

Standard Scheme

Description

This trading scheme can be used to place orders on multiple products at once. When placing an order, the partner must pass the unique order number and a list of order lines as input parameters. Each line needs to contain a part number of the ordered item, desired quantity, and order line number (which is also unique compared to all user's order lines in the MONT system). We recommend the following approach to generating order numbers:

Assume an order has been placed in the partner's system for 2 products: abc-11111 and 6qr-00006. The system assigned a unique number to the order, for example KEYS-00004280. In this case, when the order is placed in the MONT system, the partner's number is used as it exists in the system (KEYS-00004280), and KEYS-00004280-N number is assigned to each line where N is order line sequential number.

ATTENTION!

When designing their proprietary system, the partner needs to remember that keys/links/files ordered via ESD cannot be returned and an order cannot be canceled after placement. Therefore, we recommend that you place an order in the MONT system only after the end users effects a payment. The need to follow this recommendation depends on the partner's business processes.

System operation procedure

1. [GetPriceChanges](#). The partner receives current prices. [Learn more...](#)
2. [GetMenu](#). The partner receives a list of product categories for their website data mart. [Learn more...](#)
3. [GetProduct](#). The partner downloads a description for each product.
4. The buyer selects a product to be ordered on the partner's website.
5. [CheckLimits](#). The partner checks their available funds limit. [Learn more...](#)
6. The partner checks product properties. If the product to be ordered does not require a license form to be filled out, proceed to **step 10**.
7. [GetUserInfoDescriptionFor](#). The partner receives license form fill out guidelines for the selected product.
8. The buyer fills out the license form following the guidelines describing fields. [Learn more...](#)
9. [CheckEndUserInfo](#). The partner checks whether the license form is filled out correctly.
10. [AddOrder](#). The partner adds the order to the MONT system.
11. [GetOrder](#). The partner regularly requests the placed order until its status changes to Processed Successfully or Error



Some partners who use cash registers ask to send them both the key and the product installation/activation guidelines to be printed on the receipt. [Learn more...](#)

Partner requirements

- The partner shall be registered with the MONT system with a valid user name and password.
- The partner shall have a list of product part numbers in the MONT system they intend to sell. At least one product shall be unlocked for sale.
- The partner shall have all required contracts. Contract status is displayed by calling the [CheckLimits](#) method.
- The partner shall have a sufficient funds limit to buy products.
- If the partner uses receipts, the MONT system shall be configured to Prioritized Key Delivery when keys are immediately delivered in response to the request.

To specify signup details, contact your personal MONT manager or write to ESDTeam@mont.com

Integration procedure

Partner integration with MONT services comprises two steps:

1. Development, debugging and testing in the test system
2. Launch in the production system

Obtaining test access

All MONT services are available in both production and test (sandbox) environments. To ensure system integrity, partners can work in the sandbox only during the development stage. To obtain access (user name and password), send a message to webstore_support@mont.com with the

subject 'Test access for auto-placement of orders'. The reply will contain test user name and password.

User name/password for auto-placement of orders is in no way connected with user name/password for placement of orders via web-interface (portal.mont.ru, eShop). A different user name is ALWAYS used for auto-connection.

Transfer to production platform

When the partner believes that integration is completed on their side and they are ready to switch to production mode, they need to test integration. Integration testing procedure is described for each module (and included in service description) and can sometimes be modified (for example, if the partner integrates with eKey, but will only buy keys for Microsoft Office). After testing is successfully completed, the partner is provided with a user name in the production system. To receive a user name, send a message to webstore_support@mont.com.

If you have questions or something is out of order during or after integration, you can send a message to webstore_support@mont.com.

Retrieve Information about Limits and Reseller Account

Before a partner engages in selling, they need to retrieve information about their available limit (for each channel) and additional information about their account.

Common scenario of retrieving information about available limit and account

1. Partner calls the [GetMyDetails](#) method. The reply has a set of fields with information about the partner's account, such as country code, company name, main current code, etc.
2. Partner calls the [CheckLimits](#) method. It retrieves the [ResellerLimitContract](#) array. Each element in the array has the LimitType field. Its value is ESD for a ESD channel or CSP for a CSP channel.
3. We recommend our partners to check the limit when placing new orders.

