
KORM B2B Subscription Service - Yearly and Pay As You Go Subscriptions

Integration guide

Document version: 4.0
Document revision date: 06.04.2020

Revision history

Current revision

| Version | Date | Editor | Comments |
|---------|------------|-------------|---|
| 4.0 | 06.04.2020 | Natalya Kim | The document is approved by Sergey Romanov, Valery Ziborov, Maxim Telitsyn (INC000001532783). |

Previous revisions

Table of contents

| | | |
|--------|---|----|
| 1. | Introduction..... | 5 |
| 1.1. | Document purpose..... | 5 |
| 1.2. | Target audience..... | 5 |
| 1.3. | Definitions, acronyms, and abbreviations..... | 5 |
| 1.4. | Request for the document update..... | 6 |
| 2. | General description..... | 7 |
| 2.1. | Overview..... | 7 |
| 2.2. | Getting started..... | 7 |
| 2.3. | General technical description..... | 8 |
| 3. | Technical requirements..... | 11 |
| 3.1. | Test environment..... | 11 |
| 3.1.1. | User authentication and authorization..... | 11 |
| 3.1.2. | Reference..... | 11 |
| 3.1.3. | Endpoints..... | 11 |
| 3.2. | Production environment..... | 11 |
| 3.2.1. | User authentication and authorization..... | 11 |
| 3.2.2. | Reference..... | 12 |
| 3.2.3. | Endpoints..... | 12 |
| 3.3. | Data types and validation..... | 12 |
| 3.3.1. | General rules..... | 12 |
| 4. | API methods..... | 14 |
| 4.1. | Create..... | 14 |
| 4.1.1. | Description..... | 14 |
| 4.1.2. | Endpoints..... | 14 |
| 4.1.3. | Create request format..... | 14 |
| 4.1.4. | Create response format..... | 16 |
| 4.2. | HardCancel..... | 17 |
| 4.2.1. | Description..... | 17 |
| 4.2.2. | Endpoints..... | 17 |
| 4.2.3. | HardCancel request format..... | 17 |
| 4.2.4. | HardCancel response format..... | 17 |
| 4.3. | ModifyExpiration..... | 18 |
| 4.3.1. | Description..... | 18 |

| | | |
|--------|---|----|
| 4.3.2. | Endpoints..... | 18 |
| 4.3.3. | ModifyExpiration request format..... | 18 |
| 4.3.4. | ModifyExpiration response format | 20 |
| 4.4. | ModifyQuantity | 20 |
| 4.4.1. | Description..... | 20 |
| 4.4.2. | Endpoints..... | 21 |
| 4.4.3. | ModifyQuantity request format | 21 |
| 4.4.4. | ModifyQuantity response format..... | 22 |
| 4.5. | ModifyAttributes | 22 |
| 4.5.1. | Description..... | 22 |
| 4.5.2. | Endpoints..... | 22 |
| 4.5.3. | ModifyAttributes request format..... | 22 |
| 4.5.4. | ModifyAttributes response format..... | 23 |
| 4.6. | GetDetails | 24 |
| 4.6.1. | Description..... | 24 |
| 4.6.2. | Endpoints..... | 24 |
| 4.6.3. | GetDetails request format..... | 24 |
| 4.6.4. | GetDetails response format | 24 |
| 4.7. | GetUsage | 26 |
| 4.7.1. | Description..... | 26 |
| 4.7.2. | Endpoints..... | 26 |
| 4.7.3. | GetUsage request format | 26 |
| 4.7.4. | GetUsage response format..... | 26 |
| 5. | Appendices | 28 |
| 5.1. | Subscription lifecycle..... | 28 |
| 5.2. | The example of the GetUsage response..... | 28 |
| 5.3. | Error codes..... | 29 |

1. Introduction

1.1. Document purpose

This document provides guidelines on how to integrate a service or a system with the **KORM B2B Subscription Service** for yearly and Pay As You Go billing plans. The document provides information about the accepted protocols, request and response formats and authentication requirements.

1.2. Target audience

This document is for Distributors and developers of the **KORM B2B Subscription Service** client services.

1.3. Definitions, acronyms, and abbreviations

| | |
|--|---|
| KORM B2B Subscription Service | A platform that allows <i>Distributors</i> and <i>Resellers</i> to sell B2B subscriptions for their Resellers and Clients. B2B subscription is a more flexible way of selling Kaspersky Lab products compared to traditional license sales, in some cases being the only possible way of selling certain Kaspersky Lab solutions. |
| Activation code | Code that allows to activate (switch on) the purchased license in order to start using the Kaspersky Lab application. Activation codes are created by the KORM system as a result of purchase order processing. |
| Subscription | Product usage on limited or unlimited basis with specified parameters (expiration date, product, number of protected devices, etc.) requested by a Subscriber. |
| Yearly subscription | A subscription with yearly billing plan, which is used for subscription licenses for a product in B2B market sector. |
| Pay as you go (PAYG) subscription | A subscription with monthly billing plan and in which modification of quantity (increase or decrease) implements from the day of modification. Used for subscription licenses for a product in B2B market sector. |
| Billing period | <p>The interval of subscription, which is provided to a Subscriber. A key characteristic that determines the cost of a service. It corresponds to Term SKU in the subscription up to one day.</p> <p>For yearly billing plan – a calendar year (depending on the length of the calendar year, the duration can vary from 365 to 366 days).</p> <p>For PAYG billing plan – a calendar month (depending on the length of the calendar month, the duration can vary from 28 to 31 days). Exceptions for PAYG subscription where the period is incomplete and will be less than a month:</p> <ul style="list-style-type: none">• for the first paid period formed at the end of the trial period• for the period formed by hard cancellation |
| Chargeable period | The period of Subscription functionality, that will be charged. This may include one or several billing periods. |

| | |
|---------------------------------|--|
| Distributor | A direct (first-level) partner that provides Kaspersky Lab products to Resellers or customers on subscription basis. |
| Hard cancel | The operation used to cancel the subscription when the Subscriber has requested to stop the existing subscription. |
| License | A permit issued by Kaspersky Lab that allows Subscriber(s) to use a product installed on a protected device(s). |
| Limited subscription | A subscription with a defined end date. |
| Pre-payment | Pre-paid payment of the yearly subscription. This means that billing takes place in the coming invoice period. |
| Post-payment | Post-paid payment of the monthly and PAYG subscription. This means that billing takes place after the current billing period and in the coming invoice period. |
| Requester | Distributor who creates and manages subscriptions via KORM B2B Subscription Service API . |
| Reseller | A second-level partner that sells Kaspersky Lab products to customers. Examples: Classic Channel Resellers, MSPs, and Hosting Providers. |
| Reporting period | Periodicity of invoicing. |
| Stock keeping unit (SKU) | Unique identifier for each distinct item that can be purchased. |
| Subscription ID | A unique identifier of a subscription in KORM B2B Subscription Service . |
| Trial period | A free period that is offered to Subscribers before the chargeable subscription period begins. |
| Unlimited subscription | <p>A subscription with open end date. Subscription expiration date is either specified as NULL or is not specified.</p> <p>Currently, all yearly and PAYG subscriptions are unlimited at the moment of creation. For unlimited subscription, product is fully functional until Requester sends hard-cancel request (HardCancel method) or the subscription gets an Expired status (as a result of the ModifyExpiration method).</p> |

1.4. Request for the document update

To request the most recent version, contact KORM.Integrations@kaspersky.com.

