

Promotion module

- [Promotion entities import / export](#)
- [Examples of popular promotions](#)
- [FIDEN2](#)
- [Discount codes](#)
- [Promotion pages](#)
- [Actions](#)
- [Rules](#)
- [Conditions](#)
- [Promotion module introduction](#)
- [Opensets](#)

Promotion entities import / export

Screenshot_2020-04-29_at_19.1 Getting started

In this article, you can find information about the entities import / export, which allows admin panel users to export already prepared promotion conditions, actions, or rules and import them on different websites or environments. The export extracts the selected promotions "as is", which means they export them without the ability to modify them on the computer and the downloaded .zip file can only be imported. Whenever the user exports the promotion rule the entities functionality automatically exports all conditions and actions used in the selected rule.

info2.svg **Remember**

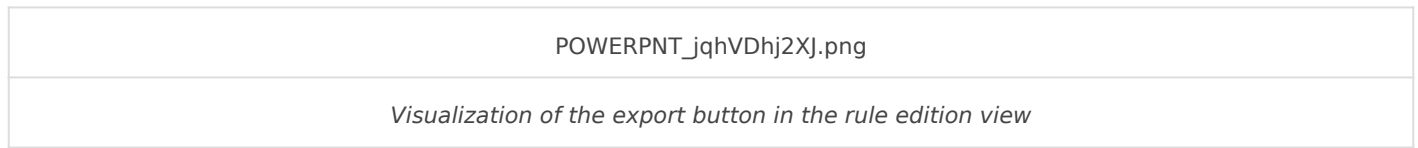
You can find more about the promotion module in the [Promotion module introduction](#) article. More detailed description of conditions, action and rules can be found in their respective articles: [Conditions](#), [Actions](#) and [Rules](#).

Table of contents:

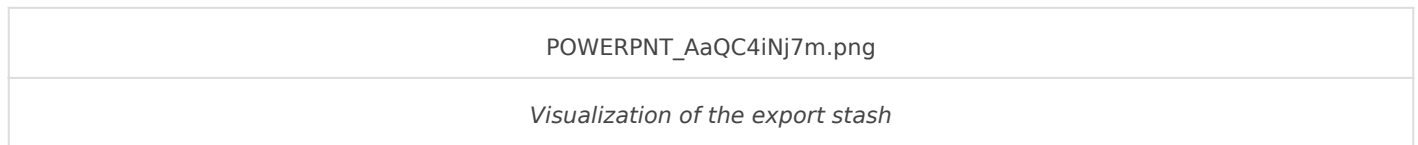
1. [Exporting entities](#)
2. [Importing entities](#)

Screenshot_2020-04-29_at_19.1 Exporting entities

Promotion conditions, actions, and rules can be exported in their respective tabs while editing them. It is possible to export multiple promotions at once, and by exporting the promotional rules all conditions and actions that are active in them are also automatically exported, so whenever a rule is exported there is no need to manually add the used in this rule actions and conditions.



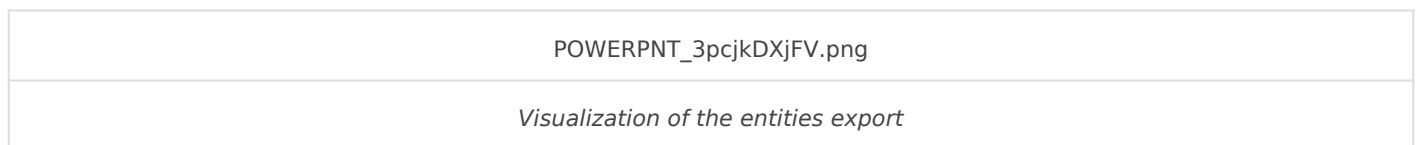
By clicking the button **Add to export (1)** you add this promotion part to the export stash **(2)**.



In the export stash, you can see all added entities with an option to clear the list or export them. By clicking the **Export** button the entities will be available to download.

info2.svg **Remember**

The export stash holds all the added entities even after exporting them. You can clear the stash by selecting the **clear** button, or it will get cleared itself whenever the browser is closed.



