DNS API

Last Revision Date: 12/31/2024

Table of Contents

1. Introduction to Markmonitor

Markmonitor provides strategic domain management solutions that help protect the revenue and reputation of the world's leading brands. Since 1999, Markmonitor has served the domain portfolio needs of businesses around the globe, including many of the most visited websites in the world. As an ICANN-accredited domain registrar since its establishment, Markmonitor leverages its extensive industry relationships, innovative technology, and broad expertise to manage and protect company domain portfolios, all with data-driven, white-glove consultation designed to maximize domain portfolio value.

With the Markmonitor API, users can easily empower their workflows and simplify DNS management with the ability to integrate with the tools of their choice to provide thorough and effective management. To ensure the efficiency and security, our API for managing of DNS provides the access to cutting-edge features including a cloud-ready DevOps interface, zone creation, deletion, and record management.

2. Objective

Enable our clients to perform DNS automation, using a Markmonitor DNS API that follows our established business rules. Clients are able to treat us as a single vendor for domains, DNS, certificates, enterprise billing, reporting, forecasting, and support.

The API guide serves the purpose of creating, updating, managing, and deleting zones, notes, and records on Markmonitor Nameservers. It encompasses a range of endpoints specifically dedicated to these tasks.

3. API Interaction Process

Step 1: Obtain an API key from DPA.

Step 2: Authenticate with valid credentials & obtain a bearer token.

Step 3: Select the specific API to use.

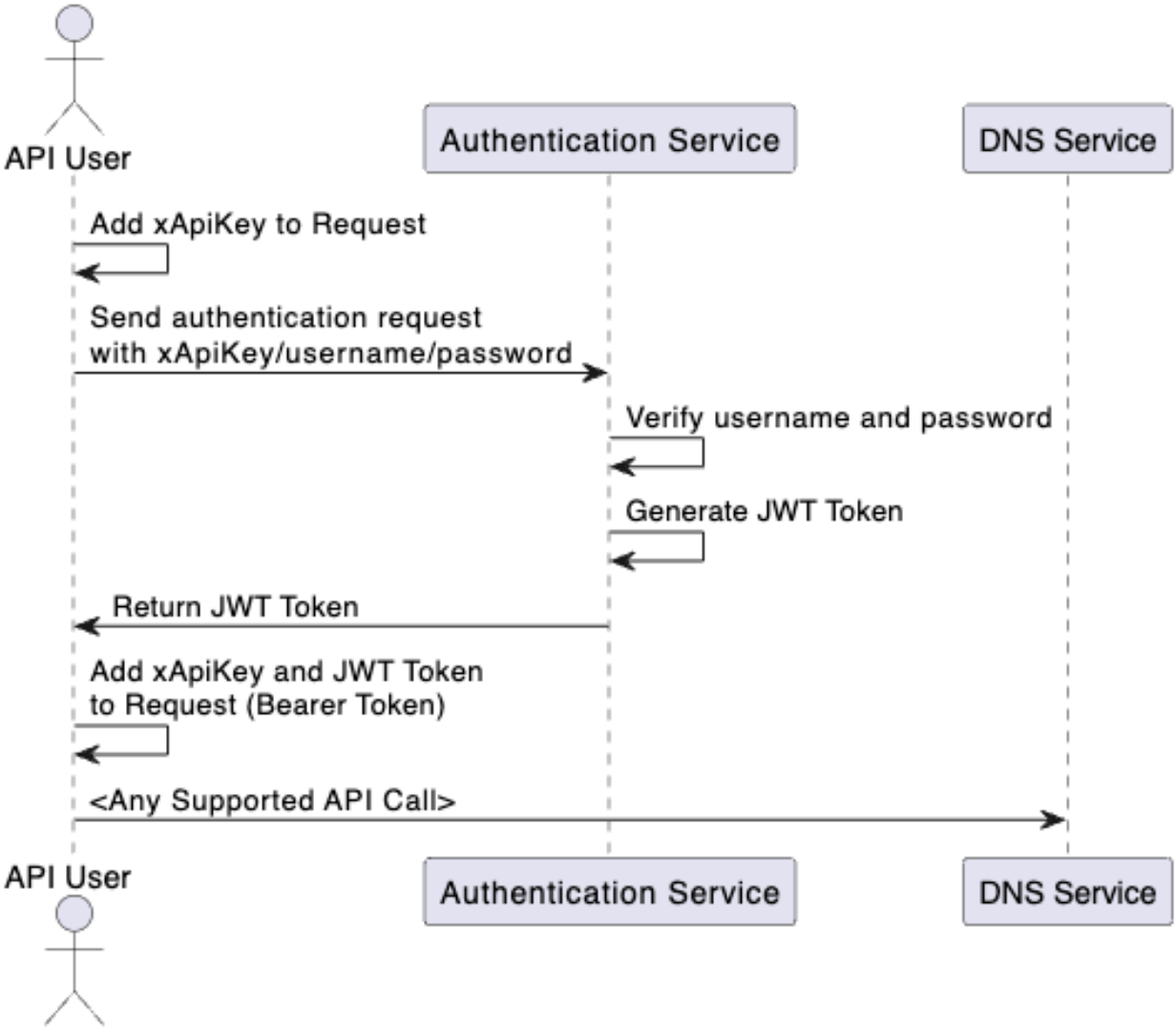
Step 4: Build a request body, add a bearer token, and API key in the request's header, and send it to the API endpoint.

