Markmonitor™

SSL Certs API

Last Revision Date: 4/18/2024

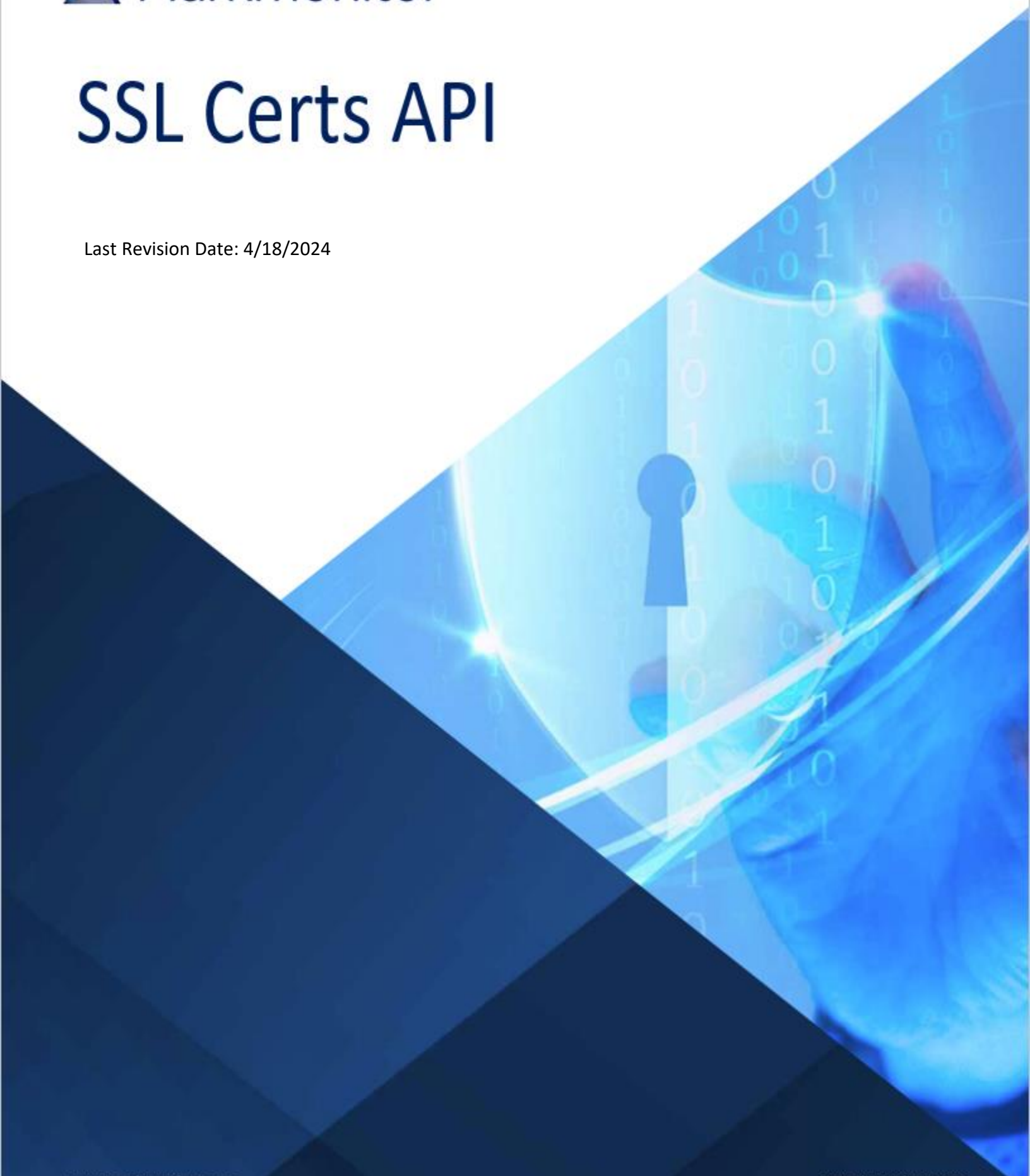


Table of Contents

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 the complex lifespan of TLS/SSL certificates 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 certificates provides the access to cutting-edge features including cloud-ready DevOps interface, state-of-the-art encryption, complete audit logs, CSR verification, and easy maintenance using SAN values.