2. General description

2.1. Overview

KORM B2B Subscription Service is a license management system that allows creating and managing orders for Kaspersky B2B product licenses on a yearly and monthly basis.



This document provides information on yearly and PAYG billing plans. For the information on monthly subscription, see [KORM Subscription Service Integration Guide for monthly subscriptions](#).

License management is performed by Requester through the integration with **KORM B2B Subscription Service** via **API**.

All delivery and management of licenses should be implemented solely by means of interaction between the **KORM B2B Subscription Service** and Requester's management system.

KORM B2B Subscription Service API allows the following:

- Placing a subscription order in Kaspersky Order Management System (**KORM**).
- Flexible management of the subscription:
 - Increasing and decreasing the number of licenses (protected devices) per subscription on a yearly or monthly basis. Decrease in **Yearly** subscription will only take effect in the next billed year.



Cancelled subscription cannot be restarted.

- Distributing the licenses on pre-payment terms (between KL and Requester).



*For more information on **KORM B2B Subscription Service API** capabilities, see [section 2.3, General technical description](#).*

The Activation code is permanent and tied to the subscription. The Activation code remains the same even after modifying the number of protected devices within subscription order or after cancelling a subscription.

2.2. Getting started

To start sales based on the subscription model, Requester needs to integrate with Kaspersky Order Management (**KORM**).

To integrate with **KORM B2B Subscription Service API**, you can do one of the following:

- Develop full integration with your current corporate order management system using APIs listed in this document.
- Create simple web service (based on the examples specified in [section 3, Technical requirements](#)) and complete it by any web-based technology user interface.
- Use the services of an external System Integrator to develop the variant of integration that is suitable for you.
- Use an order management platform already developed and maintained by a 3rd party. For example, some of these platforms will allow Distributors not only place orders, but to automate interaction with Resellers.

2.3. General technical description

KORM B2B Subscription Service is a technical backend that allows selling products basing on a subscription sales model.

The subscriber interacts with their Distributor/Reseller to manage the subscription. The Distributor/Reseller sends requests to Kaspersky Lab via the **KORM B2B Subscription Service API**. The Request is processed via KL services and the response (which will include the Subscription ID and Activation code) is sent back to the requester. At the end of the **Reporting** period (the nearest invoicing period), KL invoices Distributor for all subscriptions that were active in the **Reporting** period (from the 1st day of the month till the last day of the same month inclusively), except subscriptions that were still in the trial period.

Only **Unlimited** (subscription expiration date is either specified as NULL or is not specified) subscription type is available.

The lifecycle of buying subscription is as follows (see [Figure 1](#)):

1. Subscriber raises request directly to Distributor or through the Reseller to purchase product license.
2. Distributor creates a new subscription by sending request to KL via the **KORM B2B Subscription Service**.
3. The **KORM B2B Subscription Service** then:
 - a. Sends the Activation code and Subscription ID to Distributor.
 - b. Uploads the Activation code into KL Activation services.
4. Distributor delivers the Activation code to the Subscriber/Reseller.
5. Subscriber activates the product with the Activation code provided.



- *The Activation code can be activated by Subscriber immediately upon subscription creation.*
- *SLA for providing the Activation code to Distributor is 30 seconds. When configuring the timeout interval on the client, make sure that the interval is more than this value.*
- *SLA for modifying subscription parameters is 24 hours.*

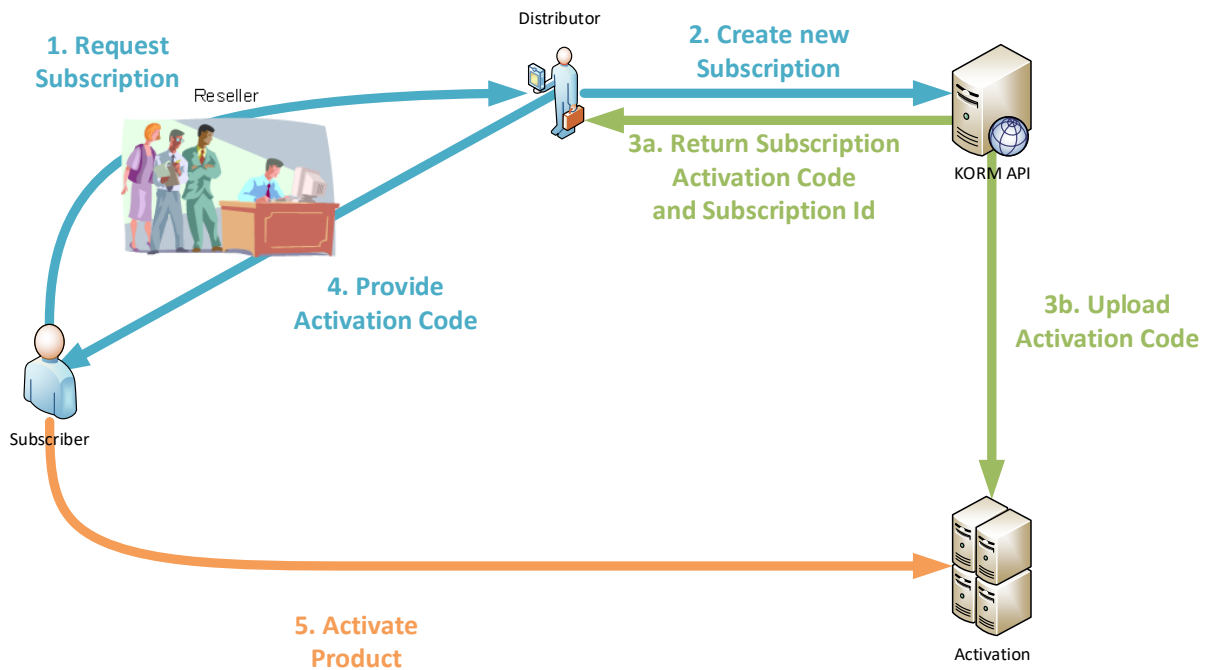


Figure 1. Interaction between KL and Distributor via KORM Subscription Service

The main methods that change subscription state are displayed in [Figure 2](#):

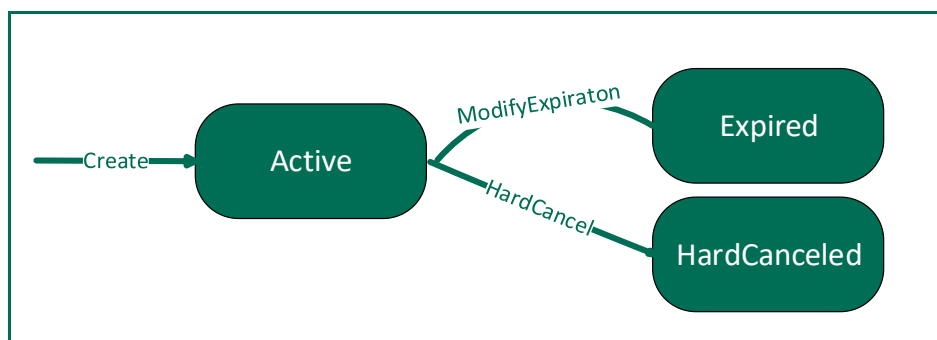


Figure 2. Subscription states

As shown on the diagram, a subscription cannot be renewed from any inactive state (**HardCanceled** or **Expired**). After the subscription becomes inactive, to remain subscribed you will have to create a new subscription that will contain a new Activation code.