End User Information

When placing an order for certain products, the vendor requires that particular end user information should be specified (hereinafter — license form). Different products may also require different information.

When placing an order for a product, you need to pass a license form if the LicenseFormRequired flag is selected for the product (in the [GetPriceChanges](#) method results).

Before placing an order, call the [GetUserInfoDescriptionFor](#) method by passing a description language and list of products to be ordered. Results will include a list of fields to be filled out by the end user. Each field has a text description and a set of fill out requirements (regular expression, length, etc.).

Please, note that certain fields must be filled out by the partner rather than the end user. Field type is defined in the description of the UserType field. Values in such fields must be auto-populated by the partner's system and these fields must not be displayed to the end user. Currently, we only have one field of this type: [Kaspersky PIN](#).

Unfortunately, some form fill out rules cannot be described using regular expressions (for example, a rule like 'Company field is required only if the user is a legal entity'). These rules are included in the description to help users filling out the form. To check whether the form is filled out properly, use the [CheckEndUserInfo](#) method. As each product has its own form, the check must run as follows:

1. The form is collected for each product in the order. To do this, all filled-out fields are filtered to select those that have a product's part number from the RelatedPartNums array in the description.
2. The form collected for each product is then checked using the [CheckEndUserInfo](#) method. If some fields contain errors, they will be marked with the HasError flag in the method call results and the error text will be displayed in the ErrorText field.
3. If the filled-out form contains errors, these are displayed to the user to be corrected, and then steps 1-2 are repeated.

When a license form is added to an order line, fields must be filtered by product, just as in the case of form check.

Download and Update a Price List

Before a partner can sell products (available to the partner depending on the addendum), they need to obtain a list of these products with corresponding prices and system features.

1. To download the price list, the partner needs to call the [GetPriceChanges](#) method (eKey service) with the parameter version = 0.
2. In response, the method will return the list of all products available for sale.
3. Always use the [GetPriceChanges](#) method (eKey service) with the parameter version = 0
4. In addition to price and part number, each [PriceContract Type](#) element in the result includes various additional information that can impact the order. It must also be stored and used on the partner's side.
5. We recommend that you update prices at least once per day and immediately **before placing an order**.

Print Guidelines on Receipts

Some partners who use cash registers ask to send them both the key and the installation guidelines. To do this, the MONT system can store product installation/activation guidelines. Guidelines to be printed on receipts have parameters set up for each partner individually.

Parameters:

- Height — maximum number of lines per receipt, including the title, to be returned to the user.
- Length — length of the receipt line. Title and body of the guidelines will be automatically broken down into lines of specified length and returned in xml format with a number of lines set in the Height parameter.
- The body (guidelines) is included in the product on the MONT's side.

Set up guidelines in the MONT system

To set up guideline size, the partner needs to contact partner support at webstore_support@mont.com and provide the following:

- Login
- Test or production platform
- Message stating the need for setting up guidelines to be printed on receipts
- Length and height of receipt in characters (optimal line length for a receipt is 30–40 characters, receipt height is unlimited)

Attention! Partner shall amend their own system to print guidelines on receipts.

Receive guidelines with the order

[AddOrderWilInstruction](#) method can be used to place an order and immediately receive guidelines in the reply.

[GetOrderWilInstructions](#) method can be used to receive guidelines for a placed order.

Create a Product Catalogue

You can use the Content service to create a catalogue of products with corresponding descriptions. Using this service to sell keys is not mandatory for resellers.

You don't need to authorize to use most Content service methods.

Common scenario of content retrieval

1. The customer can use the [GetAvaliableCountries](#) method to retrieve a list of available countries.
In some cases, a single family/product may have different descriptions for different countries. As these products are rare, the most common country code for all translations is DF (Default). It means that a product with this country code will have identical translations in all languages regardless of the country, except individual descriptions with a country code different from DF.
2. The customer can use the [GetDefaultMenu](#) (Content) method to retrieve a category tree
3. The customer can obtain a list of all available languages using [GetAvaliableLanguages](#) (Content)
4. Each tree item in the [MenuElementContract](#) menu has the FamilyGroupIds property with a list of ID's assigned to this category (menu item). The customer can use the price list (where each item has a FamilyID) to calculate which menu items are available for sale (have at least 1 product). [Potential algorithm](#)
5. Then, the customer downloads a description for each available product or family in a desired language using the [GetFamily](#) and [GetProduct](#) (Content) methods

Additional scenario of content retrieval

1. To avoid requesting descriptions of all available products every time, the customer may only retrieve a list of families/products that have lately been modified in any way. To do this, use the [GetChanges](#) (Content) method. This method uses the last requested content version as input.

