



Domain Management API – Usage Guide

Document Version	1.1
Last Updated	18 January 2026
Applies to	B2C License Holders
Author	Curaçao Gaming Authority (CGA)

Contents

Domain Management API – Usage Guide	1
1 Purpose and Scope	3
2 Regulatory Background	3
3 Domain Management Framework	3
3.1 3.1 Primary and Secondary Domains	3
3.2 Certificate Display Options.....	4
4 Domain Management Options	4
5 Domains API Overview	4
5.1 Key Capabilities	4
6 Support on the Domain API Module.....	5
7 Base URL & Format	5
8 Authentication & Security	5
8.1 Authentication	5
9 Domain Endpoints	6
9.1 General Domain Information	6
9.2 Domain Creation.....	7
9.3 Domain Active Status	8
9.4 Domain Movement.....	8
9.5 Domain Display Relationships	9
9.6 Domain Deletion	9
10 Applications	9

1 Purpose and Scope

This document provides guidance on the use of the CGA Domains Management API (“Domains API”), which enables licensed B2C operators to manage domain registrations programmatically.

The Domains API provides an alternative to manual domain management via the CGA Portal and supports the full lifecycle of domain registration, verification, structuring, and certificate presentation.

This guide is intended for:

- Licensed B2C operators
- Technical teams and system integrators
- Compliance and regulatory officers

2 Regulatory Background

Under the National Ordinance for Games of Chance (LOK) and applicable CGA licensing conditions, B2C license holders are required to:

- Register all domains used to offer licensed gambling services
- Ensure domains are validated and approved by the CGA
- Operate only on domains linked to a valid Operator Certificate

Once approved, registered domains are linked to the Operator’s Certificate, which is publicly accessible via:

<https://cert.cga.cw>

3 Domain Management Framework

3.1 Primary and Secondary Domains

Operators may operate multiple domains under the same brand.

Under the updated CGA domain framework:

- One or many domains are designated as the Primary (Parent) Domains
- Additional domains can be registered as Secondary Domains under the Primary Domains

- Secondary domains are logically linked to their parent domain

3.2 Certificate Display Options

For each domain, the operator may choose to:

- Display the secondary domains under the parent domain certificate; or
- Issue a separate certificate for the secondary domain

Secondary domain Certificates can also optionally display their primary domain certificate.

This configuration applies equally whether domains are managed via the Portal or the API.

4 Domain Management Options

Method	Description
CGA Portal	Manual management via the Domain Management Module
Domains API	Programmatic domain management via REST API

This document covers Domains API usage only.

For manual operations, refer to the CGA Portal User Manual.

5 Domains API Overview

5.1 Key Capabilities

The Domains API allows operators to:

1. Register new primary or secondary domains
2. Update domain metadata
3. Link or unlink secondary domains
4. Select certificate display preferences
5. Retrieve domain status and verification state
6. Deactivate or remove domains (subject to CGA rules)

All changes made via the API are reflected in the CGA Portal and Operator Certificate.

6 Support on the Domain API Module

For questions or assistance related to the Domain Management API, operators are requested to submit a support ticket via the CGA Portal using the customer account assigned to them by the Curaçao Gaming Authority.

Requests submitted outside the official portal workflow may not be processed.

7 Base URL & Format

- All endpoints are under <https://cert.cga.cw/api>.

For convenience the domain part of the endpoint shall not be repeated in the rest of the document.

- Send JSON with headers:

Content-Type: application/json,

Accept: application/json,

and for protected routes **Authorization: Bearer <token>**.

The **<token>** is sent as a response to the login (authentication) request if it is successful.

Note: Authenticated routes use rate limiting, therefore avoid rapid-fire requests.

8 Authentication & Security

8.1 Authentication

The Domains API uses token-based authentication. API admin credentials are issued by the CGA for its customers once a customer is able to access the portal, it is recommended to create a user account for the API giving it the role of domain manager. However any account can be used to access the API module.

Tokens must be included in the request header

- **POST** <https://cert.cga.cw/api/login>
with body: { "email": "user@example.com", "password": "secret" }.
- On success returns:
- { "token_type": "Bearer", "token": "...", "company_id": ..., "company_name": "..." }.

- Use **Authorization: Bearer <token>** on all protected endpoints.
- **GET** /api/user (with token) returns the details of authenticated user.
- **Response fields:**

id,	User Unique ID
user_name	User name as registered in the portal
email	User's email as registered in the portal
company:	Company data
company_id	Company Unique ID
company_name	Company Name
company_number	Company Registration Number

9 Domain Endpoints

9.1 General Domain Information

- **GET** /api/domains:

Domains for the signed-in user's organization.