Full subscription lifecycle is displayed in [Figure 5](#) (see section [5.1, Subscription lifecycle](#)). It shows available methods for each subscription period (**Trial** and **Chargeable**) and changed subscription states that result from invoking a method.

This guide describes interaction of Requester's client applications with KL via **KORM B2B Subscription Service API** methods, which are published as Internet web services. API methods allow Distributor to execute the following business cases:

| Method name | Business case | The method can be invoked for subscription states | After invoking the method, subscription state changes to |
|----------------------------|----------------------------------|---|--|
| Create | Create product subscription | No state | Active |
| HardCancel | Immediate cancel of subscription | Active | HardCanceled |

| Method name | Business case | The method can be invoked for subscription states | After invoking the method, subscription state changes to |
|----------------------------------|--|---|---|
| ModifyExpiration | Turn off auto-renewal or turn on auto-renewal | Active | Active (until the Expiration Date, then – Expired) for turning off auto-renewal Active for turning on auto-renewal |
| ModifyQuantity | Modify the quantity of protected devices | Active | Active |
| ModifyAttributes | Modify external links, end-user data, email for sending artifacts, affiliate discount code and Approval code in the subscription | Active | Active |
| GetDetails | Get subscription details about current subscription period | All states | All states |
| GetUsage | Get extended usage details | All states | All states |

3. Technical requirements

3.1. Test environment

3.1.1. User authentication and authorization

To integrate with **KORM B2B Subscription Service** via API for demo purposes, the Requester should contact an Account Manager at KL. The Account Manager is responsible for the following:

1. Registering the Distributor account.
2. Granting permissions.
3. Supplying root SSL certificate(s) if needed.
4. Generating and providing a client SSL certificate for demo purposes.

An email with the client SSL certificate download link and installation instructions is sent to the Distributor automatically when the account is registered. After installing the certificate, the Distributor can use the **KORM B2B Subscription Service API** (for test environment endpoints, see section [3.1.3, Endpoints](#)).

3.1.2. Reference

Additional **KORM B2B Subscription Service** reference is available at <https://api.korm.kaspersky.com/Subscriptions/v1.0/Index.html> . The page contains the following information:



To access the specified page, you must install a client SSL certificate. For more information, see section [3.1.1, User authentication and authorization](#).

- Brief service description
- Version history
- REST interface description:
 - Endpoints
 - JSON examples
- Useful links

3.1.3. Endpoints

| Interface | Test endpoint |
|-----------|---|
| REST | https://api.demo.korm.kaspersky.com/Subscriptions/v2.0/ |

3.2. Production environment

3.2.1. User authentication and authorization

To integrate with **KORM B2B Subscription Service API** on production, Requester should contact an Account Manager at KL. The account manager is responsible for the following:

1. Registering the Distributor account.
2. Granting permissions.
3. Generating and providing a client SSL certificate for production purposes.

An email with the client SSL certificate download link and installation instructions is sent to the Distributor automatically when the account is registered. After installing the certificate, the

Distributor can use the **KORM B2B Subscription Service API** (for production environment endpoints, see section [3.2.3, Endpoints](#)).

3.2.2. Reference

Additional **KORM B2B Subscription Service** reference is available at [<endpoint>](#). The page contains the following information:



To access the specified page, you must install a client SSL certificate. For more information, see section [3.1.1, User authentication and authorization](#).

- Brief service description
- Version history
- REST interface description:
 - Endpoints
 - JSON examples
- Useful links

3.2.3. Endpoints

| Interface | Production endpoint |
|-----------|---|
| REST | https://api.korm.kaspersky.com/Subscriptions/v2.0/ |

3.3. Data types and validation

3.3.1. General rules

3.3.1.1. Format description

Below is an example of a request description in **KORM B2B Subscription Service**.

| # | Parameter | Data type | Description | R/O |
|-----|-------------------|-----------|--|-----|
| 1 | ExampleParameter1 | String | Description of the ExampleParameter1. | O |
| 1.1 | ExampleParameter2 | String | Description of the ExampleParameter2. Attention! This parameter is required only if the ExampleParametr1 is used. | R |
| 2 | ExampleParameter3 | Int | Description of the ExampleParameter3. | R |

Each parameter of a request is described by the following column values:

- **#** — sequential number of the parameter. Compound parameters are highlighted in grey.
- **Parameter** — name of the parameter.
- **Data type** — base class of the parameter data.



- **Data type** is not specified for compound parameters.
- For strings (**Data type** = **String**), UTF-16 Unicode format is used. Every symbol is allowed. Maximum length and additional rules, if any, are specified in the **Description** column (see below).

- **Description** — full description of the parameter. Additional rules that are not defined by **Data type** may also be detailed here if appropriate. If the specified limitations are violated, then the **KORM B2B Subscription Service** will respond with an error.

- **R/O** —whether the parameter is one of the following:
 - **R** — required parameter. Sending a request without this parameter will always result in an error.
 - **O** — optional parameter. This parameter can be omitted.



- *Despite being listed as **Optional**, a parameter can still be required under some conditions. If such conditions exist, then they are described in the **Description** column.*
- *If the main block (like 1) is **Optional**, then all sub-parameters (1.1, 1.2...) are **Optional** too, even if they are marked as “R”. They are required only if you use the whole main block.*

3.3.1.2. Data validation

Each request is validated by the **KORM B2B Subscription Service**. When receiving any incorrect data, the service responds with an error. No additional validation on the Requestor's side is necessary.

4. API methods

4.1. Create

4.1.1. Description

The Requester invokes the **Create** method to create licenses that allow protecting several devices on a subscription basis.

When creating the subscription, Requester specifies billing plan (currently **Yearly** and **PAYG** are supported).

For **Yearly** and **PAYG** billing plans, after creating the subscription, a **Trial** period is provided to the Subscriber for the specified period (according to settings).