2. Objective

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

The API guide serves the purpose of ordering certificates for different organizations. It encompasses a range of endpoints specifically dedicated to tasks such as buying a certificate and getting information about various certificates.

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.

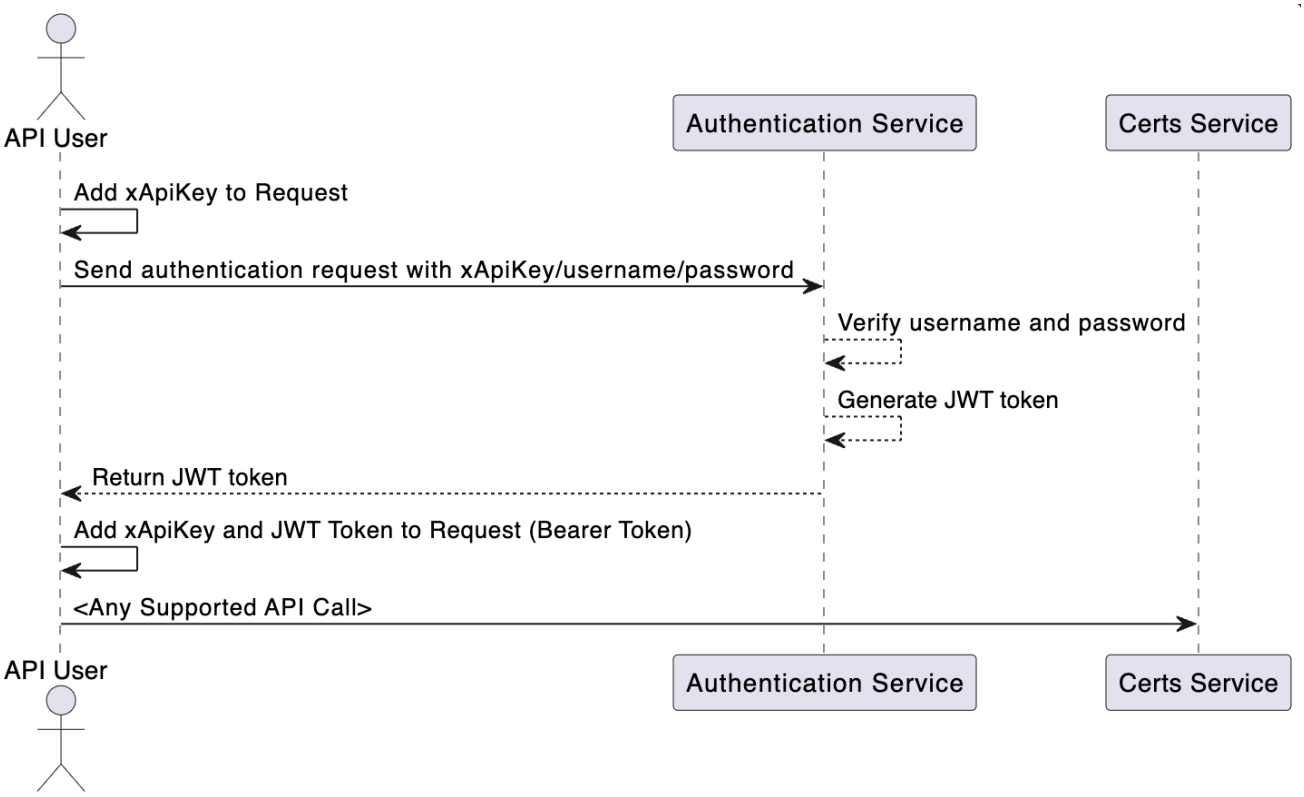
Step 4: Build a request body, add a bearer token, and API key in the request's header, and send it to the API endpoint.

