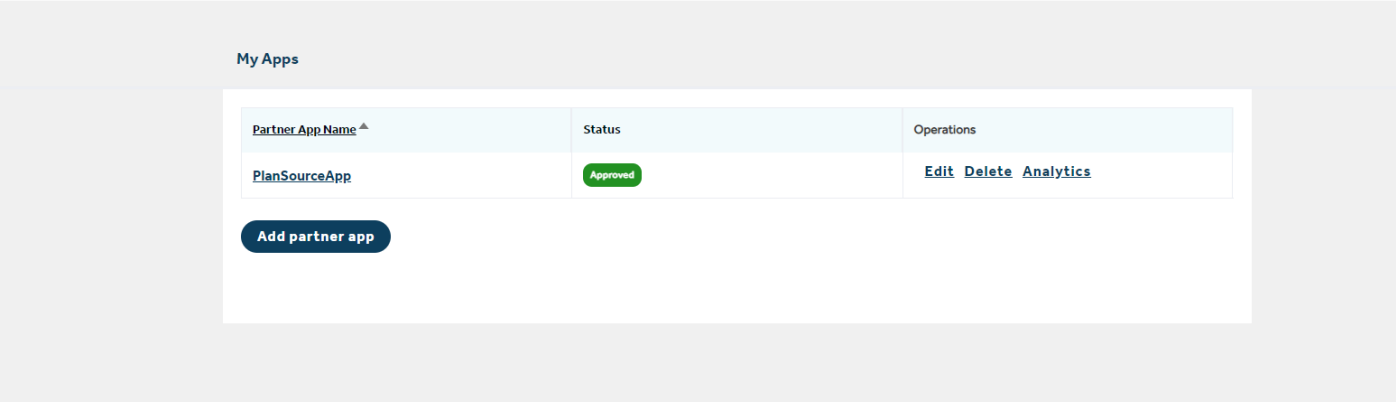
Once Submit is selected and successfully login, you will be routed to the Guardian Connect home page.