The exported .zip file shouldn't be unpacked or modified, as it contains all the necessary information for our system to reproduce these conditions, actions, and promotions whenever they are imported.

Screenshot_2020-04-29_at_19.1

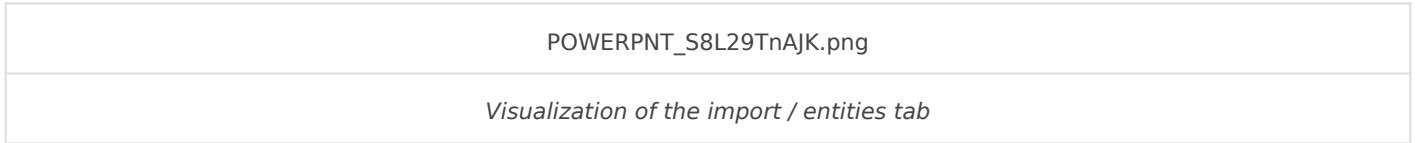
Importing entities

With a ready exported .zip file with entities, we are able to import it to any environment and country. This process consists of three steps:

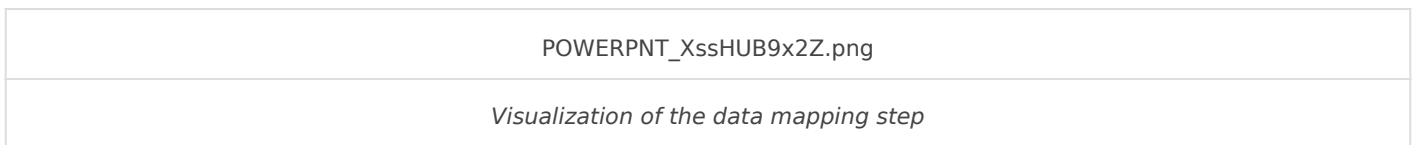
1. Select file - Selection of a .zip file with entities
2. Data mapping - mapping entities to the existing promotions or creating new ones
3. Import - importing the entities and viewing the report.

This is done in:

Import / Export > Import / entities

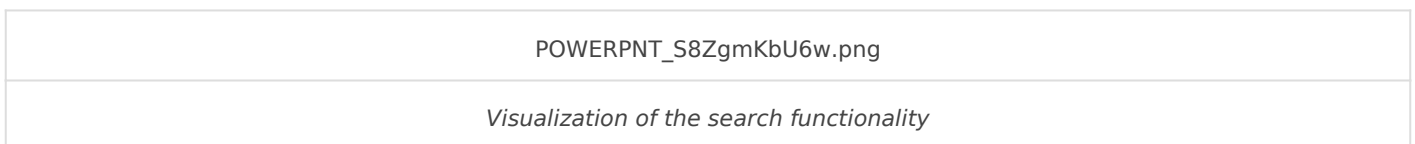


In the import entities tab, we have to choose the previously downloaded .zip file with promotions and then select the **import file** button. This will proceed with the import process to the second step, which is data mapping.

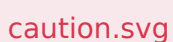


In the data mapping step, there are listed all conditions, actions, and rules that are imported into a selected environment. In this step you can find the following columns:

- **Entity** - Entity type (condition / action / rule), name, and external ID.
- **Status** - Status of the entity after our internal check. If everything is correct the status will be set to **Valid**. If there is an issue with an entity it won't get imported.
- **Local data link** - The system is verifying existing promotions on the environment and if there is a match between the imported one it automatically matches it to the existing one. If no matches were found it will create a new condition / action / rule. In this column users are able to choose if they want to match the import to the existing promotion, and if so which one, or to create a completely new promotion. Two buttons are available:
 - **View** - Is only visible whenever there is a selected match for an entity. This button opens the matched entity, so the user can verify it.
 - **Search** - Allows users to search available promotions and select them, so the system will match the imported promotion to the selected one.



With all entities verified, we can proceed to the third step by selecting the **Accept mapping & import file**.