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 Cert 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 cert 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 SSL certificates 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. SSL Certificate Overview

An SSL (Secure Sockets Layer) certificate, at its most fundamental level, is a digital certificate that establishes a secure, encrypted connection between a web server and a user's web browser. It guarantees that all data transferred between the two parties is private and secure against unauthorized access.

The Certificate Management API solution offered by Markmonitor makes it easy to quickly acquire, view, and manage SSL certificates. This makes tracking certifications within the organization easier.

Types of SSL certificates

DV SSL Certificates: As only a single domain verification step is needed to confirm ownership of domain records, DV certificates are comparatively simple to obtain.

OV SSL Certificates: These certificates can be readily acquired using a one-step domain verification procedure with a previously validated corresponding organization. They are appropriate for most websites and provide up to 256-bit encryption, showing a standard padlock icon in web browsers.

EV SSL Certificates: E-commerce and banking websites frequently employ EV certificates, which are widely recognized for the highest degree of security and trust and undergo rigorous validation.

Markmonitor offers all three types of certificates from reputable Certificate Authorities like DigiCert, Symantec, GeoTrust, and Thawte, allowing users to choose based on security needs.

The following elements are typically needed to comply with the requirements for an SSL certificate:

- Domain Ownership: Certificate authorities (CAs) confirm that the user owns or has control over the domain for which the certificate is requested. This can entail verifying information related to domain registration.
- For Organization Validation (OV) or Extended Validation (EV) certificates, extra identity verification of the organization is necessary. This entails confirming the company's legitimacy and its actual address.
- Phishing and malware checks: CAs run checks to make sure the domain isn't connected to phishing or malware-related activity.

6.1. Managing SSL Certificates

With features including ordering, revoking, and reissuing certificates, the Markmonitor API is useful for managing SSL certificates. This is how it makes these features achievable:

6.1.1. Obtaining SSL Certificates - Step by Step

Step 1: Creating Organization: Users need to set up an organization to be able to start the process by providing every necessary detail, including the name of the organization, its address, its city, state, country, ZIP code, phone number, a list of contacts with the contact types, and the provider. When setting up an organization, a user must create necessary contacts at the same time.

Step 2: Creating Additional Contacts if needed: The user needs the following details to add a new contact to an existing organization: organization ID, provider, contact types, email address, phone number, job title, and first and last name.

Step 3: Completing Necessary Validations: Markmonitor's CA partner will reach out to the user’s organization through publicly available phone numbers and contact details, to work with the provided contact to validate the organization. If extended validation has been requested, those contacts will also need to be validated in addition to the organization.

Step 4: Creating an Order: As an essential step in the SSL certificate issuance process, creating an order requires providing mandatory fields, such as contacts and cert information.

Note: For existing Markmonitor SSL users, see section *8.1 Existing Markmonitor users*.

Step 4: Creating an Order: As an essential step in the SSL certificate issuance process, creating an order requires providing mandatory fields, such as organization id, contacts, cert type, locale, provider, and cert information.

6.1.2. Validate Control of the Domain

To validate control of the domain using the Markmonitor API, the user can choose from 4 different validation types:

- EMAIL: To demonstrate control over the domain, an email recipient follows the instructions in a confirmation email sent for the domain. The confirmation process consists of visiting the link provided and following the instructions on the page.
- DNS_CNAME_TOKEN: User needs to create a new CNAME record at their DNS provider, paste the unique token in the hostname field, select CNAME as a record type and enter *dcv.digicert.com* in the host field (or equivalent).
- HTTP_TOKEN: User needs to host a file containing a provided random value at a predetermined location on their website: [your-domain]/.well-known/pki-validation/fileauth.txt.
- DNS_TXT_TOKEN: Add a provided token to the domain's DNS as a TXT record.

7. SSL Certificate API

SSL Certificate service base URL:

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

7.1. Organization

The definition of "organization" is important for an SSL certificate lifecycle since it is necessary when placing certificate orders. An organization specifies the legal entity or company for whom the certificate is intended.