Guardian

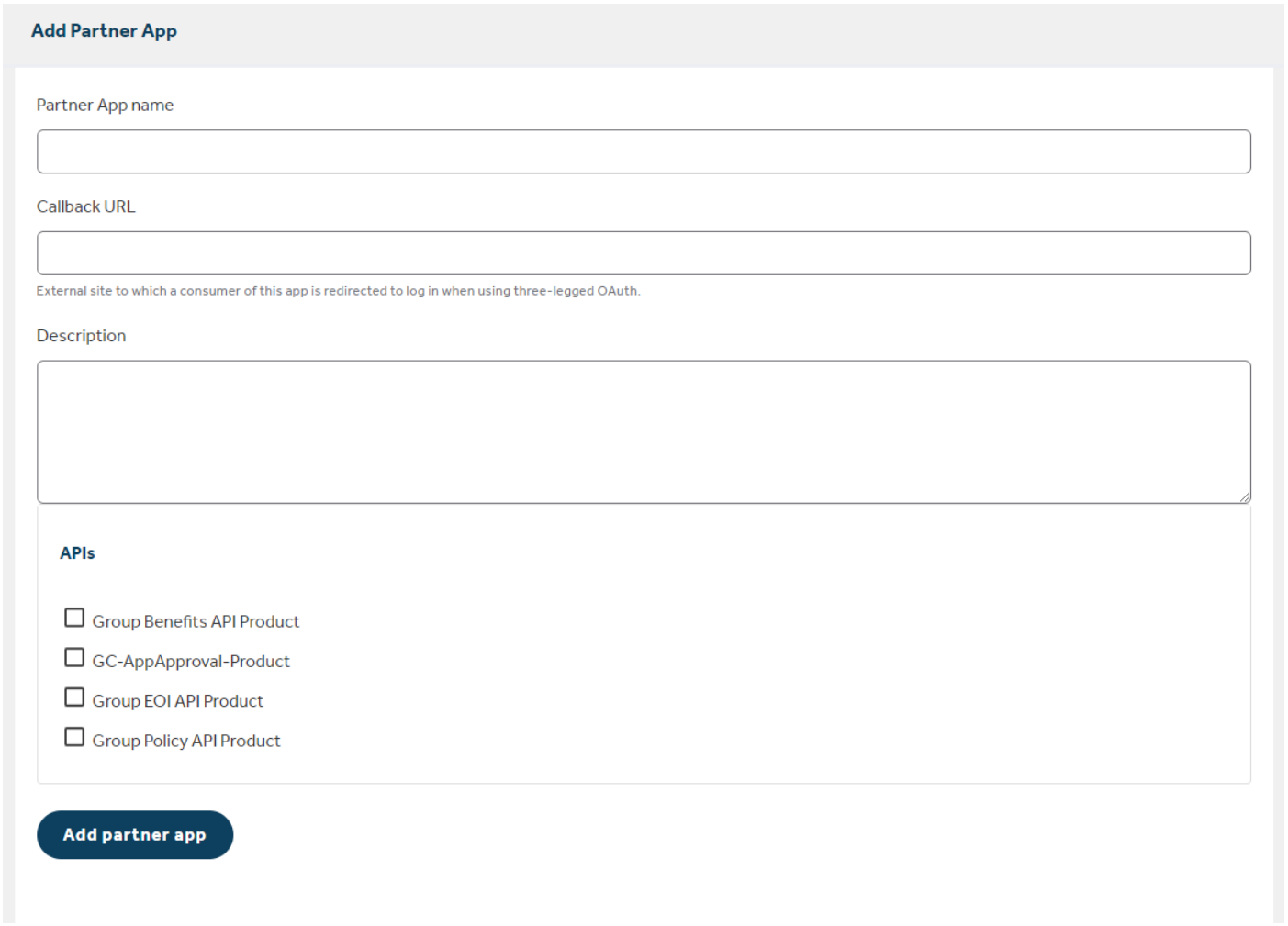

Guardian
API Explorer | Getting Started | FAQ | Contact Us
Login/Register
Welcome to the Guardian API Portal
Let's build
Latest Announcements
Guardian Connect is live now!

Creation of Developer APP

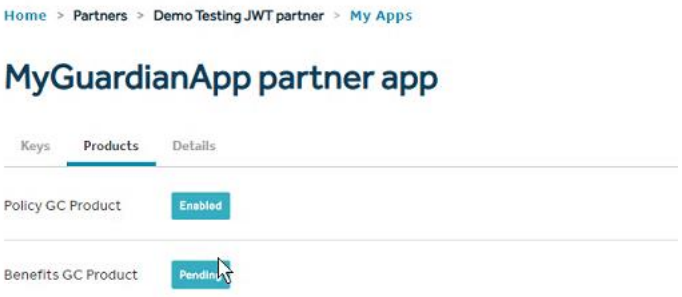
From the home page, select My Apps. Select Register new partner app