4.1.2. Endpoints

| Interface | Production endpoint | Test endpoint |
|-----------|---|---|
| REST | https://api.korm.kaspersky.com/Subscriptions/v2.0/api/Subscription/create HTTP method: POST | https://api.demo.korm.kaspersky.com/Subscriptions/v2.0/api/Subscription/create HTTP method: POST |

4.1.3. Create request format

The table below describes the set of parameters that need to be specified to invoke **Create** method. In case of an incorrect request, **KORM** will respond with a corresponding error message.

| # | Parameter | Data type | Description | R/O |
|-------|--------------|-----------|---|-----|
| 1 | BillingPlan | String | The period for which Subscriber is charged for using a KL product on subscription basis. Currently accepted values: <ul style="list-style-type: none"> Yearly PAYG | R |
| 2 | Sku | String | Stock keeping unit (sale item) in KL price list. Attention! Only one SKU can be requested. | R |
| 3 | Quantity | Integer | Quantity of protected devices. Attention! The value must match the SKU band. | R |
| 4 | Customer | | Customer information | R |
| 4.1 | Contacts | | Customer contacts | R |
| 4.1.1 | CompanyName | String | Subscriber company name. | R |
| 4.1.2 | Email | String | Email | O |
| 4.1.3 | Phone | String | Phone number | O |
| 4.1.4 | CustomerCode | String | Subscriber code | O |
| 4.2 | Address | String | Subscriber address | R |
| 4.2.1 | AddressLine1 | String | Address line 1 | O |
| 4.2.2 | AddressLine2 | String | Address line 2 | O |
| 4.2.3 | City | String | City | O |
| 4.2.4 | State | String | State | O |
| 4.2.5 | Zip | String | Zip code | O |

| # | Parameter | Data type | Description | R/O |
|--------|------------------------|-----------|--|-----|
| 4.2.6 | Country | String | Country (ISO 3166-1 Alpha 3 code); Length – 3 symbols | R |
| 5 | Distributor | | Distributor information | R |
| 5.1 | Partner | String | Partner code. Max length – 10 symbols. | R |
| 5.2 | Reseller | String | Reseller PIN. Max length – 10 symbols. Note: for testing purposes, the TE27PT00 Reseller PIN can be used. Attention! The field can be either Optional or Required, depending on the Partner's settings. Reseller details are expected to be provided before the first invoice by the use of ModifyAttributes method. | R/O |
| 6 | ExternalReference | | References in Distributor's system | O |
| 6.1 | ExternalSubscriptionId | String | Subscription identifier in Distributor's system | O |
| 6.2 | ExternalOrderId | String | Order ID in Distributor's system | O |
| 6.3 | ExternalLineItemId | String | Position ID in Distributor's system | O |
| 7 | Comment | String | Any information. Max length – 255 symbols. | O |
| 8 | ApprovalCode | String | Identifier of the Price special offer. Maximum length – 50 symbols. | O |
| 9 | DeliveryEmail | String | Email to which license certificate will be delivered | R |
| 10 | TermsAndConditions | | Information about data processing agreements between the customer and the partner. If sent, this block must contain at least one customer agreement. | O |
| 10.1 | CustomerAgreements | Array | Customer agreements that contain an acceptancy flag and an agreement text each. | R |
| 10.1.1 | AgreementAccepted | Boolean | Flag that indicates whether the customer accepted the agreement text. | R |
| 10.1.2 | AgreementText | String | Agreement text. | O |
| 10.1.3 | AgreementTextHash | String | Hash of the agreement text. If both the AgreementText and AgreementTextHash parameters are sent, then the AgreementTextHash parameter is ignored by KORM . | O |
| 11 | AffiliateDiscountCode | String | Code by which KL partner receives an additional discount to cover the costs of the services provided by affiliates. Maximum length – 50 symbols. | O |
| 12 | Expiration | | Subscription expiration moment. Attention! Not applicable for the Yearly and PAYG billing plans. Scheduled in later re | O |

| # | Parameter | Data type | Description | R/O |
|------|-------------|----------------------|---|-----|
| 12.1 | MomentType | Enum | <p>Possible values:</p> <ul style="list-style-type: none"> • ExactMoment • ByBillingPeriods • NearestPossible <p>For ExactMoment, subscription Expiration Date is the date and time in UTC that are specified in the ExactMoment parameter.</p> <p>For ByBillingPeriods, it is also required to specify the number of periods in the PeriodCount parameter. To stop auto-renewal after the current period, specify 1 in the PeriodCount parameter. If ByBillingPeriods is specified, then subscription Expiration Date is the billing period end date.</p> <p>For NearestPossible, subscription auto-renew will stop in the end of the billing period after the date and time that are specified in the AfterMoment parameter. If the AfterMoment parameter is not specified, then NearestPossible is the end of the current period.</p> | R |
| 12.2 | ExactMoment | Datetime ISO-8601 | Exact expiration moment of the subscription. Valid and required only if MomentType = ExactMoment . | O |
| 12.3 | AfterMoment | Datetime ISO-8601 | Moment after which the subscription will be expired in the end of the period. Valid only if MomentType = NearestPossible . | O |
| 12.4 | PeriodCount | Integer | Count of periods while the subscription will be auto-renewed. Valid and required only if MomentType = ByBillingPeriods . | O |

4.1.4. Create response format

The table below describes the parameters of **Create** response in case of a correct request and successful processing in **KORM**.

In case of a correct request and successful processing, **KORM** will respond with **HTTP status 200**.

In case of an incorrect request, **KORM** will respond with a corresponding error message. The response is returned in synchronous mode.

| # | Parameter | Data type | Description | R/O |
|---|----------------|-----------|---|-----|
| 1 | SubscriptionId | String | Unique subscription identifier. A new SubscriptionId is created by the KORM Subscription Service each time the Create method is called successfully. Max length – 50 symbols. Attention! Make sure to store SubscriptionId . It will be used for any further operations with the subscription. It also should be visible to the Requester's support and sales staff for troubleshooting purposes. | R |
| 2 | LicenceId | String | Unique identifier of the license. | R |
| 3 | ActivationCode | String | Activation code | O |



Is this instruction unclear or incorrect? [Let us know and we will improve it!](#)

4.2. HardCancel

4.2.1. Description

The **HardCancel** method is used to cancel the subscription.

Yearly and PAYG subscriptions are canceled immediately.

Yearly subscription will be charged for the whole billing period. PAYG subscription will be charged from the beginning of the billing period to the cancellation.

After a **HardCancel** method is processed successfully, the subscription state becomes **HardCanceled**.

4.2.2. Endpoints

| Interface | Production endpoint | Test endpoint |
|-----------|---|---|
| REST | https://api.korm.kaspersky.com/Subscriptions/v2.0/api/Subscription/hardcancel HTTP method: POST | https://api.demo.korm.kaspersky.com/Subscriptions/v2.0/api/Subscription/hardcancel HTTP method: POST |

4.2.3. HardCancel request format

The table below describes the parameter that need to be specified to invoke **HardCancel** method. In case of an incorrect request, **KORM** will respond with a corresponding error message.

| # | Parameter | Data type | Description | R/O |
|---|----------------|-----------|---|-----|
| 1 | SubscriptionId | String | Unique subscription identifier. Max length – 50 symbols. | R |

4.2.4. HardCancel response format

In case of a correct request and successful processing, **KORM** will respond with `HTTP status 200`.

In case of an incorrect request, **KORM** will respond with a corresponding error message.



Is this instruction unclear or incorrect? [Let us know and we will improve it!](#)

4.3. ModifyExpiration

4.3.1. Description

The **ModifyExpiration** method is used to stop auto-renewal of a subscription or to restore auto-renewal of orders.

When user stops auto-renewal of a subscription, the license is valid until the end of the current billing period. The license and subscription have an Expiration Date equal to the end date of the current billing period. At the time of expiration, the subscription status becomes **Expired**. If subscription was *Unlimited*, it becomes *Limited*.