- Response Fields:

id	Domain Unique ID
name	Domain name
authorization_model	The type of authorization. Should always be application. Some legacy authorization may have 'sub-licensee'. Please change immediately through the portal to 'application'
authorization_status	Should be 'New', 'In-progress', 'Granted', 'Revoked', 'Withdrawn'
authorization_number	The application/license number
active	If this application is still active
verified	If the domain has been verified by the CGA
parent_id	Null or the primary/parent domain ID
display_relationships	Whether this domain's online certificate should display its relationships to other domains.

- **GET** /api/domains/application:

Domains grouped by application. Same as above but grouped by application.

- **GET** /api/domains/summary:

Same as above but it includes the domain token information; requires auth.

For the user's org; returns domain_name, token, relationship (parent/sub), display_relationships, optional parent_id, org id/name, application number/status, license_status.

- **GET** /api/domain/<id>/token:

Returns token for a domain id. Use the Domain ID in the <id> field

9.2 Domain Creation

- **POST** /api/domains

To create a single or multiple domains. Auth is required for this method. The following fields are required in the body:

- **Single example:**

```
{"application_id": 1, "name": "example.com", "parent": null, "active": true, "display_relationships": true }
```

- **Bulk example:**

```
{"application_id": 1, "names": ["example.com", "sub.example.com"], "parent": null, "active": true, "display_relationships": false }
```

Field Name	Requirement	Type	Description
application_id	Required	Numeric	The Unique Application ID under which this domain is operated. This can be optional from any of 9.1 above.
name	Required	String	The domain name. Notes: protocols are stripped; each domain must be a valid URL and unique). Each

			created domain includes a generated token.
Names	Required	String Array	Domain names to be added
parent	Optional	Numeric	The domain ID of the parent/primary domain if the domain/s is/are secondary.
active	Optional	Boolean	Defaults to true . There is a limit of 100 parent domains and each parent domain can have up to 100 subdomains. Toggling the active status can increase the capacity without the need to delete domains.
display_relationships	Optional	Boolean	Default to true . Indicated whether the online certificate should display its related domains.

Note. The fields **name** and **names** are mutually exclusive.

9.3 Domain Active Status

- **PATCH** /api/domain/{id}/active:

with optional body:

`{"active":true}`; if omitted, toggles the current active state.

- **PATCH** /api/domains/active

With required body:

`{"domain_ids":[1,2,3], "active":true/false}` ; if ``active`` is omitted, each domain is toggled.

9.4 Domain Movement

- **PATCH** /api/domain/{id}/parent

With optional body: `{"parent_id": <parentID> }`.

Where {id} is the domain unique ID to be transferred and <parentID> is the domain ID that is its primary. If the body is left blank then the domain is made a parent/primary domain

Notes:

1. Must stay within the same organization;
2. prevents cycles and self-parenting.
3. Cannot move a domain that has subdomains.
4. Response includes parent details when a parent is set.

9.5 Domain Display Relationships

- **PATCH** /api/domain/{id}/relations

Toggles the `display_relationships` field status of the domain with {id}

9.6 Domain Deletion

- **DELETE** /api/domains

With required body:

```
{ "domain_ids": [1,2,3] }:
```

Bulk deletes the domains in array. Rejects domains outside your org or those having subdomains; returns `deleted_ids` or `blocked/unauthorized ids`.

- **DELETE** /api/domain/{id}:

Single delete domain with {id}; disallowed if subdomains exist.

10 Applications

- **GET** /api/applications:

Lists applications for the user's org (id, number, status, timestamps).

Quick cURL Examples

- Login:

```
curl -X POST https://your-host/api/login -H "Content-Type: application/json" -d "email:user@example.com,password:secret"
```

- List domains:

```
curl -H "Authorization: Bearer TOKEN" https://your-host/api/domains
```

- Bulk add domains:

```
curl -X POST https://your-host/api/domains -H "Authorization: Bearer  
TOKEN" -H "Content-Type: application/json" -d  
"application_id:1,names:[one.com,two.com]"
```

- Bulk delete domains:

```
curl -X DELETE https://your-host/api/domains -H "Authorization: Bearer  
TOKEN" -H "Content-Type: application/json" -d "domain_ids:[1,2,3]"
```

Client Setup Checklist

1. Always send **Content-Type: application/json** and **Accept: application/json**.
2. Include **Authorization: Bearer <token>** after logging in.
3. Handle HTTP 401 for bad/expired tokens, 403 for org/permission mismatches, and 422 for validation errors (e.g., invalid domain or subdomain constraints).