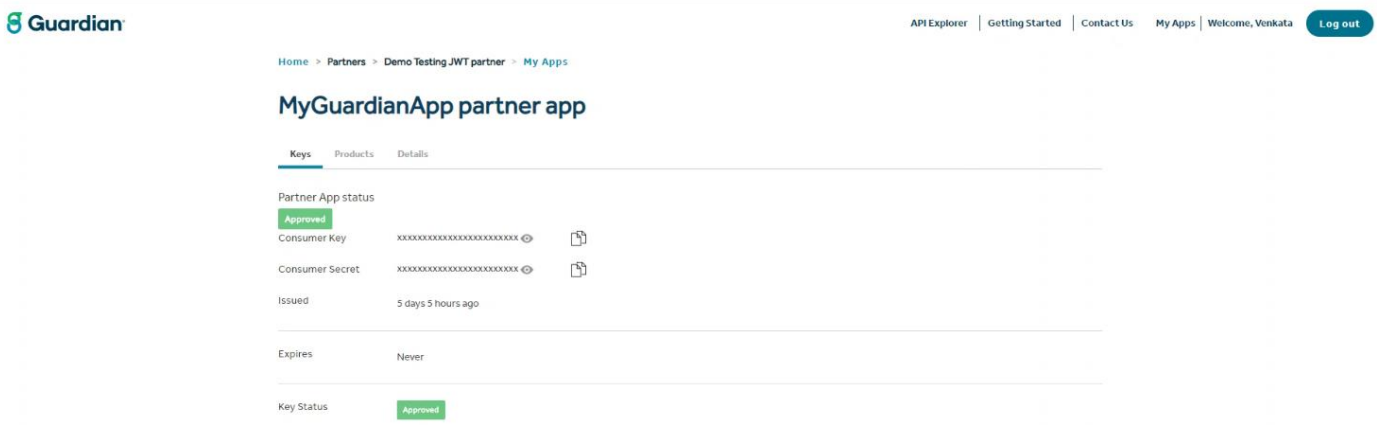
Field Name	
Partner App name	Enter the name you would like to identify the partner app as
Callback URL	Field can be left blank
Description	Freeform to describe the partner app
API's	Search the API's and Select **Group Benefits API Product, Group Policy API Product and Group EOI API Product**



Once app is created, the partner app will appear in MyApps. Under Products your new products will appear. The Blue boxes will state Pended. – (Please send an email to API_Admin@glic.com to approve your products) Once your products have been approved the status will update to Enabled.



Once all steps are completed and app is approved, the consumer key and secret will be active.



The reference ID will be sent in a separate email from Guardian.

Once the consumer key, consumer secret and reference ID have been received, next steps can be started.

Guardian

Security: OAuth 2.0 and JWT

Introduction

A JSON Web Tokens (JWT) is a self-contained, open standard JSON object used to securely exchange information between 2 or more parties. This information is in the form of a set of claims, which are key/value pairs generally used to provide unique attributes about a user or a 3rd party. JWT is commonly implemented in OAuth 2.0 and OpenID Connect (OIDC) type integrations. The following document provides detailed guidelines on usage, generation, consumption, and validation of JWT using Apigee Edge.

JWT as OAuth 2.0 Access Tokens

JSON Web Tokens are used as OAuth 2.0 tokens while requesting access to protected resources, such as APIs. The authorization severs or issuer is responsible to generate the JWT after successfully authenticating and Authorizing a client App. These apps identify themselves using an API key and secret along with some other additional attributes. This token is signed by the issuer/authorization server using a private key or shared key. Signing the token ensure that payload will immediately become invalid if tampered by the client application or user. This ensures that the integrity of the payload is intact. These tokens are used as short lived bearer token, which means the bearer of the token will be able get access to the requested data when they present it to the authorization server, hence it is important to ensure this token is protected and securely stored. A compromised token can be revoked by the authorization server. To provide maximum security, it is important to ensure these tokens have a short life, typically 30mins to an hour.

Guardian

Generate JWT Token

JWT Token Endpoints

Environment	Method	Endpoint	IP whitelisting Needed
PROD	POST	https://api.guardianlife.com/auth/oauth/v2/token/generate	**Yes**

Input parameters

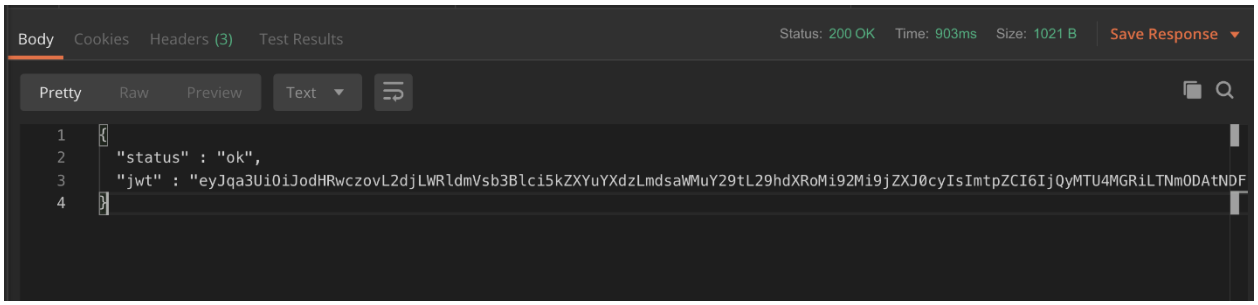
Here are the required inputs to the JWT token endpoint