OV certificates involve an extensive validation procedure that includes independent Certificate Authorities (CAs) verifying the organization's details, examining the organization's legal registration, and authenticating the organization's physical presence. By ensuring the validity of the entity requesting the certificate, this rigorous validation process strengthens online security and confidence. The validation of an organization remains effective for 13 months. After that period, a new validation process will

automatically be initiated by the Certificate Authority's system to uphold the integrity of the SSL certificate.

7.1.1. Create an organization

URL: `https://api.markmonitor.com/cert/v1/organization` (POST)

A user can use the POST method to create an organization using Markmonitor's organization API endpoint. To accurately determine the organization's identity a user must provide the organization's details, such as the organization's name, address, city, state, country, zip code, and list of contacts with their contact types. When creating an organization, Markmonitor's system automatically submits it for validation, so no additional action is needed. The validations on the organization are based on the contact validations, thus a user must create at least 1 contact at the same time. New contact entities are created based on the information provided.

Response: After a successful execution of the organization create request, the system returns an organization object together with generated IDs (resourceId) for all newly created entities.

7.1.2. Get Information about a specific organization

URL: `https://api.markmonitor.com/cert/v1/organization/{resourceId}` (GET)

To get information about the specific organization, the user can use the GET method and provide the organization's ID (resourceId).

Response: The system returns the complete organization entity that, in addition to already provided organization details in the organization creation endpoint.

7.1.3. Get a paged list of organizations based on provided filters

URL: `https://api.markmonitor.com/cert/v1/organization` (GET)

A user can use this endpoint as a search functionality where organizations from their account can be filtered based on the organization name or provider ID. Note: By default, if name or provider ID are not provided, the results will be filtered based on the account and the user associated with the provided JWT.

Response: A list of organization entities based on the provided filtering value.

7.1.4. Get paged history of a specific organization based on provided filters

URL: `https://api.markmonitor.com/cert/v1/organization/{resourceId}/history` (GET)

To get a complete history list for a specific organization, with all changes and actions taken over time. The user can provide the organization's ID (resourceId) and send a request to this endpoint. Filtering options for this endpoint include the user ID associated with the history line and/or the text value included in the history line.

Response: A paged list of the history of the organization which includes the date when the history line was created, a 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.

7.1.5. Get a specific history for the organization

URL: `https://api.markmonitor.com/cert/v1/organization/{resourceId}/history/{historyResourceId} ` (GET)

A user is also able to get a single history entity for a specific organization by providing organization and history resourceId.

Response: A single history entity that includes the date when the history line was created, the username of a user who made a change that initiated the generation of a history, a text field describing the change/action made, and an ID for each history entity. Note: If the system did it, no username will be in the response.

7.2. Contact

When ordering an SSL Certificate, there are 2 types of contacts with different responsibilities:

- ORGANIZATION_CONTACT: ensures that the organization and its legal name are accurately represented on the SSL certificate. It must exist for an organization to be validated, which is required for the issuance of OV and EV certificates (together with validated EV approver). This type of contact also receives the issued certificate, as well as notifications about its expiration.
- EXTENDED_VALIDATION_APPROVER: approves the certificate issuance for EV certificates for the organization. Extended Validation (EV) is the strongest SSL certificate for establishing robust online trust. Managed by a Certificate Authority (CA), EV certificates are renowned for their stringent validation procedures. An email is dispatched to the EV contact specified during the certificate ordering

process, and the user is required to approve the certificate from there, enabling the progression of the order.

Note: Both contact types can be assigned to the same individual. Markmonitor also creates a TECHNICAL_CONTACT in each organization that is assigned to every order to provide you with all necessary support.

7.2.1. Create a contact

URL: `https://api.markmonitor.com/cert/v1/contact` (POST)

When creating a new contact for an existing organization, the user can take advantage of the POST method by providing particulars such as the user's first and last names, job title, email address, phone number, organization ID (resourceId), provider and a list of contact types in the JSON request body. Like organizations, EV contact also needs to undergo validation, so the Markmonitor system automatically submits them to initiate the validation process.