When user restores auto-renewal of orders, the expiration date of the subscription and license will be cleared.



*User cannot modify expiration date of hard-canceled or expired subscription. It is available in the subscription's **Active** state only.*

4.3.2. Endpoints

| Interface | Production endpoint | Test endpoint |
|-----------|---|---|
| REST | https://api.korm.kaspersky.com/Subscriptions/v2.0/api/Subscription/modifyexpiration HTTP method: POST | https://api.demo.korm.kaspersky.com/Subscriptions/v2.0/api/Subscription/modifyexpiration HTTP method: POST |

4.3.3. ModifyExpiration request format

The table below describes the set of parameters that need to be specified to invoke the **ModifyExpiration** method. In case of an incorrect request, **KORM** will respond with a corresponding error message.

| # | Parameter | Data type | Description | R/O |
|---|----------------|-----------|--|-----|
| 1 | SubscriptionId | String | Unique subscription identifier. Max length – 50 symbols. | R |
| 2 | Expiration | | Subscription expiration moment. Attention! For the Yearly and PAYG billing plans, Expiration should not be specified to restore auto-renew. | O |

| # | Parameter | Data type | Description | R/O |
|-----|-------------|----------------------|---|-----|
| 2.1 | MomentType | Enum | <p>Possible values:</p> <ul style="list-style-type: none"> • ExactMoment • ByBillingPeriods • NearestPossible <p>Attention! For the Yearly and PAYG billing plans, user should wait for the end of the current period to cancel auto-renew.</p> <p>For ExactMoment, subscription Expiration Date is the date and time in UTC that are specified in the ExactMoment parameter.</p> <p>For ByBillingPeriods, it is also required to specify the number of periods (current is not included) in the PeriodCount parameter. To stop auto-renewal after the current period, specify 0 in the PeriodCount parameter. If ByBillingPeriods is specified, then subscription Expiration Date is the billing period end date.</p> <p>For NearestPossible, subscription auto-renew will stop in the end of the billing period after the date and time that are specified in the AfterMoment parameter. If the AfterMoment parameter is not specified, then NearestPossible is the end of the current period.</p> <p>If MomentType is not specified, then subscription Expiration Date is NULL.</p> | R |
| 2.2 | ExactMoment | Datetime ISO-8601 | Exact expiration moment of the subscription. Valid and required only if MomentType = ExactMoment . | O |
| 2.3 | AfterMoment | Datetime ISO-8601 | Moment after which the subscription will be expired in the end of the period. Valid only if MomentType = NearestPossible . | O |
| 2.4 | PeriodCount | Integer | Count of periods while the subscription will be auto-renewed. Valid and required only if MomentType = ByBillingPeriods . | O |

4.3.4. ModifyExpiration response format

In case of a correct request and successful processing, **KORM** will respond with `HTTP status 200`.

In case of an incorrect request, **KORM** will respond with a corresponding error message.



Is this instruction unclear or incorrect? [Let us know and we will improve it!](#)

4.4. ModifyQuantity

4.4.1. Description

The **ModifyQuantity** method is used to modify (i.e. increase or decrease) the number of protected devices within the subscription without changing the Activation code. There is no need to reactivate the previously activated product.



*SKU may be changed automatically as a result of modifying the quantity of protected devices. To verify, invoke the **GetDetails** method (see section 4.6, [GetDetails](#)).*

For the details on the method realization, see the diagram below:

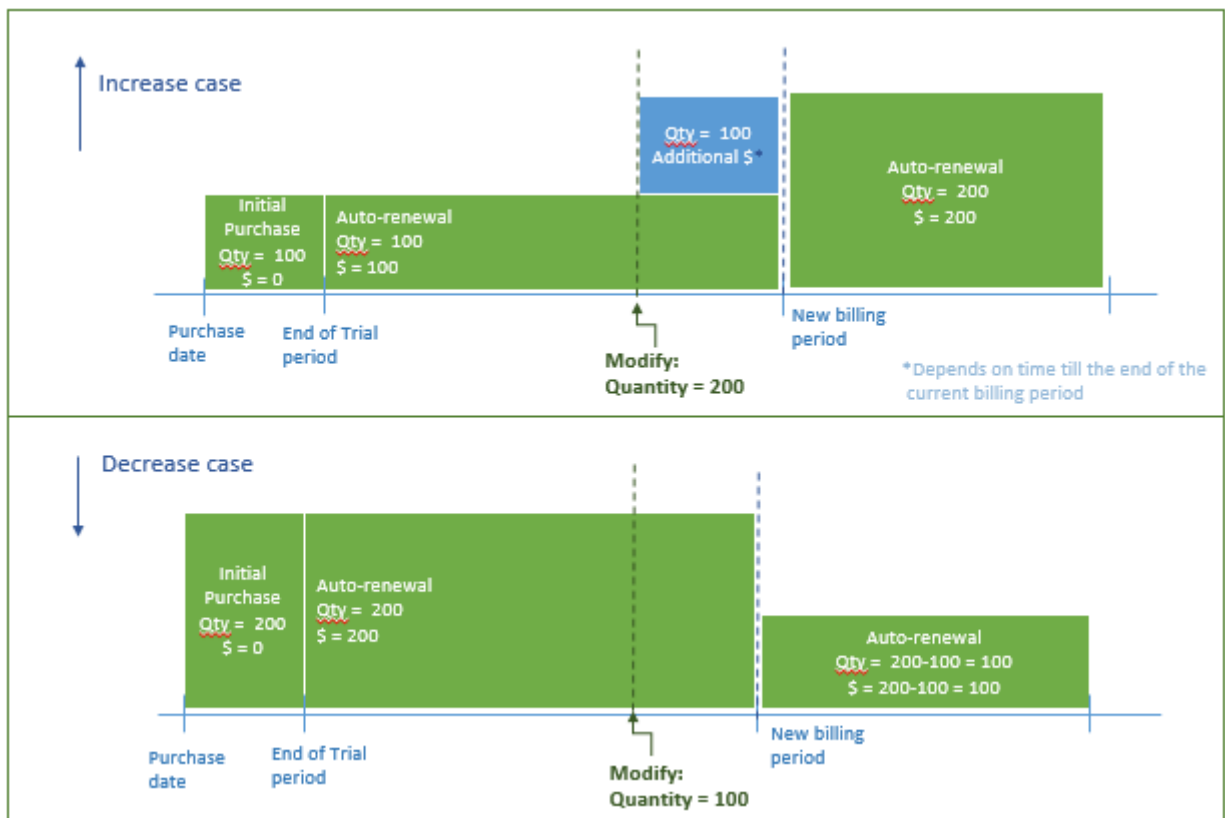


Figure 3. Increase or decrease of the quantity of protected devices within a billing period (Yearly)