4. Rate-Limiting

The API's rate-limiting function is crucial for ensuring fair and equal use of system resources. The rate limit for the DNS API is 100 requests per minute. It prevents each user or application from overloading the system by limiting the rate at which users can submit requests. By doing this, it is made sure that all users can still access and use the API services.

5. Authentication Methods

Markmonitor uses two forms of authentication for gaining access to system-secured resources, which include: API Key and JWT (JSON Web Token). We require both values to be present for a request to be successfully accepted and processed.



The diagram illustrates the usage flow encompassing various components engaged in the authentication process. Initially, the API user provides their username, password, and API Key (x-api-key) within a REST request to the authentication service, which then authenticates the user's credentials and generates a JWT as a response. In the subsequent API calls to the DNS service, the user appends the obtained JWT as a bearer token for validation and authentication purposes.

5.1. API Key

An API (Application Programming Interface) key is a popular way of authentication where a code is used to identify and authenticate an application or user. *Markmonitor's applications require an "**x-api-key**" header as part of the incoming requests*.

5.1.1. Getting the API Key

Users need to reach out to their DPA to obtain an API key for utilizing the new authentication and DNS API.

5.1.2. Use the API Key

Include the provided value in the request's header with an "x-api-key" header name.

5.2. JWT (JSON Web Token)

The JWT (JSON Web Token) is a compact, independent method of sending data securely. It consists of a header, a payload, and a signature, which are its three primary components. When making API requests, the user must put the token in the "Authorization" header.
To generate and manage JWTs, as well as provide authentication for safe access to the API, Markmonitor created the Authentication API.

Authentication service base URL:

https://api.markmonitor.com/auth/v1

5.2.1. Getting the JWT (Authenticate Endpoint)

URL: https://api.markmonitor.com/auth/v1/auth/authenticate (POST)

To start using the JWT, the user must provide a username and password as part of the request body for the authenticate endpoint.

Response: After a successful authentication request, the system returns a JWT value.

Users must include this token as a "bearer token" in the permission header of their requests to use it for upcoming API calls. This provides secure access to other parts of the system.

5.2.2. JWT Timeout and How to Refresh the Token

URL: https://api.markmonitor.com/auth/v1/auth/refresh (GET)

The JWT has a 90-minute timeout period. The authentication API has a GET endpoint that can be used to refresh the token. By providing the existing JWT via the authentication header this endpoint will renew and return a new value for a previously generated JWT.

6. DNS General Information (Overview)

In its simplest form, a domain name is a familiar string of letters attached to a Top-Level Domain (TLD). For example, in markmonitor.com, "markmonitor" is the chosen phrase and ".com" is the TLD.