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

7.2.2. Get Information about a specific contact

URL: `https://api.markmonitor.com/cert/v1/contact/{resourceId}` (GET)

To get information about the specific contact, the user can use the GET method and provide the contact's ID (resourceId).

Response: The system returns the contact entity.

7.2.3. Get a paged list of contacts based on provided filters

URL: `https://api.markmonitor.com/cert/v1/contact` (GET)

A user can use this endpoint as a search functionality where contacts from their account can be filtered based on the email address. Note: By default, if email is not provided, the results will be filtered based on the account and the user associated with the provided JWT.

Response: A list of contact entities based on the provided filtering value.

7.2.4. Modify a specific contact

URL: `https://api.markmonitor.com/cert/v1/contact/{resourceId}` (PATCH)

A user can use this endpoint to modify any value of the existing contact. If the contact being modified is of type EV approver, the organization will be submitted for re-validation.

Response: The system returns modified contact entity.

7.2.5. Delete contact

URL: `https://api.markmonitor.com/cert/v1/contact/{resourceId}` (DELETE)

A user has the ability to delete a contact by providing a contact ID (resourceId) as a path variable. The following requirements must be met for a contact to be deleted:

- Contact cannot be deleted if it is associated with an active order
- Contact cannot be deleted if it is the last contact associated with an organization or the last EV contact

Note: Each time an EV approver is deleted, the corresponding organization will be submitted for re-validation.

7.2.6. Get paged history of a specific contact based on provided filters

URL: `https://api.markmonitor.com/cert/v1/contact/{resourceId}/history` (GET)

To get a complete history list for a specific contact, with all changes and actions taken over time, the user can provide the contact's ID (resourceId) and send a request to this endpoint. Filtering options for this endpoint include the user ID associated with the history line and/or the text value included in the history line.

Response: A paged list of the history of the contact which includes 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 backend system made a change, no username will be in the response.

7.2.7. Get a specific history for the contact

URL: `https://api.markmonitor.com/cert/v1/contact/{resourceId}/history/{historyResourceId} ` (GET)

A user is also able to get a single history entity for a specific contact by providing contact and history IDs (resourceIds).

Response: A single history entity that includes the date when the history line was created, the user who made a change that initiated the generation of a history, a text field describing the change/action made, and an ID for each history entity. Note: If the system did it, no username will be in the response.

7.3. Cert

Regarding the cert entities, the Markmonitor system associates each certificate with a unique order in a one-to-one relationship. Therefore, all actions that pertain to management of certificates will be explained in the Order section.

7.3.1. Parse CSR

URL: `https://api.markmonitor.com/cert/v1/cert/parse` (POST)

Certificate Signing Requests (CSRs) are parsed via this POST endpoint. The JSON body of a request accepts a list of CSRs passed in a String format. This endpoint makes it easier to handle CSR data quickly when ordering certificates, ensuring that the correct information is appropriately processed and ready for the following stages of the certificate ordering process. It cooperates with other endpoints, such as the endpoint for certificate issuance requests, to make ordering certificates within the system easy and automated. Note: This is not a necessary step in the process of ordering certificates, but another tool for users to verify their CSR records.

Response: The system returns a list of CSR objects that include many parsed fields, such as CSR, common name, organization name, organizational unit, locale, state, country, email, signature algorithm, etc.

7.4. Order

In obtaining an SSL certificate, the order entity plays the main role. Most important functionalities in the certificate lifecycle are handled via Order API endpoints. Entities described in previous sections (such as organization, and contacts) are required to be completed prior to placing an order and obtaining the SSL certificate.

Orders can be placed while the organization is pending validation, but certificate issuance within the order is contingent upon organization validation completion. Once an EV order request is made, order approval is requested via email where the EV contact assigned to

that order is prompted to click on the provided link. Furthermore, the order fulfillment notification emails are sent to the email address provided for SSL Contacts.

