



Matrix42 Marketplace API Documentation

Version 01

Copyright Notice

All rights reserved, copyrights © Matrix42 AG 2003-2020.

This documentation is protected by copyright. All rights reserved by Matrix42 AG. Any other usage, in particular dissemination to third parties, storage within a data system, distribution, editing, speech, presentation, performance, and storage, are prohibited. This applies for the document in parts and as a whole. This document is subject to changes. The software described in this documentation is continuously developed, which may result in differences between the documentation and the actual software. All product names and trademarks used are the property of their respective owners

MATRIX42

Matrix42 AG
Elbinger Straße 7
D-60487 Frankfurt am Main
Telefon: +49-69-6677-3838-0
Telefax: +49-69-6677-8865-7
E-Mail: info@matrix42.com
Website: <http://www.matrix42.de>
Marketplace: <https://marketplace.matrix42.com>

Table of Contents

Matrix42 Marketplace Partner Center	4
Preparation	4
API Usage	6
Authentication and Headers	7
Result structure	8
Particular method routes	10
Check	10
MplProduct	10
MplOrder (with ID)	11
MplOrder	12
MplSubscription	13
MplPartnerCustomer	14
Data Contracts	15
Response	15
Generic Response	15
Enum-Value	15
MplSubscriptionInterval Enum-Value	16
MplCurrencyCode Enum-Value	16
MplProductStatus Enum-Value	16
MplContractKind Enum-Value	16
MplMetric Enum-Value	16
MplProduct	17
MplOrderRequestAddress	17
MplOrderRequestItemSku	18
MplOrderRequestItem	18
MplOrderRequest	18
MplOrderRequestResponseSubscription	19
MplOrderRequestResponse	19
Vendor	20
MplOrderCancellationSubscription	20
MplOrderCancellationResult	20
MplPartnerCustomer	21

Matrix42 Marketplace Partner Center

Marketplace is a central platform for obtaining product extensions, applications and trainings. It offers our customers and prospects a better overview of the extensive range of innovative Workspace Management Apps and Hardware products.

The Partner Center is a specific part of the Matrix42 Marketplace, that allows you to manage your own customers, as well as their subscriptions and orders. In the following pages, we will guide you through each area with further details on each page to make the most out of the API for the Matrix42 Marketplace Partner Center.

For further details please refer to the website <https://marketplace.matrix42.com> or contact the Marketplace Team marketplace@matrix42.com

The objective for the **Matrix42 Marketplace Partner-Center API** is to provide participants of the Marketplace Partner Program to access functionality surrounding ordering processes via a reliable AI.

Preparation

To participate in the Partner Center and to use the Marketplace API, the Partner needs to sign up for a Matrix42 Company Account. The registration can be conducted at <https://accounts.matrix42.com/my/register> .

Please note that if a company is already registered as a Partner, it is unnecessary to check the box at *Create Company Account*. However, if a company is not yet registered at the time of sign-up, the box needs to be checked and additional fields filled in.



Welcome to Matrix42.

Sign up and register in order to gain access to your Matrix42 Services

Mr

First Name

Last Name

E-Mail Address

Password

Password confirmation

English

Aiugfg

- Create company account
- I accept the privacy policy

Register

Once done, the Matrix42 Marketplace Team will validate the provided registration information to verify the account and authorize for the usage of the Marketplace API. Through the registration a unique Customer Number is assigned to the Partner.

After the registration, a Partner can create personal access tokens (PAT) at <https://accounts.matrix42.com/my/account> . These are used in the authentication mechanism for the API (see below).

Personal access tokens

You can use personal access tokens instead of a password to allow applications outside of the browser to access resources in your account.

Access token	Expires on	
MPLAPIToken	Nov 21, 2020	Delete token
Browser Extension_08c1	Apr 17, 2020	Delete token
Browser Extension_8389	Jun 16, 2020	Delete token
Browser Extension_39at	Sep 10, 2020	Delete token
Browser Extension_3fe2l	Apr 18, 2020	Delete token
iPhone 14DF0181-048A-4	Jul 20, 2017	Delete token
Browser Extension_694fbf	Apr 24, 2019	Delete token

Name:

Expires in:

Create token

Export Data

You can export all accounts/companies, that were created after a specified date in CSV format.