 **Warning**

Please verify thoroughly the mapping process, as it is not possible to revert the overwritten mapped conditions / actions / rules.

In the last, third step - **Import** we can see the result of the entity import.

POWERPNT_Pfxar8UsSW.png
<i>Visualization of the import step</i>

In this step we can see a table with three columns:

- **Entity** - Entity type (condition / action / rule), name, and external ID. Additionally in this step, we will see **information** regarding the selected entity if some actions couldn't be executed i.e. If the imported entity is imported from PL PROD to CZ PROD there might be a mismatch with used altshops, so in the information tab, we will see information that manual assignment of correct altshops is required for such rule.
- **Status** - Status of the import. If a new entity was **created** it will show Created and if an entity is updating the matched promotion it will say **updated**.
- **Link** - Direct link to created or updated condition / action / rule.

If there are any unexpected errors while importing an entity please contact the Tau Ceti helpdesk at **helpdesk@tauceti.email**.

Examples of popular promotions

Screenshot_2020-04-29_at_19.1

Getting started

In this article, you will find an example of the promotions that are used most frequently on multiple Yves Rocher websites. This article will only show you possible combinations of conditions and actions without the details, of how to create such promotions. You can find articles about condition, action, and rule creation with all their possibilities in the following articles:

- [Promotion module introduction](#)
- [Conditions](#)
- [Actions](#)
- [Rules](#)

Table of contents:

1. [Gift of choice after set cart value is exceeded](#)
2. [Discount on 2 most expensive products when discount code is added to the cart](#)
3. [Free delivery over set cart value](#)
4. [Two products for one when selected products are in the cart](#)
5. [Third product from the list for free when selected discount code is used](#)
6. [Additional FIDEN2 loyalty points when customer exceeds set cart value](#)

Screenshot_2020-04-29_at_19.1

Gift of choice after set cart value

is exceeded

Gift of choice is a list of gifts that customers can choose from. In this example, the customer will be able to do that after his cart value is over 290Kc

To create such a promotion, you have to create an **Order Value** condition. In this example, we used the following JSON:

```
{"prices":"promotion","include_payed_gifts":"TRUE","use_default_excluded_list":"FALSE","consider
```

Additionally, you have to create a **Gift of choice** action with example JSON:

```
{"sku_list": ["F31623","F94615","F01989","250PT","NOGIFT"],"min_cart_value_for_description":"290
```

For a promotion to work, you have to assign the above condition and action to the rule.

POWERPNT_Dvh4jgJLQ4.png

Visualization of the rule configuration of the promotion.

For an **Order value** condition to activate above 290Kc, we have to set the logic operator to **>= (1)** and the value to **290 (2)**. The action doesn't require any operator, or value, as it doesn't use any parameters except the ones written in the JSON.

Below, you can see how a customer sees the **Gift of choice (3)** promotion, whenever he has more than 290Kc of products in his cart.

POWERPNT_jjERXWkywh.png

Visualization of the gift of choice promotion on the website

Screenshot_2020-04-29_at_19.1

Discount on 2 most expensive

products when discount code is added to the cart

One of the most common promotion types uses discount codes as a condition with various actions. In this case, when the customer types the discount code to the cart, his two most expensive products will get discounted by -40%.

Such promotion uses the **Discount code in cart** condition. This condition doesn't use any parameters.

To discount the most expensive products in the cart, we have to create a **discount most expensive product** action. In this example, we will use the following JSON parameters:

```
{"type": "percentage", "prices": "catalog", "exclude_products": ["green_point_flag"], "quantity_limit": 2}
```

For a promotion to work, you have to assign the above condition and action to the rule.

POWERPNT_fMbHqUcB6q.png

Visualization of the rule configuration of the promotion.

For a **Discount code in cart** condition to activate, we have to set the logic operator to = **(1)** and value to desired promotion code **(2)** - in this case, we will use **MAILING11_22points** one. In the action field, we have to specify the percentage discount customers will receive on the two most expensive products, as the type parameter in the action is set to percentage. In this example, we set it to **40 (3)**.