Different types of SSL certificates Markmonitor supports, each with its own verification level:

- SSL_EV_BASIC: Extended Validation (EV) SSL certificate with basic verification. EV certificates provide the highest level of assurance to website visitors, as they require rigorous validation of the requesting entity's identity.
- SSL_OV_BASIC: Organization Validation (OV) SSL certificate with basic verification.
- SSL_EV_SECURESITE: Extended Validation (EV) SSL certificate for the SecureSite product. This suggests a high-assurance EV certificate for securing websites.
- SSL_OV_SECURESITE: Organization Validation (OV) SSL certificate for the SecureSite product. Similar to SSL_OV_BASIC, this would indicate a lower-level validation for organizational identity.
- SSL_EV_SECURESITE_PRO: Extended Validation (EV) SSL certificate for the SecureSite Pro product.
- SSL_OV_SECURESITE_PRO: Organization Validation (OV) SSL certificate for the SecureSite Pro product. Similar to SSL_OV_BASIC and SSL_OV_SECURESITE, but likely associated with the Pro version.
- SSL_DV_GEOTRUST: Domain Validation (DV) SSL certificate issued by GeoTrust.
- SSL_OV_GEOTRUST_TRUEBIZID: Organization Validation (OV) SSL certificate issued by GeoTrust with the TrueBizID product.
- SSL_EV_GEOTRUST_TRUEBIZID: Extended Validation (EV) SSL certificate issued by GeoTrust with the TrueBizID product.
- SSL_DV_THAWTE: Domain Validation (DV) SSL certificate issued by Thawte.
- SSL_EV_THAWTE_WEBSERVER: Extended Validation (EV) SSL certificate for the Thawte WebServer product.
- SSL_OV_THAWTE_WEBSERVER: Organization Validation (OV) SSL certificate for the Thawte WebServer product.

7.4.1. Create an order

URL: `https://api.markmonitor.com/cert/v1/order` (POST)

When creating a new order for an existing organization, the user can use the POST method by providing the necessary information in the request body. This includes the

organization ID, a list of contacts, type of the cert, locale, provider, and cert information (such as common name, DNS names, CSR, DVC method, etc.)[1] in the JSON request body. Markmonitor API supports two SSL algorithms (algorithmHash): RSA and ECC. Furthermore, the default value is a full chain certificate. SHA_1 is an option that can be selected (by setting rootHash to SHA_1), however, some non-browser applications do not support SHA_1 so we do not recommend changing from default unless necessary.

Response: After a successful order create request, the system returns a 202 "Accepted" response code and the order object. You can then make GET requests with the order ID (resourceId) to query the status of the order.

7.4.2. Get Information about a specific order

URL: `https://api.markmonitor.com/cert/v1/order/{resourceId}` (GET)

To get information about the specific order, the user can use the GET method and provide the order's ID (resourceId). This way, users can use the Markmonitor API that allows them to programmatically interact with their orders and certificates.

Response: The system returns the order entity together with a complete cert entity and, the information about the order status.

Below is provided a list of all order statuses:

- CREATED: Initial status for all orders accepted by the Cert API.
- REISSUE_REQUEST_PENDING: Reissue request has been received by Cert API, but not yet accepted by provider/DigiCert
- DIGI_PENDING: Cert order has been received by DigiCert. This is the provider's initial order status.
- DIGI_REISSUE_PENDING: Reissue request received by DigiCert and is in process.
- DIGI_REISSUE_FAILED: DigiCert was unable to fulfill reissue request.
- DIGI_PROCESSING: Order was approved and is being processed by DigiCert.
- DIGI_ISSUED: Order was validated, and certificate can be downloaded.
- DIGI_REVOKED: Order was revoked.
- DIGI_CANCELED: Order was canceled.
- DIGI_NEEDS_APPROVAL: The order request must be approved before DigiCert can process the order.
- DIGI_WAITING_PICKUP: For client certificates, the order is ready, and a user has received an email with a link to generate the certificate.

- DIGI_REJECTED: Order request was rejected at DigiCert.
- DIGI_EXPIRED: Order has expired.
- DIGI_FAILED: DigiCert was unable to fulfill the order creation request.