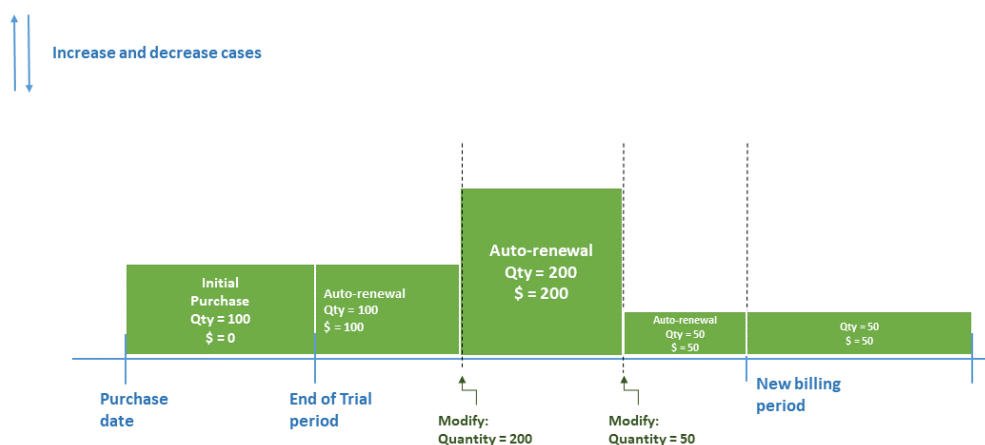


Figure 4. Increase or decrease of the quantity of protected devices within a billing period (PAYG)



For yearly subscription:

- When increasing the number of protected devices, the license will be changed immediately after the **ModifyQuantity** method is processed (i.e. as soon as a successful response is sent). The subscription will be charged for the increased number of protected devices from the date of modification until the end of the billing period.
- When decreasing the number of protected devices, the license will be changed starting from the next billing period (year). Up until then, the number of protected devices (i.e. the maximum number of activations) will remain unchanged. The subscription for the decreased number of protected devices will be charged starting from the next billing period, if there were no other modify quantity requests.

For PAYG subscription:

- When increasing or decreasing the number of protected devices, the license will be changed immediately after the **ModifyQuantity** method is processed (i.e. as soon as a successful response is sent). The subscription will be charged for the increased or decreased number of protected devices from the date of modification until the end of the billing period or until another modification.
- If there were several quantity changes in one day, then subscription will be charged for the last change on this day by UTC.

4.4.2. Endpoints

| Interface | Production endpoint | Test endpoint |
|-----------|---|---|
| REST | https://api.korm.kaspersky.com/Subscriptions/v2.0/api/Subscription/modifyquantity HTTP method: POST | https://api.demo.korm.kaspersky.com/Subscriptions/v2.0/api/Subscription/modifyquantity HTTP method: POST |

4.4.3. ModifyQuantity request format

The table below describes the set of parameters that need to be specified to invoke **ModifyQuantity** method. In case of an incorrect request, **KORM** will respond with a corresponding error message.

| # | Parameter | Data type | Description | R/O |
|---|----------------|-----------|--|-----|
| 1 | SubscriptionId | String | Unique subscription identifier. Max length – 50 symbols. | R |
| 2 | Quantity | Int | The number of protected objects that need to be protected within the subscription. | R |

4.4.4. ModifyQuantity response format

In case of a correct request and successful processing, **KORM** will respond with `HTTP status 200`.

In case of an incorrect request, **KORM** will respond with a corresponding error message.



Is this instruction unclear or incorrect? [Let us know and we will improve it!](#)

4.5. ModifyAttributes

4.5.1. Description

The **ModifyAttributes** method is used to modify external links, end-user data, email for sending artifacts, affiliate discount code and Approval code in the subscription.



All changes will be applied starting from the next orders.

4.5.2. Endpoints

| Interface | Production endpoint | Test endpoint |
|-----------|---|---|
| REST | https://api.korm.kaspersky.com/Subscriptions/v2.0/api/Subscription/modifyattributes HTTP method: POST | https://api.demo.korm.kaspersky.com/Subscriptions/v2.0/api/Subscription/modifyattributes HTTP method: POST |

4.5.3. ModifyAttributes request format

The table below describes the set of parameters that need to be specified to invoke **ModifyAttributes** method. In case of an incorrect request, **KORM** will respond with a corresponding error message.

| # | Parameter | Data type | Description | R/O |
|-------|----------------|-----------|--|-----|
| 1 | SubscriptionId | String | Unique subscription identifier. Max length – 50 symbols. | R |
| 2 | Customer | | Customer information | R |
| 2.1 | Contacts | | Customer contacts | R |
| 2.1.1 | CompanyName | String | Subscriber company name | R |
| 2.1.2 | CompanyName | String | Subscriber company name | R |
| 2.1.3 | Email | String | Email | O |
| 2.1.4 | Phone | String | Phone number | O |
| 2.1.5 | CustomerCode | String | Subscriber code | O |
| 2.2 | Address | | Subscriber address | R |
| 2.2.1 | AddressLine1 | String | Address line 1 | O |
| 2.2.2 | AddressLine2 | String | Address line 2 | O |
| 2.2.3 | City | String | City | O |
| 2.2.4 | State | String | State | O |
| 2.2.5 | Zip | String | Zip code | O |
| 2.2.6 | Country | String | Country (ISO 3166-1 Alpha 3 code); Length – 3 symbols | R |

| # | Parameter | Data type | Description | R/O |
|-----|------------------------|-----------|--|------|
| 3 | ExternalReference | | References in Distributor's system | O |
| 3.1 | ExternalSubscriptionId | String | Subscription identifier in Distributor's system | O |
| 3.2 | ExternalOrderId | String | Order ID in Distributor's system | O |
| 3.3 | ExternalLineItemId | String | Position ID in Distributor's system | O |
| 4 | AffiliateDiscountCode | String | Code by which KL partner receives an additional discount to cover the costs of the services provided by affiliates. Maximum length – 50 symbols. | O |
| 5 | Distributor | | Distributor information Attention! Not applicable for the Yearly and PAYG billing plans. | O |
| 5.1 | Partner | String | Partner code. Provided by KL account manager after registering the account at KL Partner Portal. Max length – 10 symbols. | R |
| 5.2 | Reseller | String | Reseller PIN. Max length – 10 symbols. Note: for testing purposes, the TE27PT00 Reseller PIN can be used. Attention! The field can be either Optional or Required, depending on the Provider's settings. | R/O* |
| 6 | DeliveryEmail | String | Email to which order artifacts will be delivered | R |
| 7 | ApprovalCode | String | Identifier of the Price special offer. Max length – 50 symbols. Attention! <ul style="list-style-type: none"> If there is an ApprovalCode in the subscription at the time of the ModifyAttributes request, and the user does not indicate the value of the ApprovalCode in the request, then an error occurs. If the ApprovalCode in the subscription does not match the value of the ApprovalCode indicated by user (except NULL values), then an error occurs. | O |

4.5.4. ModifyAttributes response format

In case of a correct request and successful processing, **KORM** will respond with HTTP status 200.

In case of an incorrect request, **KORM** will respond with a corresponding error message.



Is this instruction unclear or incorrect? [Let us know and we will improve it!](#)

4.6. GetDetails

4.6.1. Description

The **GetDetails** method is used to get detailed information about the subscription status and attributes in the current billing period.