Created After:

Export Users Export Companies

For the technical setup for the API functionalities a Partner needs to provide a set of master data to Matrix42 that contains a full billing address and a valid billing email-address. The data is necessary to create valid Orders in the system.

API Usage

For the implementation of any client software to use the API it is necessary to understand the context in which this API exists as well as the processes existing within the Matrix42 Marketplace that apply to the retrieval and placement of Order-related data, including Product, Shipping, Subscription, and Order information.

The host for the API is a subdomain for the Matrix42 Website.

<https://mpl-api.matrix42.com/>

The particular routes included in the API are the following:

- ▶ **/mpl-api/Check** GET
Request for testing connectivity

- ▶ **/mpl-api/MplProduct** GET
DictionaryRequest for the list of current Marketplace-Products

- ▶ **/mpl-api/MplOrder** POST
Create a new Order

- ▶ **/mpl-api/MplOrder/{id}** GET
Load client-relevant data of an existing Order

- ▶ **/mpl-api/MplSubscription/{id}** DELETE
Cancel an existing Subscription

- ▶ **/mpl-api/MplPartnerCustomer/{id}** GET
Load Customers for Partner

Authentication and Headers

The authentication of all requests is based on a bearer token submitted in the header of the respective request.

The mechanism involves two steps. One is to obtain a personal access token (PAT) from <https://accounts.matrix42.com/my/account> (see above), the other is to obtain an access token via code from the exchange URL and use it in the authorization header of the request that is sent to the Marketplace API.

C# sample Code using PAT token to obtain token:

```
var httpWebRequest =  
    (HttpWebRequest)WebRequest  
        .Create("https://accounts.matrix42.com/api/session/tokens/exchange");  
httpWebRequest.Method = "GET";  
httpWebRequest.PreAuthenticate = true;  
httpWebRequest.Accept = "application/json";  
httpWebRequest.Headers.Add("Authorization", "Bearer " + pat_token);
```

The request has no payload body.

The result is a JSON with a single field containing the final token.

```
{  
  "RawToken": "eyJ0eXAiO..."  
}
```

The resulting Token can be used in all further requests as a Bearer Token in the Authorization Header.

Scope of retrievable Data

The data that can be queried by a Partner is restricted to those products that have been explicitly marked for availability in the Partner Center by Matrix42.

The initial list of orderable products is limited to the EgoSecure products (see below). Over time, more and more products will be marked as available for the Partner Center, so the list will grow. The intention is to perform a gradual release of products.

Product	License	Vendor
Cloud hosting Upgrade to Australia -	per User	Matrix42 AG
Cloud hosting Upgrade to Germany	per User	Matrix42 AG
Cloud hosting Upgrade to Switzerland	per User	Matrix42 AG
Matrix42 Endpoint Security Analytics SaaS	per User	Matrix42 AG
Matrix42 Endpoint Security Encryption SaaS	per User	Matrix42 AG
Matrix42 Endpoint Security GDPR SaaS	per User	Matrix42 AG
Matrix42 Endpoint Security Prevention SaaS	per User	Matrix42 AG
Matrix42 Endpoint Security Protection SaaS	per User	Matrix42 AG

Result structure

All routes return a **Result** object that contains overall information about the success of a request, as well as the requested payload data.

Field	Data-type	JSON-Example
Status-Value	Integer <i>Code for the condition of the response</i>	<pre>{ "StatusValue": 200, "Message": "Everything alright.", "ExceptionMessage": null, "ExtraInfo": null }</pre>
Message	String <i>Short informative notice</i>	
Exception-Message	String, optional <i>Elaborate, technical informative notice</i>	
Extra-Info	Dictionary <i>List of Key-Value pairs</i>	

The Status-Value can contain any of the following code values as integer:

200	OK	Request succeeded
403	Not allowed	Request failed due to an authorization restriction
404	Not found	Request failed due to unavailable result payload data

If the request resulted in a response carrying payload data, then the **Result** structure contains additional informational fields, as well as a **Data** field which is an array of the requested payload (it is an array even if only a single instance is returned).