The Domain Name System (DNS) is often described as a phonebook, or cell phone contact list, for the internet. Its primary function is to translate human-friendly domain names (like markmonitor.com) into machine-friendly IP Addresses (64.124.14.70).

This translation is essential because, while humans are better able to use memorable, alphabet-centric names, web browsers, servers, computers, and other devices do not like to work with the alphabet. These machines are programmed to use numbers.

DNS Terms

Authoritative Name Server: An authoritative name server is a nameserver (DNS Server) that holds the actual DNS records (SOA, A, MX, NS, etc.) for a particular domain.

Domain Name System (DNS): The Domain Name System is used to change internet domain and computer names into IP addresses. It is based on a hierarchy of servers and databases located throughout the Internet and internal networks.

Domain Name Server: A name server is a specialized server on the Internet that handles queries or questions from your local computer about the location of a domain name. They are like phonebooks for domain names.

IP Address: An Internet Protocol Address (IP Address) is a number assigned to any device (e.g., computer, printer, mobile phone) connected to a network that uses the Internet Protocol for communication.

Internet Service Provider (ISP): A company or organization that provides access to the internet.

Recursive Server: A server that holds a cached copy of internet browsing history for a designated amount of time. If a cached copy does not exist, this server then communicates with the authoritative server for a new copy of a web page.

Root Server: A DNS Server that answers requests for a particular TLD.

TLD: A Top-Level Domain (TLD) is a domain at the highest level in the hierarchical Domain Name System of the Internet. For example, in the domain name www.example.com, the Top-Level Domain is .com.

TLD Nameserver: A Top-Level Domain (TLD) nameserver stores information about domains that share a common extension (for example .com).

6.1. How the Domain Name System works:

When you type a domain name into your browser, a DNS resolution process occurs behind the scenes to find the corresponding IP address.

Below are the steps for this process:

Step 1: DNS Query Initiation: When you enter a domain name, your browser sends a query to a DNS resolver (Server), often provided by your Internet Service Provider (ISP).

Step 2: Recursive DNS Resolver (Server): This resolver (server) acts like a librarian who knows where to find the information. It starts by querying a root nameserver.

Step 3: Root Nameserver: The root server responds with the address of a Top-Level Domain (TLD) Nameserver such as .com, or .org.

Step 4: TLD Nameserver: The TLD server then directs the query to the authoritative nameserver for the specific domain.

Step 5: Authoritative Nameserver: This server holds the actual IP address for the domain and returns it to the resolver (server).

Step 6: IP Address Retrieval: The resolver (server) sends the IP address back to your browser, which can then load the website.

7. Zone General Information (Overview)

A zone is a portion of the DNS namespace that is managed by a specific organization or administrator. It allows for more granular control over DNS components, such as authoritative nameservers.

All the information for a zone is stored in a zone file. A Zone File is a text file that holds DNS records, and these records provide instructions for how and where domain names resolve.

Zone Terms

A Records: Address Records ('A' Records) are what DNS really boils down to. An 'A' record gives you the IP address of a domain. That way, users that try to go to www.example.com will get to a specific IP address. An 'A' record must always point to an IP address and can sometimes point to multiple IP addresses for load-balancing purposes.

AAAA Records: This is an 'A' record for IPv6 addresses. Essentially, the IPv4 system has run out of IP address combinations, and IPv6 has added significantly more options for IP addresses.

CNAME Record: A Canonical Name (CNAME) Record is a type of DNS record that maps a domain name (alias) to the true (or canonical) domain. CNAME records are generally used to map subdomains to the domain hosting the content.

MX Record: This record indicates incoming and outgoing mail for a domain that is handled by the specified mail hostname.

NS Record: Name Server (NS) Records define the authoritative nameservers for a domain. They specify which nameservers are responsible for storing the DNS records for that domain.

PTR Record: A Pointer record (PTR) maps an IP address to a domain name. This is the opposite of an A record, which maps domains to IP addresses.