GetUserInfoDescriptionFor

Description

To buy certain products, end user information needs to be passed to the manufacturer. This method can be used to get description of the form to be filled out by user when buying products indicated in input parameters.

Parameters

Name	Type	Description
langCode	string	Code of language to be used for description
partNums	array of string	List of part numbers requiring a license form

Return value

Type	Description
array of EndUserInfoDescriptionContract	List where each value describes a fill out rule for a single field of the license form

EndUserInfoDescriptionContract

It contains information about fillout rules for a particular field in the end user license form.

List of fields

Название	Тип	Описание
FieldName	string	Field name
FieldTitle	string	Text title (can be used to display title to the user on the data mart)
FieldDescription	string	Field description
RegExprs	string	Regular expression describing field fill out rules (with regard to the allowed value length). Obsolete, not recommended for use
SmallRegExprs	string	Regular expression describing field fill out rules (without regard to length)
FieldsValue	string	Field value (it's only used when an object is the result of the ActivateSubscriber method call)
AvailableValues	array of string	List of allowed values for this field (it's only used for One-to-Many fields)

Название	Тип	Описание
ErrorText	string	Error text (it's only used when an object is the result of the CheckEndUserInfo method call)
HasError	bool	Error flag (it's only used when an object is the result of the CheckEndUserInfo method call)
RelatedPartNums	array of string	List of part numbers requiring this field to be ordered
FieldType	int	Field type. 1 - text, 2 - Boolean, 3 - one-to-many
UserType	int	User this field is used for. 1 - specific user, 2 - reseller
Required	bool	Field required flag
MinLength	int	Minimum length of field value
MaxLength	int	Maximum length of field value

AddOrder

Description

This method is used to add an order to the Mont Webstore system. The input object contains order number and a list of order lines. Each line matches a single ordered item and has a unique number. It describes:

- part number of the ordered item;
- quantity;
- filled out license form with end user data;
- ID of subscriber that subscription is created for (filled out for subscriptions only).

Parameters

Name	Type	Description
orderRequest	SendOrderContract	Object containing order parameters

SendOrderContract type

Name	Type	Description
OrderNum	string	Order number in the MONT system. It is created on the partner's side and must be unique within the entire pool of partner orders.
SendOrderLines	array of SendOrderLineContract	Array containing all lines of this order

Return value

All replies from [B2BServiceV2Xm1.svc](#) and [B2BServiceV2.svc](#) web-services become a contract of the following format (the contract contains information about errors in web-service operation and a reply from the called method within the <Data> tag):

Type	Description
OrderContract	Object with the placed order containing current order status and numbers of created subscriptions

Potential errors of the **AddOrder** method.

Код	Текст ошибки
2	Product not found
4	Order with that number (order line) is already exists in the MONT system
6	Credit limit exceeded
8	Order line number field is not specified
10	Product is blocked
12	The license form has not been specified

Код	Текст ошибки
16	Reseller is blocked
17	Quantity is incorrect
25	Invalid format for the order line number. For Microsoft products, the line number must not exceed 32 characters.
61	Subscription not yet activated
71	Order number field is not set
72	There are no order lines in the order
75	Subscriber with the specified code was not found
76	Subscriber with the specified code was not activated
87	No subscriber specified for a subscription
101	Internal service error
200	Incorrect EndUserInfo Data
900	You do not need an activation form for that product
6000	Operation not allowed
6005	It is forbidden to order products with personal licenses

CheckEndUserInfo

Description

This method is used to check whether the license form for the product to be purchased is filled out correctly. The method returns a list of fields fill out rules. Rules containing errors will be marked with the HasError flag, and custom error text will be displayed in the ErrorText field. The IsError root flag of the reply contract is TRUE only if the web-service is out of operation. This flag does not indicate if there are errors in the checked license form.