Field	Data-type	JSON-Example
Status-Value	Integer	<pre> { "StatusValue": 200, "Message": "Hello world.", "ExceptionMessage": null, "TotalCount": 1, "Offset": -1, "Limit": -1, "Data": [{ "Id": 45614, "Name": "Acme Inc", "CustomerNumber": "2345678", "BillingName": "Acme Inc.", "Street1": "High Road 15", "Zip": "12345", "City": "Neverland", "Country": "DE", "Mail": "boss@acme.net" }] } </pre>
Message	String	
Exception-Message	String, optional	
Total Count	Integer	
<i>Overall number of available results</i>		
Offset	Integer	
<i>Start-index in paged data</i>		
Limit	Integer	
<i>Number of results in paged data</i>		
Extra Info	Dictionary	
<i>Additional key-value-paired information</i>		
Data	Array	
<i>Response payload</i>		

The functionality of the API and requesting data from it subject to the Matrix42 ACS authentication including the assignment of appropriate rights to individual client systems.

For some routes language-dependent results are returned. The selection of the language is made via the HTTP-Header

Particular method routes

The particular routes with their respective parameters are as follows:

Check

/mpl-api/Check **GET**

The purpose of the Check route is to simply allow for a client to test the connection, as to ensure the proper URL and authentication. The request has no URL arguments.

Response

```
{
  "StatusValue": 200,
  "Message": "Friday, 13 March 2020 14:08:42",
  "ExceptionMessage": null
}
```

MplProduct

/mpl-api/MplProduct **GET**

The purpose of the MplProduct route is to send a request for a list of current Marketplace-Products that are available for ordering within the Partner Center context. The request has no URL arguments.

The request results in MPL Products that contain all necessary information for display to an end-customer and for subsequently placing an order.

Response

```
{
  "StatusValue": 200,
  "Message": null,
  "ExceptionMessage": null,
  "TotalCount": 8,
  "Offset": -1,
  "Limit": -1,
  "ExtraInfo": {
    "Request-Language": "English [ LCID = 9 ]"
  },
  "Data": [ (Array of MPL Product) ]
}
```

The full specification of the result structure is contained in the Data Contracts section of this document. Due to its complexity it is not rendered here.

MplOrder (with ID)

/mpl-api/MplOrder/{id} **GET**

The purpose of the MplOrder route is to load client-relevant data of an existing Order. The URL argument is the Order-ID of an existing Order.

Request body

```
{
  "CustomerId": 65996,
  "CustomerName": "Matrix42 AG",
  "ShippingInfo": {
    "FirstName": "Stefan",
    "LastName": "Gies",
    "Company": "Matrix42 AG",
    "Address1": "Elbinger Strasse 7",
    "City": "Frankfurt am Main",
    "Postcode": "60487",
    "Country": "DE",
    "Email": "stefan.gies@matrix42.com"
  },
  "OrderItems": [
    {
      "ProductId": 162269,
      "ProductName": "Cloud hosting Upgrade to Germany",
      "Quantity": 1
    },
    ...More entries...
  ]
}
```

The request results in a single record containing the information of one Order, including the Subscription information derived from the Order.

Response

```
{
  "StatusValue": 200,
  "Message": null,
  "ExceptionMessage": null,
  "TotalCount": 8,
  "Offset": -1,
  "Limit": -1,
  "ExtraInfo": {
    "Request-Language": "English [ LCID = 9 ]"
  },
  "Data": [ (Array of MPL Order with one entry) ]
}
```

The full specification of the request and result structures is contained in the Data Contracts section of this document. Due to its complexity it is not rendered here.

MplOrder

/mpl-api/MplOrder **POST**

The purpose of the MplOrder route is to place a new Order within the Matrix42 Marketplace. The request has no URL arguments, and it conveys the Order information from the request body. The required Order information is based on the data obtained via the MplProduct route.

The request results in a single record containing the information of one Order, including the Subscription information derived from the Order. The result is identical in structure to the one retrieved via the previous MplOrder GET route with the Order-ID as argument.

Response

```
{
  "StatusValue": 200,
  "Message": null,
  "ExceptionMessage": null,
  "TotalCount": 8,
  "Offset": -1,
  "Limit": -1,
  "ExtraInfo": {
    "Request-Language": "English [ LCID = 9 ]"
  },
  "Data": [ (Array of MPL Order with one entry) ]
}
```

The full specification of the result structure is contained in the Data Contracts section of this document. Due to its complexity it is not rendered here.

MplSubscription

/mpl-api/MplSubscription/{id} **DELETE**

The purpose of the MplSubscription route called with the DELETE verb is to cancel existing Subscriptions. The URL argument is the comma-separated list of IDs of the Subscription, as obtained via either MplOrder call (POST for Order creation or GET by ID for Order retrieval).