SOA Record: The record that identifies the zone file's owner. It's almost like the WHOIS Record for a zone file in that it provides contact information for the owner and integral information for the zone to function: it holds information about caching, Time to Live (TTL), and the email address for the owner. This is what also

ties the zone file to the owner, sets the rules for the zone file, and provides information about who to contact if there are issues.

SRV Record: A Service record (SRV) is used to define the location of servers for specific services such as VoIP (Voice over IP), instant messaging, and other applications that require service discovery.

TXT Record: Holds text information for various purposes, such as SPF records for email validation.

8. DNS API

DNS service base URL:

https://api.markmonitor.com/dns/v1

8.1. Zone

A zone is a portion of the DNS namespace that is managed by a specific organization or administrator. It allows for more granular control over DNS components, such as authoritative nameservers.

All information for a zone is stored in a zone file. A zone file is a text file that holds DNS records, and these records provide instructions for how and where domain names resolve.

The Markmonitor DNS API allows users to create, update, manage, and delete zones on Markmonitor Nameservers programmatically through their own systems. This can eliminate the manual step of logging directly into the Markmonitor portal and allows for programmed changes to occur automatically once required approvals have been received.

8.1.1. Create a zone

URL: https://api.markmonitor.com/dns/v1/zone (POST)

A client can create a zone by making a POST request to the zone endpoint. The user must send all data that defines a DNS Zone in their API request, by passing a complete zone object as a request body. This means that every required property and record must be provided at the time of creation. These properties are the zone type, the maximum number of DNS record sets allowed in the zone, and the

authoritative nameservers for the zone. If applicable, this may also include the virtual networks for DNS registration and DNS resolution.

Additionally, the zone creation endpoint allows users to pass an optional list of records to be created at the same time as that zone. This means that while creating the zone, the user can also include the DNS records such as A records, CNAME Records, MX Records, and TXT records.

This saves time as users can create and update a zone simultaneously.

Additionally, this endpoint will require one of the following values set in the **type** property:

Standard: A standard zone will be created as specified above. Passing a complete zone body and indicating **type=STANDARD**, will create a standard zone for the user.

Premium: A user can create a premium zone by following the specified steps above while indicating **type=PREMIUM**. For this zone type, the API user can specify additional FORWARDING record types to create multiple FORWARDING records on the zone at the time of creation.

Forwarding: A user can create a forwarding zone by following the specified steps above, while indicating **type=FORWARDING**. Additionally, a new **forwardTo** property will be required to specify where the zone will forward to. The client should only provide this property when using **type=FORWARDING** to set up the DNS records.

Response: After successful execution of the zone create request, the system will return a zone object together with a generated ID (resourceId) for the newly created zone.

8.1.2. Update a zone

URL: https://api.markmonitor.com/dns/v1/zone/{resourceId} (PATCH)

A client can use the PATCH method to update a zone using Markmonitor's zone API endpoint. To modify any value of an existing zone, the user can provide the zone's ID (resourceId) as a path parameter. This endpoint allows users to update specific

values without needing to submit an entire zone object in the request body. This can be done by passing one or more properties with correct values/objects.

Response: After successful execution of the zone update request, the system will return a modified zone according to the specified values/objects.

8.1.3. Get a specific zone

URL: https://api.markmonitor.com/dns/v1/zone/{resourceId} (GET)

A user can utilize the GET method to retrieve the details and configuration of a specific DNS zone, using Markmonitor's zone API endpoint. To do so, the user must provide the zone's resourceId as the path parameter.

Response: After successful execution of the GET request, the system returns the information about the requested zone. This includes the various details and zone files.

8.1.4. Search for zones

URL: https://api.markmonitor.com/dns/v1/zone (GET)

To retrieve a list of zones associated with the authenticated user's account, the user can utilize the GET method using Markmonitor's zone API endpoint.

The user can manipulate their search results and page view using the query parameters laid out below:

Query parameters are as follows:

Names: Zones can be searched for in bulk. To do this, a user must provide a list of zone names and each zone name must be separated by a comma only. There should be no spaces. Each name in this provided list will undergo a substring match process, meaning it will return results that match some or all of the provided string.

