API Overview

Author Kodmyran AB

Introduction

The Kodmyran Commerce platform provides several very different APIs, firstly the old SOAP API is still available but deprecated; secondly a range of new JSON/REST based APIs enables access to all parts of the platform without restrictions; and thirdly the JSON headless API provides a complete headless implementation. All APIs except the SOAP API are described in this documentation.

You can find the *obsolete* documentation for the SOAP API here:

SOAP API Documentation

Terms

- A *consumer* refers to the party calling the service, e.g. a webshop, an ERP system or a reporting tool
- A *provider* refers to the Kodmyran Commerce API that provides that specific service

The purpose of each API

- The integration API is used by webshops and other similar parties. It provides a unique view for each consumer such that different parties can be in different positions in the synchronization stream at any given time. Kodmyran Commerce has a very flexible synchronization model and supports several different patterns. Please have a look at the use-cases for the integration API to understood fully how it can be used.
- The entity API exposes every nut and bolt of Kodmyran Commerce. It is a simple API to extract or update any entity (object) in the database. However this API can be quite daunting to use for a beginner as deep understanding of fields and their relationships are required. This API is mainly intended for integration with upper-level systems (ERP) where the

built-in synchronization modules in Kodmyran Commerce do not support that ERP system directly. It can also be used for special purposes.

- The service API exposes a great deal of service calls into Kodmyran Commerce. E.g. calls to fully deliver an order. Where the entity API describes objects, but no actions on the objects other than changing fields, the service API provides the actions. Each HTTP call into the service is fully transactional and either completes in its entirety or is aborted with no changes to the backend.
- The report API is used to run various statistical reports and present the data in a form that can be displayed in BI tools, dashboards and similar.
- The media API is used to upload, query and remove files from the upload area of Kodmyran Commerce
- The headless API differs from the other APIs greatly, it is used when you want to use your own frontend (CMS) and simply query Kodmyran Commerce for e.g. the products to display in a category. The consumer requests a certain number of operations in an incoming command list and gets a number of responses wrapped in return. Do not use this API to integrate with a system that provides its own cart handling (e.g. complete e-commerce solutions like Magento, WooCommerce or Prestashop); use the integration API for those cases.

Please note that all APIs except the headless one execute in the superuser mode of Kodmyran Commerce (within the context of the adminuser the API key belongs to), whereas the headless API execute in user mode within the context of a particular end-user.

All API endpoints with the exception of the headless API are presented as OpenAPI/Swagger compatible services and are fully described by the swagger.json file.

Tools such as Postman, SwaggerUI, SoapUI, ReadyAPI etc. can easily be used for simulation and testing.

All strings are to be sent as UTF-8 per JSON standards.

You can access the swagger.json file for integration with various tools at:

`https://testaccount.shop4sale.se/admin/api/swagger.json`

Replace *testaccount* with your own domain provided when your account was setup.

Transactional rules

A single call to the JSON API is fully transactional, a failure during processing will result in a database rollback of all data changed during that call. The API

may in certain cases store individual updates and re-apply these after a rollback. This is intended to be used for call tracking/tracing only and not for normal operation and is invisible to API users.

Overload protection

All the end-points are regulated by an API call limit/minute, except the headless API. This limit is global for the entire account and is applied to all calls that can be authenticated. Calls that terminate prior to authentication are not included in the limit (e.g. fetching swagger.json or issuing some OPTION commands over HTTP).

The limit has a default value of 20 calls/minute but can be adjusted if required, contact Kodmyran for more information if this limit is too low.

Security

Kodmyran Commerce takes security very seriously and requires authentication of all calls, the use of HTTPS and provides several consistency checks to prevent misuse.