The request results in a list containing information of the canceled Subscriptions grouped by originating Order, such as the success of the cancellation as well as the remaining run-time.

```
{
  "StatusValue": 200,
  "Message": null,
  "ExceptionMessage": null,
  "TotalCount": 4,
  "Offset": 0,
  "Limit": 0,
  "Data": [
    {
      "Id": 123456, /* Order ID */
      "Success": true,
      "Message": null,
      "Subscriptions": [
        {
          "Id": 45678, /* Subscription ID*/
          "Success": true,
          "Message": null,
          "StartDate": "2017-06-01T00:00:00",
          "NextPaymentDate": null,
          "EndDate": "2020-06-01T00:00:00"
        },
        ... more entries ...
      ]
    }
  ]
}
```

The full specification of the result structure is contained in the Data Contracts section of this document. Due to its complexity it is not rendered here.

MplPartnerCustomer

/mpl-api/MplPartnerCustomer **GET**

The purpose of the MplPartnerCustomer route is to return a list of customers with which a Matrix42 Partner enterprise has current interactions. The information conveyed in the result

The request results in a list of basic Customer information that suffices to create the Shipping information of an Order for an already existing Customer.

```
{
  "StatusValue": 200,
  "Message": null,
  "ExceptionMessage": null,
  "TotalCount": 216,
  "Offset": -1,
  "Limit": -1,
  "Data": [
    {
      "Id": 66666,
      "Name": "Matrizius",
      "CustomerNumber": "6789012",
      "BillingName": "Matrizius",
      "Street1": "Der Beste Weg 3",
      "Street2": "Matrizius GmbH & Co. KG",
      "Zip": "76137",
      "City": "Karlsruhe",
      "Country": "DE",
      "Mail": ""
    },
    ... more entries ...
  ]
}
```

The full specification of the result structure is contained in the Data Contracts section of this document. Due to its complexity it is not rendered here.

Data Contracts

Response

This is the basic contract for all responses returned from the API.

```
{
  StatusValue (integer, optional),
  Message (string, optional),
  ExceptionMessage (string, optional),
  DebugInfo (string, optional),
  ExtraInfo (Key-Value-pairs list, optional)
}
```

Generic Response

This is the basic contract for all responses returned from the API that carry payload data.

```
{
  StatusValue (integer, optional),
  Message (string, optional),
  ExceptionMessage (string, optional),
  DebugInfo (string, optional),
  TotalCount (integer, optional),
  Offset (integer, optional),
  Limit (integer, optional),
  ExtraInfo (inline_model_2, optional),
  Data (Array[Specific Response data], optional)
}
```

Enum-Value

All Enum-Value contracts share the same structure. The only difference is the concrete values the Value and Raw fields can take. This construct serves to transport language-dependent enumeration values. The Raw field contains the invariant human-readable name, while the Caption field contains the human-readable name in the requested language.

```
{
  Value (integer, optional),
  Raw (string, optional),
  Caption (string, optional)
}
```

MplSubscriptionInterval Enum-Value

0	None
1	Month
2	Year

MplCurrencyCode Enum-Value

0	None
1	EUR

MplProductStatus Enum-Value

0	None
1	Published

MplContractKind Enum-Value

0	Unspecified
1	Subscription
2	Hardware
3	Service

MplMetric Enum-Value

0	None
1	Offsite Service
2	One Time
3	Onsite Service
4	Per Agent
5	Per Customer Entity
6	Per Device
7	Per Installation
8	Per User
9	Remote Service
10	Per Server
11	Per Business Application
12	None
13	Offsite Service
14	One Time
15	Onsite Service

MplProduct

This is the contract that defines the properties of a Product that can be ordered from the Marketplace.

```
{
  Id (integer, optional),
  Name (string, optional),
  Key (string, optional),
  ShortDescription (string, optional),
  Description (string, optional),
  SignupFee (number, optional),
  RegularPrice (number, optional),
  SubscriptionPrice (number, optional),
  SalesPrice (number, optional),
  SubscriptionPriceInterval (EnumValue[MplSubscriptionInterval], optional),
  CurrencyCode (EnumValue[MplCurrencyCode], optional),
  ProductLogoUrl (string, optional),
  Status (EnumValue[MplProductStatus], optional),
  Tags (Array[string], optional),
  Categories (Array[string], optional),
  ContractType (EnumValue[MplContractKind], optional),
  Metric (EnumValue[MplMetric], optional),
  ProductTypeName (string, optional),
  PublishingDate (string, optional),
  LastUpdate (string, optional),
  SkuSubscription (string, optional),
  SkuSignupFee (string, optional),
  SkuRegularPrice (string, optional),
  Vendor (Vendor, optional)
}
```