7.4.3. Get a paged list of orders based on provided filters

URL: `https://api.markmonitor.com/cert/v1/order` (GET)

A user can use this endpoint as a search functionality where orders can be filtered based on the number of filters: provider ID, organization ID, list of certificate types, list of certificate orders statuses, a list of common names, and valid until date. Note: By default, if no filters are provided, the results will be filtered based on the account and the user associated with the provided JWT.

Response: A list of order entities based on the provided filtering value.

7.4.4. Get a list of prices

URL: `https://api.markmonitor.com/cert/v1/order/prices` (GET)

A user can make requests to this endpoint to retrieve information about the prices for each SSL product Markmonitor offers.

Response: A list of price entities. Price entity includes information about cert type, provider, term, total price, and a list of products with product name, item and price value.

7.4.5. Download a certificate from an order

URL: `https://api.markmonitor.com/cert/v1/order/{resourceId}/downloadCert` (GET)

To download a certificate from a specific order and in a specific format, the user can use this endpoint by providing the order's ID (resourceId) as a path parameter and format as a query parameter. Markmonitor's certificate API supports two format types: p7b and pem_all. Note: a request needs to have the *Accept* header set to *application/x-pkcs7-certificates*.

Response: The endpoint returns the certificate in the requested format.

7.4.6. Get paged history of a specific order based on provided filters

URL: `https://api.markmonitor.com/cert/v1/order/{resourceId}/history` (GET)

To get a complete history list for a specific order, with all changes and actions taken over time, the user can provide the order's ID (resourceId) and send a request to this endpoint. Filtering options for this endpoint include the user ID associated with the history line and/or the text value included in the history line.

Response: A paged list of the history of the order which includes 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.

7.4.7. Get a specific history for the order

URL:
`https://api.markmonitor.com/cert/v1/order/{resourceId}/history/{historyResourceId}` (GET)

A user is also able to get a single history entity for a specific order by providing order and history IDs (resourceId).

Response: A single history entity that includes the date when the history line was created, the username of a user who made a change that initiated the generation of a history, a text field describing the change/action made, and an ID for each history entity. Note: If the system did it, no username will be in the response.

7.4.8. Change order based on the action

URL: `https://api.markmonitor.com/cert/v1/order/{resourceId}/{action}` (PATCH)

When a user wants to make changes to an existing order based on its ID (resourceId), they can take advantage of the PATCH method by providing the action that they would want to have carried out.

Available actions are:

- cancel
- revoke
- reissue
- send DCV email
- validate domains
- update additional emails
- send certificate email

Response: The system returns modified order entity.

7.4.8.1. Cancellation of a certificate

Cancellation prevents a certificate from being issued or activated, ensuring that it stays inactive. Cancel a pending order for a new or renewed certificate can be done if the order status is CREATED, DIGI_PENDING, or DIGI_WAITING_PICKUP. To cancel a pending reissue, the order status must be DIGI_REISSUE_PENDING. Once an order has been issued, it cannot be canceled, instead, it must be revoked.

7.4.8.2. Revocation of a certificate

A certificate that has already been issued is revoked which prevents clients and web servers from accepting compromised or suspicious certificates. Revoking a certificate is permanent and cannot be undone.

7.4.8.3. Reissue of a certificate

Reissuing a certificate allows you to add, remove, or swap domain names, update the CSR, or change the signature hash of a certificate without submitting a new order. A reissued certificate has a new providerId but the same order providerId. After a reissue is approved, a new certificate is issued and needs to be reinstalled.

Only already issued orders (in status DIGI_ISSUED) can be reissued. If any of the values need to be changed in the new certificate they can be passed as part of the JSON body of the request. All new values are validated the same way as in the case of a new order creation. Note: the common name of a certificate can't change during the reissue process, but the user can alter the list of DNS names (SAN values).

7.4.8.4. Send DCV email for a certificate

A user can request for domain control validation (DCV) emails for a domain to be resent if the order's status is DIGI_PENDING or DIGI_REISSUE_PENDING. This is useful if the organization's contact did not receive the original DCV email.