Template: A user can indicate whether to use a template for filtering purposes. If the user sets **template=true** as part of the query parameters, it will only include zone templates in the response.

Types: The zone types can be provided to refine the search results. To search for zones by type, the API user will need to use one or more of these values: STANDARD, PREMIUM, or FORWARDING. For reference, these zone types are defined in section *8.1.1 Create a zone*.

Page: The page number of the results to retrieve uses a 0-based index, meaning the first page is indexed as 0, the second page as 1, and so on.

Size: The system will default to providing 50 search results per page. For a size other than the default, the API user can add the size parameter to the query string and assign it to the size desired, e.g. size=2000.

Sort: A user can specify the order in which results should be sorted based on certain properties, with an optional indication of whether the sorting should be in ascending or descending order. The available values for sorting are name, premium, type, and template.

Response: After successful execution of the GET request, the system will return a list of zones based on the specified search query parameters.

8.1.5. Delete a specific zone

URL: https://api.markmonitor.com/dns/v1/zone/{resourceId} (DELETE)

To delete an existing zone associated with the authenticated user's account, the user can utilize the DELETE method using Markmonitor's zone API endpoint.

The user must provide the zone ID (resourceId) as a path parameter to specify which zone is to be deleted.

Please note that the API user must have the proper permissions to delete a zone.

Response: After successful execution of the DELETE request, the system will delete the specified zone.

8.2. Notes

Zone notes are provided to add hints about a zone.

8.2.1. Create a zone note

URL: https://api.markmonitor.com/dns/v1/zone/{resourceId}/note (POST)

A user can utilize the POST method to create a zone note, using Markmonitor's zone API endpoint. The user must send all data that defines the zone note in their API request, by passing a complete zone note object as a request body. The user must provide the parent zone's resourceId as the path object, to ensure the note is created for the correct zone.

Response: After a successful execution of the zone note create request, the system will return the completed note for the provided resourceId.

8.2.2. Get a zone note by resource ID

URL: https://api.markmonitor.com/dns/v1/zone/{resourceId}/note/{zoneNoteResourceId} (GET)

To retrieve a zone note through Markmonitor's zone API, the user can utilize the GET method. Both the resourceId of the parent zone and the zoneNoteResourceId must be provided to ensure the correct note is retrieved for the specified zone.

The resourceId identifier is used to specify the zone a user wishes to retrieve information from, while the resource id of the zone note (zoneNoteResourceId) indicates which specific note is being requested.

Response: After successful execution of the GET request, the system will return the requested note (unique to the zoneNoteResourceId) for the specified zone (unique to the resourceId).

8.2.3. Get a list of Zone Notes from a Zone

URL: https://api.markmonitor.com/dns/v1/zone/{resourceId}/note (GET)

A user can utilize the GET method to retrieve the zone notes of a specific DNS zone, using Markmonitor's zone API endpoint. To do so, the user must provide the zone's resourceId as the path parameter to ensure the information received is for the desired zone.

Query parameters are as follows:

Text: A user can search by specific text.

Start date: A user can search by the start date.

End date: A user can search by the end date.

Page: The page number of the results to retrieve uses a 0-based index, meaning the first page is indexed as 0, the second page as 1, and so on.

Size: The system will default to providing 50 search results per page. For a size other than the default, the API user can add the size parameter to the query string and assign it to the size desired, e.g. size=2000.

Response: After successful execution of the GET request, the system will return the list of zone notes for the specified search criteria.

8.3. History

Zone history keeps track of changes to each zone and its records.

8.3.1. Search for history of a zone

URL: https://api.markmonitor.com/dns/v1/zone/{resourceId}/history (GET)

To get a complete history list for a specific zone, with all changes and actions taken over time, the user can provide the zone's ID (resourceId) as a path parameter and send a request to this endpoint. Historical records of a zone often indicate performed actions such as date of creation, updates and added records, zone note creation, etc.

Query parameters are as follows:

Text: A user can search by specific text.

Start date: A user can search by the start date.

End date: A user can search by the end date.