Parameters

Name	Type	Description
endUserInfo	array of EndUserInfoContract	Filled out license form. You can obtain fillout rules using the following method: GetUserInfoDescriptionFor
langCode	string	Code of license form description language (for example, RU)
partNum	string	Part number of the product to be purchased with the license form

Return value

Type	Description
array of EndUserInfoDescriptionContract	License form fillout errors.

GetMenu

Description

This method returns a list of product family categories in the required language. ([Content](#) service)

Parameters

Type	Name	Description
int	id	Menu ID
string	langCode	Required menu language

Return value

Type	Description
Array of MenuContract	Menu containing an updated category version and tree

MenuElementContract

It contains information about menu item (category)

Список полей

Название	Тип	Описание
FamilyGroupIds	List<int>	List of family ID's
Id	int	ID
ImageUrl	bool	Availability of family image URL
LegacyId	int	
Level	int	Nesting level
OrderIndex	int	Sorting order
ParentId	int	Parent Id
Title	string	Name

GetOrder

Description

This method is used to obtain information about the order placed in the MONT system.

Parameters

Name	Type	Description
orderNumber	int	Order number

Return value

Name	Type	Description
OrderNum	string	Order number
OrderStatus	int	Current order status. Possible values are described in a dictionary OrderLineStatusEnum If all order lines are successfully completed, order status is Completed Successfully. Otherwise, order status is In Progress. (It means an order can never have the Error status.)
DateCreated	DateTime	Order creation date
OrderLines	array of OrderLineContract	List of order lines

It should be noted that the Microsoft vendor has products with FulfillmentType = 1. For these products, the vendor limited the link expiry period to 24 hours. To request an updated link, use the same GetOrder method. The reply will contain updated links that are valid at the time of method call.

Possible errors

Code	Error text
3	Order cannot be found

GetPriceChanges

Description

This method is used to obtain a list of changed products, starting with the indicated version. In the current implementation, versions are no longer supported and any passed parameter (version) value will return the current system version and a list of all products available to the user (just like when 0 parameter is passed). To ensure validity, we recommend that you pass 0 every time to obtain the list of products and update it in your system.

Parameters

Name	Type	Description
version	int64	Latest price change version in reseller's system. Always pass 0

Return value

Type	Description
PriceChangesContract	Object with changed prices where version is greater than the version value indicated in the parameter

PriceChangesContract

String value	Number value	Description
InProcess	1	the order takes time to complete, or not all the order lines are completed (In process)
Error	2	Completed with error (Error) - Possible errors
Succeeded	3	Completed successfully (Succeeded)

GetProduct

Description

This method returns description of a single product in the specified language for the specified country.

Parameters

Type	Name	Description
string	partnum	Product ID — part number
string	langCode	Code of language to be used for description
string	countryCode	Code of the country reseller belongs to

Return values

ProductContract type

Contains product information

List of fields

Name	Type	Description
PartNumber	string	Part number is a unique product ID in the MONT system
Title	string	Name
UrlImage	string	Image URL-address
FamilyGroupId	int?	Code of family (product group)
VendorId	int	Unique vendor ID in the Content service
VendorCode	string	Unique vendor code in all Mont services
OrderNum	int	Value of product's sorting index used to sort the list of products before displaying on screen. Sorting index has a value for all products within a product family
Preview	string	Preview
ShortDescription	string	Short description
FullDescription	string	Full description
ImportantInformation	string	Important information
SystemRequirements	string	System requirements
InstallInfo	string	Installation guidelines
EventText	string	Event information
Specs	List< SpecContract >	Array of specifications

Name	Type	Description
LangCode	string	Code of language description is requested for
CountryCode	string	Code of country description is returned for*
IsContentAvailable	bool	Indicates whether description exists for this product
Media	Object	Object containing previews for product images and videos

Media object

Name	Type	Description
Images	array of Object	Object containing a list of images for product preview
Videos	array of Object	Object containing a list of videos for product preview

Images object

Name	Type	Description
MediaImageContract	Object	Object containing image parameters

MediaImageContract object

Name	Type	Description
ImageUrl	String	Image URL

Videos object

Name	Type	Description
MediaVideoContract	Object	Object containing video parameters

MediaVideoContract object

Name	Type	Description
Name	String	Video name
URL	String	Video URL
VideoType	Int	Video type 1 - YouTube link, 2 - direct link

EndUserInfoDescriptionContract Type

It contains information about fillout rules for a particular field in the end user license form.