Type	Attribute / Key	Expected value	Example
query param	grant_type	client_credentials	grant_type=client_credentials
query param	nonce	Unique random alphanumeric string. This string must be unique for each token request. The purpose of this attribute is to prevent replay attacks.	nonce=randomvalue
Header	Authorization	Basic <Base64 encoded (Consumer Key:Consumer Secret)	Basic RUefddfZzMVrfRgfmxUA=='
Body	claims-subject	Reference ID provided as part of partner registration process. {"claims": {"subject": "<reference id>"}}	{"claims": {"subject": "9ef4a793-c9fe-456d-99ae-6a53d866e3e7"}}

Valid Token Endpoint Response

{
"status" : "ok",
"jwt" : "eyJqa3UiOiJodHRwczovL2djLWRldmVsb3Blci5kZXYuYXdzLmdsaWMuY29tL29hdXRoMi92Mi9jZXJ0cyIsImtpZCI6IjQyMTU4MGRiLTNmODAtNDFhZi1iNzU1LWQ1OTEyZjA5NDgxZiIsInR5cCI6IkpXVCIsImFsZyI6IlJTMjU2In0.eyJzdWIiOiI5ZWY0YTc5My1jOWZlLTQ1NmQtOTlhZS02YTUzZDg2NmUzZT

ciLCJhdWQiOiJFRnNqR0xVV0VqOFBydUFoaWRTQWFSUWxxMGQ5QVpEWSIsInZlbmRvcl9pZCI6IjllZjRhNzkzLWM5ZmUtNDU2ZC05OWFlLTZhNTNkODY2ZTNlNyIsImlzcyI6ImFwaXN2Y2F1dGguZGV2LmF3cy5nbGljLmNvbSIsImV4cCI6MTU3MjI3NTIzMiwiaWF0IjoxNTcyMjcxNjMyLCJub25jZSI6ImdkZmdkczEiLCJqdGkiOiJiOTNlYTNiNi03NjU4LTQxYzgtOGQ2YS1jZTQ1NTJmZjY3OGYifQ.U1pHvXUaSEwbJRsZR928aH162lBpFwZf7vz_wtPUrisPWiI27eqMPEwRJUNap2v6uDJmLlBzWFtza_5U-sWu6p7y9KOI1NHoSwiGaS4K1lBk0fyqGSXMxgXqklpeI_EkDR-GcjS0rZ7D5uC1ivHrhVoEiMVB_9WRXKEQYOKrgxNGM7JlnLy8PXxN6BcmEWTAmXeViU9WerjrstUkErH7RqtlhffB4dKPtgamTAtPlh9W3TUJ0OI-xrkfki-sIJlgmA3ReDjidkFn926jRzgA237qnF5pDwjrbXDH3jaNPNVVsW-nvg2OrR5L9Rn6PDbzZUOYVr7iknFMup6vi94wPA"
}



Status Codes

Status codes	Response	Reason
200 OK	Valid JWT response	NA
401 Request missing Authorization Data	{ "error": { "code": 401.01, "message": "Request missing Au } }	Authorization header missing
400 - Bad Request	{ "error": { "code": 400.01, "message": "Missing required fields" }	Any required attribute is missing such as nonce or subject

Status codes	Response	Reason
	} 	

401 - unauthorized	{
	"error": {
	"code": 401.01,
	"message": "Invalid Nonce"
	}
	}
		Same nonce used again. Nonce should be unique for each token request

401 - Invalid Subject or Partner/App not active	{
	"error": {
	"code": 401.01,
	"message": "Unauthorized user"
	}
	}
		Subject is invalid - Subject must be the unique vendor reference id provided by the Guardian Connect API during new partner registration process. All API keys are associated with a given subject and must match for validation to be successful.
		