7.4.8.5. Validate domains for a certificate

This endpoint is used to submit provided domains to be validated for a given validation type. The validation type is determined based on the cert type and can be OV or EV.

7.4.8.6. Update additional emails of a certificate

This action is used to update the additional emails associated with an order. Email addresses added to the additional email array receive notifications for renewals, reissues, and duplicates for the specified order. The user needs to pass a list of additional emails for this action to be completed.

7.4.8.7. Send an email with a certificate

User can take advantage of this endpoint to send a copy of an issued certificate to email addresses associated with contacts for a given order.

8. Things to know

8.1. Existing Markmonitor users

Suppose a user has previously created SSL certs entities in their accounts (organizations, contacts, orders) through Markmonitor Portal or with the help of their DPAs, they are not required to recreate these entities via the API. They can log in to the Markmonitor portal to obtain entities' IDs (resourceIds) or provider IDs and use described GET endpoints to query for corresponding entities. If a user is not able to find the required information, they can contact their DPA for help. This will ensure a smooth transition to the SSL Certs API.

8.2. Monitor organization validation and order statuses

Markmonitor will not inform a user about validation and order status changes during the process. A user is responsible for periodically initiating GET requests (on a specific organization and order) to monitor status changes. This way, the user will be able to be informed about the process of the certification issuance and the phase it is in at any point in time.

8.3. Keypair & Certificate Signing Request (CSR) Generation

To obtain an SSL certificate, providing a CSR is a necessary step. It is a uniquely formatted message produced by a server or other device. The CSR contains crucial details including the public key, domain name, and company information.

The following details commonly appear in the CSR:

- Common Name (CN): This is the FQDN, or fully qualified domain name, for which the certificate is sought.
- Organization (O): The official name of the company.
- Organizational Unit (OU): A division or other entity inside an organization.
- City (L), State (ST), and Country (C): Geographical details.
- Public Key: This is the public key that the server generated.

Users can generate CSRs in their preferred way. One of the common ways to generate CSR, especially in the tech industry, is via the OpenSSL library run in the command line.

The command to generate the CSR and KEY files for a domain *example.com* might look like this:

```
  openssl req -new -newkey rsa:2048 -nodes -out example_com.csr -keyout example_com.key -subj "/C=US/ST=Idaho /L=Meridian/O=Example Inc./OU=QA/CN=example.com"
```

8.4. SSL Certificate Installation

The installation of an SSL certificate depends on the web server software that users are running. Here's a general outline for different web servers:

For Apache, users typically need to edit their Apache server configuration file to specify the paths to their SSL certificate, private key, and intermediate certificate.
In Nginx, they'll configure the paths to their SSL certificate and private key in the Nginx server configuration.
If the user is using IIS on Windows Server, they can use the Internet Information Services (IIS) Manager to import the SSL certificate and bind it to their website.
Hosting control panels like cPanel or Plesk often offer the option to install SSL certificates directly from the control panel.

8.5. Renew Before Expiration

It is important to monitor the expiration date of any SSL certificate so that a new certificate is issued and installed before expiration. Failure to do so can result in users of your website receiving error messages in their browser when visiting your site and compromise the security and trust of your website.

To renew your SSL certificate, create a new order with an updated CSR, using the documented Order endpoints.

8.6. HTTP Return codes

There are several possible HTTP return codes that SSL certs API can return.

On successful execution of a request, a user gets *200 (OK)* for all requests, *201 (CREATED)* for organization and contact create (POST), and *202 (ACCEPTED)* for order create (POST) and "renew order" (PATCH) 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)*.

8.7. Path Variables and Query Parameters

8.7.1. '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 order/organization/contact 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.

8.7.2. 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://certs.markmonitor.com/v1/organization?providerId=4068
https://certs.markmonitor.com/api/v1/order?providerId=1858474&certificateTypes=SSL_EV_BASIC

https://certs.markmonitor.com/v1/organization?size=3&page=6