List of fields

Name	Type	Description
FieldName	string	Field name
FieldTitle	string	Text title (can be used to display title to the user on the data mart)
FieldDescription	string	Field description
RegExprs	string	Regular expression describing field fill out rules (with regard to the allowed value length). Obsolete, not recommended for use
SmallRegExprs	string	Regular expression describing field fill out rules (without regard to length)
FieldsValue	string	Field value (it's only used when an object is the result of the ActivateSubscriber method call)
AvailableValues	array of string	List of allowed values for this field (it's only used for One-to-Many fields)
ErrorText	string	Error text (it's only used when an object is the result of the CheckEndUserInfo method call)
HasError	bool	Error flag (it's only used when an object is the result of the CheckEndUserInfo method call)
RelatedPartNums	array of string	List of part numbers requiring this field to be ordered
FieldType	int	Field type. 1 - text, 2 - Boolean, 3 - one-to-many
UserType	int	User this field is used for. 1 - specific user, 2 - reseller
Required	bool	Field required flag
MinLength	int	Minimum length of field value
MaxLength	int	Maximum length of field value

MenuElementContract

It contains information about menu item (category)

List of fields

Name	Type	Description
FamilyGroupIds	List<int>	List of family ID's
Id	int	ID
Imageurl	bool	Availability of family image URL
LegacyId	int	
Level	int	Nesting level
OrderIndex	int	Sorting order
ParentId	int	Parent Id
Title	string	Name

OrderLineStatusEnum Dictionary

Describes possible order statuses

Possible enumeration values

String value	Number value	Description
InProcess	1	In process
Error	2	Completed with error
Succeeded	3	Completed successfully

ResellerLimitContract Type

Contains complete information about partner's limit.

List of fields

Name	Type	Description
Limit	double	Partner's funds limit for product purchases
LimitType	string	Type of limit. It can be ESD or CSP
Blocked	bool	Product purchases blocked for partner
BlockReason	string	Block reason (if known)

Sample Code (PHP)

- [Sample using HTTP methods](#)
- [Sample using SOAP methods](#)

Login into Webstore system (PHP)

```
function do_https($url, $postfields) {

    $login = "";

    $pass = "";

    $ch = curl_init($url);

    // Login

    curl_setopt($ch, CURLOPT_HTTPAUTH, CURLAUTH_BASIC);

    curl_setopt($ch, CURLOPT_USERPWD, $login . ':' . $pass);

    // during the test, disable certificate authenticity verification 'SSL
    certificate problem, verify that the CA cert is OK'

    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);

    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);

    // return result in variable

    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);

    if ($postfields)
    {

        curl_setopt($ch, CURLOPT_POST, 1);
```

```

    curl_setopt($ch, CURLOPT_HTTPHEADER, Array("Content-Type:application/xml;
charset=utf-8", "Content-Length:" . strlen($postfields)));

    curl_setopt($ch, CURLOPT_POSTFIELDS, $postfields);

}

$xmlString = curl_exec($ch);

$httpCode = curl_getinfo($ch, CURLINFO_HTTP_CODE);

$xml = simplexml_load_string($xmlString);

curl_close($ch);

return $xml;

}

```

Placement of simple order (PHP)

```

$orderXML = new SimpleXMLElement($orderXMLString);

$orderXML->OrderNum = 'MyUnicumOrderNumber-1234567890';

$orderXML->PartNum = 'T5D-00432';

$orderXML->quantity = 1;

$postString = $orderXML->asXML();

$returnOrder =
do_https("https://sandbox.webstore.mont.ru/B2BService.svc/web/AddOrder",
$postString);