MplOrderRequestAddress

This is the contract to carry address information for a location included as the shipping or billing address of an Order.

```
{
  FirstName (string, optional),
  LastName (string, optional),
  Company (string, optional),
  Address1 (string, optional),
  Address2 (string, optional),
  City (string, optional),
  State (string, optional),
  Postcode (string, optional),
  Country (string, optional),
  Email (string, optional)
}
```

MplOrderRequestItemSku

This is the contract to carry the SKU Identifiers for the three kind of prices referenced in an Order-Item.

```
{
  SubscriptionPrice (string, optional),
  SignUpFee (string, optional),
  RegularPrice (string, optional)
}
```

MplOrderRequestItem

This contract carries the information about the item of an Order including the identification of the Product that is being ordered.

```
{
  ProductId (integer, optional),
  ProductName (string, optional),
  Quantity (integer, optional),
  VariationId (integer, optional),
  SKU (MplOrderRequestItemSku, optional)
}
```

MplOrderRequest

This contract is the container structure to carry the information about an Order that is placed in the Marketplace via the API.

```
{
  CustomerId (integer, optional),
  CustomerName (string, optional),
  Notes (string, optional),
  ShippingInfo (MplOrderRequestAddress, optional),
  OrderItems (Array[MplOrderRequestItem], optional),
  meta_data (Array[MplOrderMeta], optional)
}
```

MplOrderRequestResponseSubscription

This contract defines the structure of a Subscription that was derived and returned from an Order placed in the Marketplace.

```
{
  SubscriptionId (integer, optional),
  Period (string, optional),
  Interval (string, optional),
  Status (string, optional),
  StartDate (string, optional),
  NextPaymentDate (string, optional),
  EndDate (string, optional),
  LastOperationSuccess (boolean, optional),
  LastOperationMessage (string, optional),
  Items (Array[MplOrderRequestItem], optional)
}
```

MplOrderRequestResponse

This contract represents the data returned from the placement of an Order in the Marketplace at the moment of its creation as well as at later requests.

```
{
  Id (integer, optional),
  OrderKey (string, optional),
  CreationTimeUtc (string, optional),
  CustomerName (string, optional),
  Status (string, optional),
  TotalTax (number, optional),
  Total (number, optional),
  ShippingTax (number, optional),
  ShippingTotal (number, optional),
  DiscountTax (number, optional),
  DiscountTotal (number, optional),
  ItemsCount (integer, optional),
  Subscriptions (Array[MplOrderRequestResponseSubscription], optional)
}
```

Vendor

This small contract contains the basic information about a Vendor of a particular Product that can be ordered from the Marketplace.

```
{
  Id (integer, optional),
  Name (string, optional),
  Key (string, optional)
}
```

MplOrderCancellationSubscription

The result of a Subscription cancellation is carried with this contract. It contains information about the success and remaining run-time of a Subscription.

```
{
  Id (integer, optional),
  Success (boolean, optional),
  Message (string, optional),
  StartDate (string, optional),
  NextPaymentDate (string, optional),
  EndDate (string, optional)
}
```

MplOrderCancellationResult

This is the container contract that is returned upon the cancellation of Subscriptions, and which aggregates these Subscriptions according to the Order from which they derived originally.

```
{
  Id (integer, optional),
  Success (boolean, optional),
  Message (string, optional),
  Subscriptions (Array[MplOrderCancellationSubscription], optional)
}
```

MplPartnerCustomer

This is the contract that holds information about a Customer that is serviced by the requesting partner. It can be used to simplify the creation of data for the placement of an Order by providing the individual Customer-Number and the data to be filled into the Shipping-address.

```
{  
  Id (integer, optional),  
  Name (string, optional),  
  CustomerNumber (string, optional),  
  BillingName (string, optional),  
  Street1 (string, optional),  
  Street2 (string, optional),  
  Zip (string, optional),  
  City (string, optional),  
  Country (string, optional),  
  Mail (string, optional)  
}
```