Page: The page number of the results to retrieve uses a 0-based index, meaning the first page is indexed as 0, the second page as 1, and so on.

Size: The system will default to providing 50 search results per page. For a size other than the default, the API user can add the size parameter to the query string and assign it to the size desired, e.g. size=2000.

Response: After a successful execution of the GET request, the system will return a paged list of the history of the zone. This will include the date when the history line was created, a username of the user who made a change that initiated the generation of history, a text field describing the change/action made, and id for each history entity. Note: If the system did it, no username will be in the response.

8.4. Templates

Zone templates are predefined configurations that simplify the creation and management of DNS zones. The API user is allowed to create a zone by assigning an existing template. The DNS records in that template will be applied to the newly created zone, saving time and effort.

8.4.1. Get a list of zone templates by the authenticated user's account

URL: https://api.markmonitor.com/dns/v1/zone/template (GET)

A client can use the GET Method to retrieve a list of zone templates by the authenticated user's account from this endpoint.

The user can manipulate their search results and page view using the query parameters laid out below:

Query parameters are as follows:

Page: The page number of the results to retrieve uses a 0-based index, meaning the first page is indexed as 0, the second page as 1, and so on.

Size: The system will default to providing 50 search results per page. For a size other than the default, the API user can add the size parameter to the query string and assign it to the size desired, e.g. size=2000.

Response: After successful execution of the GET request, the endpoint returns a list of zone templates by the authenticated user's account. The response may include details such as template names, descriptions, creation dates, etc.

8.5. Records

All information for a zone is stored in a zone file, which is a text file that holds DNS records. These records provide instructions for resolving domain names. They are entries in DNS zone files that map domains to IP addresses and other resources. Some of the more popular records are A, AAAA, NS, MX, PTR, TXT, SRV, CNAME, SOA, and CAA records *(definitions provided in the 7 Zone General Information (Overview) section).*

8.5.1. Create a Record on a zone

URL: https://api.markmonitor.com/dns/v1/{resourceId}/record (POST)

A user can utilize the POST method to create a record on an existing zone, using Markmonitor's zone API endpoint. The resourceId of the parent zone must be provided as the path parameter, to ensure the note is created for the correct zone.

To ensure the record is properly created, the user must send all data in their request by passing a complete record object as a request body. This means that every property of the record must be provided at the time of creation. The type of record might be an A, CNAME, or MX record. Additional information might be TTL, Priority, and Weight.

Response: After a successful record create request, the system will create the requested record on the specified zone.

8.5.2. Update a Record on a zone

URL: https://api.markmonitor.com/dns/v1/zone/{resourceId}/record/{recordResourceId} (PATCH)

A client can use this endpoint to modify any value of an existing record by specifying both the resourceId and recordResourceId as path variables.

The resourceId identifier is used to specify the parent zone, while the resource id of the record (recordResourceId) ensures that the correct record within the specified zone is updated.

This endpoint allows users to update specific values without needing to submit an entire zone object in the request body. This can be done by passing one or more properties with correct values/objects.

Response: After successful execution of the record update request, the system will return a modified record according to the specified values/objects.

8.5.3. Search for Records in a zone

URL: https://api.markmonitor.com/dns/v1/zone/{resourceId}/record (GET)

A user can utilize the GET method to retrieve all records within a specified zone, by using Markmonitor's zone API endpoint. To do so, the user must provide the zone's resourceId as the path parameter.

The user can manipulate their search results and page view using the query parameters laid out below:

Query parameters are as follows:

Names: Records can be searched for in bulk. To do this, a user must provide a list of records, and each record name must be separated by a comma only. There should be no spaces. Each name in this provided list will undergo a substring match process, meaning it will return results that match some or all of the provided string.

Data: The value of the data property will depend on the type of record. For context, if the record is of type=A, the system knows the data value will be an IP address. Similarly, if the record is of type=NS, the system knows the data value will be a nameserver address. To illustrate, if there is an NS record that has ns.markmonitor.com and a TXT record that has testing.markmonitor.com in the data fields, the API user can utilize the GET method on this dns/v1/record endpoint

with the query string data=markmonitor.com. This will return both the DNS and TXT records mentioned above, because both have the markmonitor.com substring in the data property.