```


Obtaining result of order fulfilment (PHP)

```
$orderXML = new SimpleXMLElement($orderXMLString);

$orderXML->OrderNum = 'MyUnicumOrderNumber-1234567890';

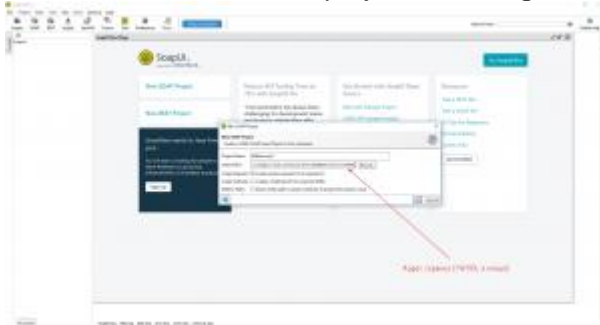
$postString = $orderXML->asXML();

$returnOrder =
do_https("https://sandbox.webstore.mont.ru/B2BService.svc/web/GetOrder",
$postString);
```

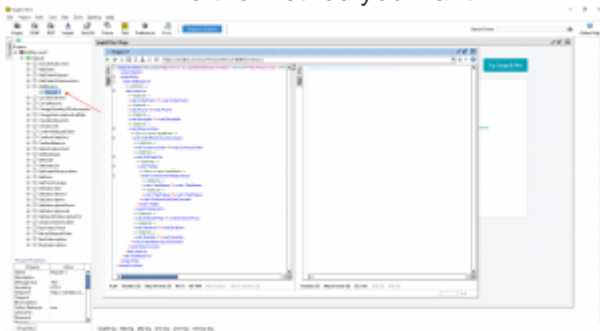
Example of running REST and SOAP requests to service using SoapUI SOAP

Method descriptions contain bodies of SOAP requests in the format used to send them to the server. You cannot re-run a request by simply copying and pasting its body from the description to SoapUI. Part of the body (Security) is formed dynamically. Below are the steps you need to take to configure SoapUI and form a valid request using the [AddReserve](#) method as an example.

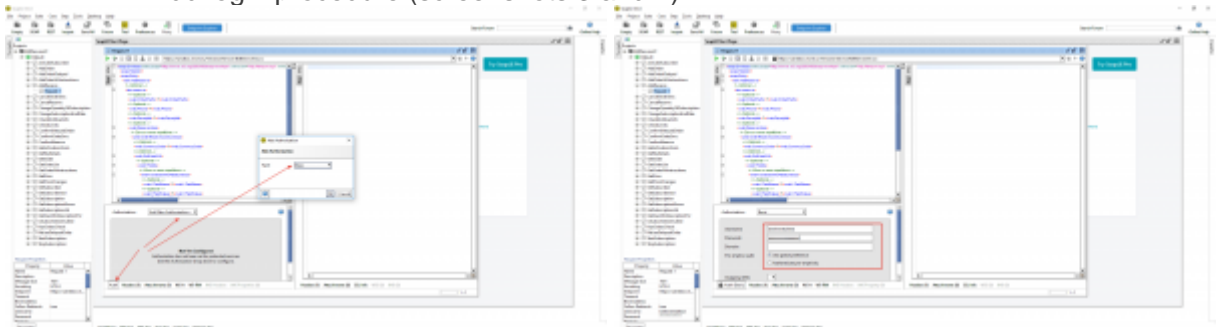
- Create a new project indicating service address



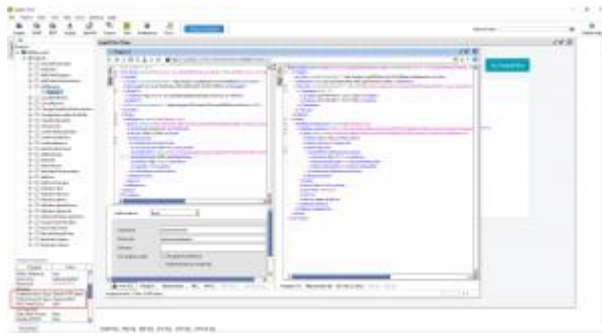
- Find the method you want



- Add login procedure (screenshots 3 and 4)



- Configure additional login parameters (configure steps are from the [SoapUI website](#))



- Paste the [SOAP example from documentation](#) into the request body box and remove the Security section.
- Send the request

REST

You can also configure login procedure using a similar process as described in the section above. In this case SoapUI encrypts logon credentials based on the BasicAuth specification.

You can add the Authorization title manually. In this case you must encrypt data and insert the result into the title.

The other mandatory condition is that you need to explicitly indicate Content-Type title - its value must be application/xml.

Below is a screenshot of a valid REST request.