4.6.2. Endpoints

| Interface | Production endpoint | Test endpoint |
|-----------|--|--|
| REST | https://api.korm.kaspersky.com/Subscriptions/v2.0/api/Subscription/getdetails?SubscriptionId=Value HTTP method: GET | https://api.demo.korm.kaspersky.com/Subscriptions/v2.0/api/Subscription/getdetails?SubscriptionId=Value HTTP method: GET |

4.6.3. GetDetails request format

The table below describes the set of parameters that need to be specified to invoke **GetDetails** method. In case of an incorrect request, **KORM** will respond with a corresponding error message.

| # | Parameter | Data type | Description | R/O |
|---|----------------|-----------|---|-----|
| 1 | SubscriptionId | String | Unique subscription identifier. Max length – 50 symbols. | R |

4.6.4. GetDetails response format

The table below describes the parameters of **GetDetails** response in synchronous mode in case of a correct request and successful processing in **KORM**.

| # | Parameter | Data type | Description | R/O |
|---------|-----------------|----------------------|--|-----|
| 1 | Details | | Subscription details | R |
| 1.1 | Status | Enum | Subscription status. Possible values: <ul style="list-style-type: none"> Active HardCanceled Expired | R |
| 1.2 | ActivationCode | String | Activation code | O |
| 1.3 | CurrentQuantity | Integer | Quantity of protected devices | R |
| 1.4 | CurrentSKU | String | Stock keeping unit (product item) in KL price list. | R |
| 1.5 | BillingPlan | String | The recurring billing period for which a Subscriber will be charged for using a KL product on subscription basis. Possible values: <ul style="list-style-type: none"> Yearly PAYG | R |
| 1.6 | ExpirationDate | DateTime ISO-8601 | Subscription expiration date. | O |
| 1.7 | Customer | | Customer information | R |
| 1.7.1 | Contacts | | Customer contacts | R |
| 1.7.1.1 | CompanyName | String | Subscriber company name | R |

| # | Parameter | Data type | Description | R/O |
|---------|------------------------|----------------------|--|-----|
| 1.7.1.2 | Email | String | Email | O |
| 1.7.1.3 | Phone | String | Phone number | O |
| 1.7.1.4 | CustomerCode | String | Subscriber code | O |
| 1.7.2 | Address | | Subscriber address | R |
| 1.7.2.1 | AddressLine1 | String | Address line 1 | O |
| 1.7.2.2 | AddressLine2 | String | Address line 2 | O |
| 1.7.2.3 | City | String | City | O |
| 1.7.2.4 | State | String | State | O |
| 1.7.2.5 | Zip | String | Zip code | O |
| 1.7.2.6 | Country | String | Country (ISO 3166-1 Alpha 3 code); Length – 3 symbols | R |
| 1.8 | Distributor | | Distributor information | R |
| 1.8.1 | Partner | String | Partner code. Max length – 10 symbols | R |
| 1.8.2 | Reseller | String | Reseller PIN. Max length – 10 symbols. Note: for testing purposes, the TE27PT00 Reseller PIN can be used. | R/O |
| 1.9 | ExternalReference | | References in Distributor's system | O |
| 1.9.1 | ExternalSubscriptionId | String | Subscription identifier in Distributor's system | O |
| 1.9.2 | ExternalOrderId | String | Order ID in Distributor's system | O |
| 1.9.3 | ExternalLineItemId | String | Position ID in Distributor's system | O |
| 1.10 | ApprovalCode | String | Identifier of the Price special offer. | O |
| 1.11 | CreatedDate | DateTime ISO-8601 | Subscription creation date. | R |
| 1.12 | AffiliateDiscountCode | String | Code by which KL partner receives an additional discount to cover the costs of the services provided by affiliates. Maximum length – 50 symbols. | O |
| 1.13 | PeriodType | Enum | Type of billing period. Possible values: <ul style="list-style-type: none"> • Free • Paid. For HardCanceled and Expired subscription will be empty. | O |
| 1.14 | PeriodStart | DateTime ISO-8601 | Start date of the current period. For HardCanceled and Expired subscription will be empty. | O |
| 1.15 | PeriodEnd | DateTime ISO-8601 | End date of the current period. For HardCanceled and Expired subscription will be empty. | O |
| 1.16 | DeliveryEmail | String | Email to which subscription artifacts will be delivered. | R |
| 1.17 | LicensedId | String | Unique identifier of the license. | R |



Is this instruction unclear or incorrect? [Let us know and we will improve it!](#)

4.7. GetUsage

4.7.1. Description

The **GetUsage** method provides extended information about the usage of yearly and PAYG subscriptions in any status in the subscription period - quantity of protected devices, type of a period (trial or paid) and start and end dates.



The Information about future dates is approximate and can change.

Information may be requested for the following periods:

- For all periods.
- Current and future.
- For previous, current and future.

4.7.2. Endpoints

| Interface | Production endpoint | Test endpoint |
|-----------|--|--|
| REST | https://api.korm.kaspersky.com/Subscriptions/v2.0/api/Subscription/getusage?SubscriptionId=Value1&RequiredPeriods=Value2 HTTP method: GET | https://api.demo.korm.kaspersky.com/Subscriptions/v2.0/api/Subscription/getusage?SubscriptionId=Value1&RequiredPeriods=Value2 HTTP method: GET |

4.7.3. GetUsage request format

The table below describes the set of parameters that need to be specified to invoke **GetUsage** method. In case of an incorrect request, **KORM** will respond with a corresponding error message.

| # | Parameter | Data type | Description | R/O |
|---|-----------------|-----------|--|-----|
| 1 | SubscriptionId | String | Unique subscription identifier. Max length – 50 symbols. | R |
| 2 | RequiredPeriods | Enum | Required periods. Possible values: <ul style="list-style-type: none"> • All (All periods will be returned) • CurrentAndFuture (Current and one future period will be returned) • PreviousAndFuture (Previous before current, current and one future period will be returned). | R |

4.7.4. GetUsage response format

The table below describes the parameters of **GetUsage** response in synchronous mode in case of a correct request and successful processing in **KORM**.

| # | Parameter | Data type | Description |
|-------|--------------|----------------------|---|
| 1 | UsageDetails | | Collection of billing periods. |
| 1.1 | Period | | Subscription period details. |
| 1.1.1 | Id | Integer | Unique identifier of the period. |
| 1.1.2 | PeriodType | String | Type of billing period. Possible values: <ul style="list-style-type: none"> • Free • Paid. |
| 1.1.3 | Start | DateTime ISO-8601 | Billing period start date. |

| # | Parameter | Data type | Description |
|---------|--------------|----------------------|---|
| 1.1.4 | End | DateTime ISO-8601 | Billing period end date. |
| 1.1.5 | usagePeriods | Array | |
| 1.1.5.1 | Start | DateTime ISO-8601 | Start date of the quantity usage. |
| 1.1.5.2 | End | DateTime ISO-8601 | End date of the quantity usage. |
| 1.1.5.3 | Quantity | Integer | Quantity activated during the specified period (May be Quantity, CurrentQuantity or other). |
| 1.1.6 | usagePeriods | | |
| 1.1.6.1 | ... | ... | ... |
| 1.2 | Period | | Billing period details. |
| 1.2.1 | ... | ... | ... |



Is this instruction unclear or incorrect? [Let us know and we will improve it!](#)

5. Appendices

5.1. Subscription lifecycle

The diagram below illustrates the stages of the subscription lifecycle and shows the methods that are available during each stage.