Record type: The record type can be provided to refine list results. These options will be as follows: A, AAAA, NS, MX, PTR, TXT, SRV, CNAME, SOA, CAA. For reference, these record types are defined in section *7 Zone General Information (Overview)*. This is a fixed list of possible values that an API user can assign to the 'Record Type' property.

Page: The page number of the results to retrieve uses a 0-based index, meaning the first page is indexed as 0, the second page as 1, and so on.

Size: The system will default to providing 50 search results per page. For a size other than the default, the API user can add the size parameter to the query string and assign it to the size desired, e.g. size=2000.

Response: After successful execution of the GET request, the system will return a page of records for the specified zone.

8.5.4. Get a specific Record from a zone

URL: https://api.markmonitor.com/dns/v1/zone/{resourceId}/record/{recordResourceId} (GET)

A user can utilize the GET method to retrieve a specific record from within a zone, by using Markmonitor's zone API endpoint. To do so, the user must provide both the parent zone's ID (resourceId) and the specific zone's id (recordResourceId) as the path parameters.

The resourceId identifier is used to specify the parent zone a user wishes to retrieve information from, while the resource id of the record (recordResourceId) indicates which specific record is being requested.

Response: After successful execution of the GET request, the system will return the requested record (unique to the recordResourceId) for the specified zone (unique to the resourceId).

8.5.5. Delete a Record on a zone

URL: https://api.markmonitor.com/dns/v1/zone/{resourceId}/record/{recordResourceId} (DELETE)

To delete a record associated with an existing zone, the user can utilize the DELETE method using Markmonitor's zone API endpoint. The user must provide both the resourceId and recordResourceId as path parameters. The resourceId identifier is used to specify the parent zone a user wishes to retrieve information from, while the resource id of the record (recordResourceId) indicates which specific record is being requested.

Please note that the API user must have full permissions on a zone object to delete that zone.

Response: After successful execution of the DELETE request, the system will delete the specified zone.

9. Things to know

9.1. HTTP Return Codes

There are several possible HTTP return codes that DNS API can return.

On successful execution of a request, a user gets *200 (OK)* for all requests, *201 (CREATED)* for zone and record create (POST) calls.

There are numerous HTTP error codes with appropriate error descriptions. They include *BAD_REQUEST (400)*, *UNAUTHORIZED (401)*, *FORBIDDEN (403), NOT_FOUND (404)*, *CONFLICT (409)*, *TOO_MANY_REQUESTS (429)*, *INTERNAL_SERVER_ERROR (500)*, *BAD_GATEWAY (502)*, *SERVICE_UNAVAILABLE (503)*, *GATEWAY_TIMEOUT (504)*.

There are numerous HTTP error codes with appropriate error descriptions. They include *BAD_REQUEST (400)*, *UNAUTHORIZED (401)*, *FORBIDDEN (403), NOT_FOUND (404)*, *CONFLICT (409)*, *INTERNAL_SERVER_ERROR (500)*, *BAD_GATEWAY (502)*, *SERVICE_UNAVAILABLE (503)*, *GATEWAY_TIMEOUT (504)*.

9.2. Path Variables and Query Parameters

9.2.1. 'resourceId' path variable

The entity's unique identifier is represented by the 'resourceId' path parameter, which is a UUID (Universal Unique Identifier) data type - a 128-bit value used to uniquely identify an object or entity on the internet. For activities involving zone/record/note/history retrieval, updating, or deletion, this identifier is required. Users can target certain entities for such operations by including the 'resourceId' in API queries, assuring accuracy in maintaining certificate data.

9.2.2. URL query parameter

Some endpoints use URL query strings to filter results. URL query strings are appended to the endpoint URL using "?", with additional query strings appended using "&".

Examples:

https://api.markmonitor.com/dns/v1/zone?types=STANDARD

https://api.markmonitor.com/dns/v1/zone?size=3&page=6