Non-headless APIs

All APIs except the headless requires all requests to be authenticated using one of the below four methods:

- An active session with a generated CSRF token, this method is used only by the GUI tools and should not be used by other parties.
- An API key present in the HTTP header, X-API-Key. This is the recommended method.
- HTTP Basic Authentication with the API key present in the user field and the magic character 'X' in the password field. This method is provided for backwards compatibility with certain dashboards (Geckoboard) that cannot pass authentication through a HTTP header. This method is not recommended and considered obsolete.
- A two-way CHAP handshake with multiple shared secrets, no password or API key is ever sent over the wire. This method is used only by Kodmyran for provisioning purposes and is not further described in this document.

Passing initial authentication is not sufficient by itself, you also need to communicate using HTTPS, and must be sending your queries to the account domain, not

the user domain. Hence you cannot call either https://www.myshop.com/admin/api or http://www.myshop.com/admin/api. All requests must be directed to https://<account>.shop4sale.se/admin/api - where account is replaced by your eight-character account name.

Once the key has been validated, and the domain name checked, the user associated with the API key is checked for proper permissions. Initially the user must possess the "Remote call: Read" and/or the "Remote call: Write" permissions (depending upon the HTTP request verb).

Once the user passes this check the role that they possess must also contain permissions to access the requested entity type. The permissions granted to that role for that object type dictates the users access. The available permissions are:

- bSelect, the ability to query data from this type of object. Without this permission, all read/GET requests are denied.
- bInsert, the ability to create new entries. Without this permission, all HTTP POST requests are denied.
- bUpdate, the ability to update an existing object. Without this permission, all HTTP PUT requests are denied.
- bDelete, the ability to delete an object. Without this permission, all HTTP DELETE requests are denied.

Headless API

The headless API requires the use of an API key to permit requests. This API key is unrelated to the API key used for the other APIs, it is an application unique string that must be provided in each call in the X-API-Key HTTP header.

To create the API key to use for the headless API you need to use the SOAP API first, and use the registerApplication call which will return an application ID/key in return. The headless API can only be used server-to-server and will not allow direct access using Javascript from a clients browser.

Synchronization patterns

Kodmyran Commerce has support for a wide variety of synchronization patterns, here are some common ways of synchronizing. The first four items are used with the entity API, the fifth option is strictly for the integration API.

1. Sync1/Sync2 based; some entities in Kodmyran Commerce contain two additional fields. There are present for the most commonly synchronized objects such as users, products and orders. The sync1 field is uniquely indexed (does not tolerate duplicates) and commonly contains the identity

of the remote system entity ID (e.g. the order ID in your system). Sync2 is similar but is not required to be unique. A common approach is to look for new objects by searching for an empty sync1 field in combination with looking at the changed field. It is also important to notice that sync1/sync2 exists as one set only. Hence it can only be used to synchronize with one other system, not multiple. Because of this limitation these fields are almost exclusively used to synchronize between Kodmyran Commerce and a higher-level ERP system. It should not be used for synchronizing e.g. webshops. By convention the sync1 field is prefixed with the remote system name (shortened) followed by a comma and then the remote ID.

2. Dirty flag; many entities in Kodmyran Commerce contain a dirty flag that is set whenever the object is written to (or initially created). You can search for all objects with the dirty flag set to find all new and changed objects. For each object, you update you clear the dirty flag by a special meta header. Like the sync1/sync2 fields this field is commonly used to synchronize with upper level ERP systems.
3. Generation; all entities in Kodmyran Commerce have a generation counter, it initially starts at generation 1 and is incremented for each change. In requests, you can set the generation you started from, if the generation number does not match the database the update is rejected with a generation mismatch error. This prevents two users overwriting the same object in a short amount of time. The use of this functionality is optional.
4. Changed/Created; all entities inside Kodmyran Commerce have two timestamps, one when the object was initially created and one for when it was last modified. The precision of the timestamp is in seconds.
5. Synchronization Views; this is a special type of view used by Kodmyran Commerce and the Integration API specifically to know which objects and changes that belong to which integration. If an entry exists in the view matching the newly updated object information is added to the synchronization log. When requested the synchronization log is sent in batches to the consumer which applies them in order.

HTTP Request Headers

All requests hitting Kodmyran Commerce must use the proper content-type. All current calls expect the content-type to be set to *application/json*