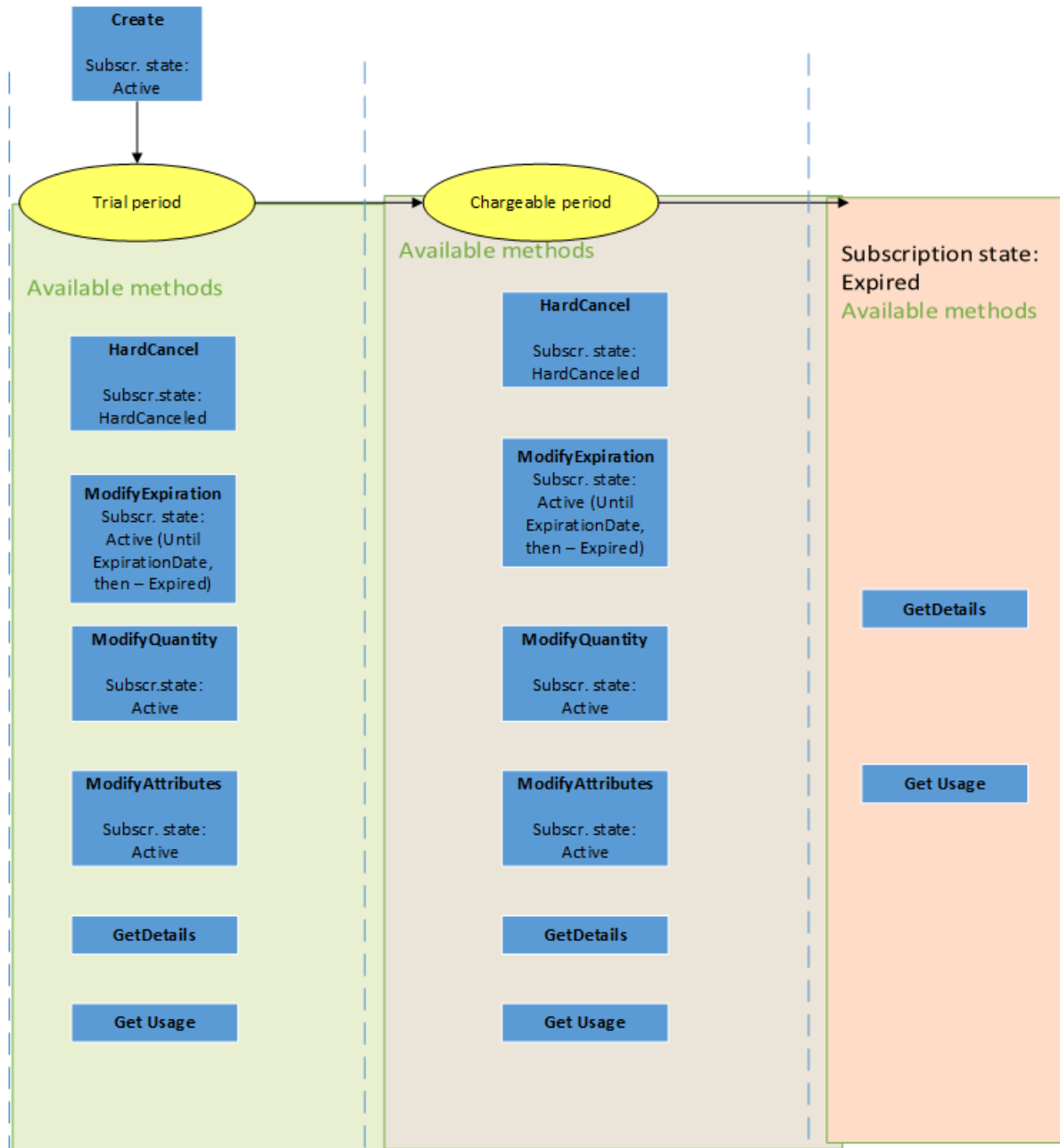


Figure 5. Subscription lifecycle

5.2. The example of the GetUsage response

```
{
  "BillingPeriods": [
    {
      "Id": 0,
      "Start": "2019-10-24T13:34:08.203Z",
```

```

    "End": "2019-10-24T13:34:08.203Z",
    "Type": "Free",
    "UsagePeriods": [
      {
        "Start": "2019-10-24T13:34:08.203Z",
        "End": "2019-10-24T13:34:08.203Z",
        "Quantity": 0
      }
    ]
  }
}

```

5.3. Error codes

| Name | Description |
|--|---|
| ApprovalCodeIsNotUnique | Specified approval code '{specifiedValue}' have been already used. |
| ApprovalCodeMismatch | Specified approval code '{specifiedValue}' does not match approval code '{originalValue}' of subscription. |
| AuthenticationFailed | <ul style="list-style-type: none"> Account with member id '{id}' is not found. Please contact your Manager at KL to get access to the system. Account with login name '{login}' is not found. Please contact your Manager at KL to get access to the system. Account with login name '{login}' is blocked. Please contact Manager at KL to unblock it. Provider for member with login name '{login}' is not found. Please contact your Manager at KL to get access to the system. |
| BillingPlanNotFound | Billing plan '{name}' not found. |
| DistributorNotApplicable | Distributor must not be specified. |
| ExpirationDateShouldBeEndOfCurrentPeriod | Subscription expiration should be the end of the current period for Yearly subscription. |
| ExpirationNotApplicable | Expiration should not be set. |
| IncorrectSubscriptionState | Subscription must be in active state. |
| Internal | Operation completed with error {0}. Please retry and if the problem still occurs, contact your Manager at KL and provide him with this error code. |
| InvalidSkuTerm | Sku should have yearly term. |
| MemberIsNotAllowedToAccessSubscription | The access is allowed only to the creator. |

| Name | Description |
|------------------------------------|---|
| PriceOfferAttributesMismatch | Approved Special price offer is not found for parameters in subscription. Please check the parameters in Special price offer: Start Date, End Date, parameters SKU and Quantity Threshold - or contact your KL manager. |
| PriceOfferDoesNotExist | Selected Approval code does not exist. |
| PriceOfferPartnerMismatch | PartnerCode in subscription does not match corresponding parameter in Special price offer. |
| PriceOfferResellerMismatch | ResellerPIN in subscription does not match corresponding parameter in Special price offer. |
| PriceOfferSubscriptionTypeMismatch | BillingPlan in subscription does not match corresponding parameter in Special price offer. |
| PriceOfferTypeMismatch | Special price type is not Deal SP. Only Deal SP can be specified on subscription. MSP SP is applied automatically after the end of the billing period. |
| SkuNotFound | Sku '{sku}' not found. |
| SkuNotFoundForQuantity | Sku based on '{sku}' not found for quantity {quantity}. |
| SubscriptionIdsUnknown | The subscription id '{id}' does not match any subscription. |
| Validation | Message does not contain any fixed part. Read message text to identify a cause of the error. |