		App/Partner is invalid or not active - Partner is marked inactive in the system.

Guardian

JWT Verification by Client App

Since we are using JWT as OAuth 2.0 access tokens, the token verification needs to happen first at the client end and later when the client requests access to a resource using a bearer token, the issuer (authorization server) will do token verification again before giving access to the requested resource.

Steps to validate a JWT

Step #	Validation	Expected result (by environments)
	Signature Verification	
1	Decode the JWT. The header and the payload are encoded and separated by "." (period)	
2	Read the header section and look for "jku" and "kid" claims	

Step #	Validation	Expected result (by environments)
	Signature Verification	
3	Verify the url against whitelisted JWKS urls	
4	Download the JWKS from the url provided in the "jku" claim	
5	Read the "n" attribute for the given "kid" claim	

6	Check the algorithm used to sign the payload using the "alg" claim from the header section	
7	Use the "n" value from the JWKS to validate the signature of the JWT payload. Proceed to next step if signature is valid	Integrity of the token is intact
Additional Security Validations (JWT validation beyond signatures)		
8	verify subject (*sub*)	subject must match the subject provided during token request
9	verify nonce	nonce must match the nonce provided during token request

Step #	Validation	Expected result (by environments)
10	verify issuer (*iss*)	Issuer must match the below listed issuers by environment **Environments** / **Issuer** PROD / api.guardianlife.com
11	verify expiry (*exp*)	current time should be prior to the value in the expiry (exp) claim.
12	verify audience (*aud*)	(*aud*) claim will contain API key. (*aud*) must match API key provided during token request

Guardian

Token Expiry

Please note tokens are only valid for 30 minutes from the time of issuance. A new token can be requested after 30 mins with a new nonce.

Verification of JWT by Issuer

Input parameters

attribute	type	Expected value	Example
nonce	query params	The nonce must match the nonce provide during token request	nonce=gdfgds1
Authorization	Headers	The JWT used as a bearer token for verification	Authorization: Bearer 'eyJqa3sdjfksjdfks'

Steps to validate a JWT

Step #	Validation	Expected results (by environments)
	Signature Verification	
1	Decode the JWT. The header and the payload are encoded and separated by "." (period)	
2	Read the header section and look for "*jku*" and "*kid*" claims	

3	Download the JWKS from the url provided in the "*jku*" claim	
4	Read the "*n*" attribute for the given "*kid*" claim	
5	Check the algorithm used to sign the payload using the "*alg*" claim from the header section	
6	Use the "*n*" value from the JWKS to validate the signature of the JWT payload. Proceed to next step if signature is valid	Integrity of the token is intact
Additional Security Validations (JWT validation beyond signatures)		
7	verify nonce	nonce must match the nonce provided during token request
8	verify expiry (*exp*)	Current time should be prior to the value in the expiry (exp) claim.

Status Codes

Status code	Response	Reason
200 OK	Valid response from the backend after successful validation of JWT	

401 Request missing Authorization Data	{
	"error": {
	"code": 401.01,
	"message": "Request missing Authorization Data"
	}
		Bearer token missing

Status code	Response	Reason
	}	

400 Bad Request	{
	"error": {
	"code": 400.01,
	"message": "Missing required attributes"
	}
	}
		Nonce is missing

401 Unauthorized	{
	"error": {
	"code": 401.01,
	"message": "Token expired or invalid"
	}
	}
		Token as either expired or already revoked

401 Invalid Nonce	{
	"error": {
	"code": 401.01,
	"message": "Invalid Nonce"
	}
	}
		Invalid Nonce. Nonce does not match the nonce provided using token request