Below you can see how the customer can see this discount in his cart **(4)**.

POWERPNT_PXCfy2CRNd.png

Visualization of the applied promotion

Screenshot_2020-04-29_at_19.1

Free delivery over set cart value

One of the most common promotions in the e-commerce platforms is free delivery. In TauCeti panel, we can set it up in two different ways:

1. Changing the price range for set delivery methods in **Altshops > delivery costs / list**
2. Creating a promotion with a free delivery action

The first method can be read about in the [Delivery costs / list - browsing and managing delivery costs](#) article.

The second method requires creating a rule that contains a selected condition and a free delivery rule. In this example, we will use an **order value** condition with JSON parameters:

```
{"prices":"promotion","consider_discounts":"TRUE","exclude_products":["green_point_flag"]}
```

For an action, we have to create a new action with a **Free delivery** type. This action type doesn't require any JSON parameters.

For a promotion to work, you have to assign the above condition and action to the rule.

POWERPNT_hAEPiQ9ZXV.png

Visualization of the rule configuration of the promotion.

For an **Order value** condition to activate above, for example, 499Kc, we have to set the logic operator to **>= (1)** and value to **499 (2)**. **Free delivery** action doesn't require any operator, or value, as it doesn't use any parameters except the ones written in the JSON.

Below, you can see how customers can see this discount in their cart **(3)**.

POWERPNT_7wq6ddVKJ8.png

Visualization of the applied promotion

Screenshot_2020-04-29_at_19.1

Two products for one when selected products are in the cart

One of the most often promotions used are two for one promotions. Such promotion can be easily configured using the **two products for one** action.

In this example, the promotion will work only when selected products are in the cart. For this, we need to create a condition with **check if SKU(s) exist** condition type. We are using the following JSON:

```
{"sku":["55794","55665","55692","55751","55724","81503"],"required_quantity":"2","consider_quant
```

For an action, we have to create a **two products for one action** with the following JSON:

```
{"allowed_products":["55794","55665","55692","55751","55724","81503"]}
```

In order for a promotion to work, you have to assign the above condition and action to the rule.

POWERPNT_nOBV4peV6a.png

Visualization of the rule configuration of the promotion.

For a **check if SKU(s) exist** condition, we have to set the logic operator to = **(1)** and value to **TRUE (2)**. **Two products for one** action doesn't require any operator, or value, as it doesn't use any parameters except the ones written in the JSON.

Below, you can see how customers can see this discount in their cart **(4)**.

POWERPNT_Pvaq7DBzMM.png

Visualization of the applied promotion

Screenshot_2020-04-29_at_19.1

Third product from the list for free when selected discount code is used

This promotion allows customers to select three products from the list and one of them (usually the cheapest one) will be for free. In order to activate such promotion customers will be required to apply a discount code in the cart.

For conditions, we need a simple **Discount code in cart** condition. This condition doesn't use any parameters.

In order for promotion to apply 3rd product for free to a certain list of products we need a **Set of products** action. Set of products can be used in various different ways according to their parameters. In the case of this promotion, we need the action with the following JSON parameters:

```
{"allow_same_sku": "TRUE", "allowed_products": ["55794", "55665", "55692", "55751", "55724", "81503"], "s
```

In order for a promotion to work, you have to assign the above condition and action to the rule.

POWERPNT_UisQ8C2ZoY.png

Visualization of the rule configuration of the promotion.

For a **Discount code in cart** condition to activate, we have to set the logic operator to = **(1)** and value to desired promotion code **(2)** - in this case, we will use **Discountcode** one. In the action field, we only have to specify a newly created action, as it doesn't require additional values.

POWERPNT_tljaB3mPri.png

Visualization of the applied promotion

Screenshot_2020-04-29_at_19.1

Additional FIDEN2 loyalty points when customer exceeds set cart value

Sometimes it is a good idea to reward customers with more FIDEN2 points whenever they spend more money on their order. In this case, it is possible to use a promotion module to automatically apply such promotion.

For this type of promotion, we require a condition that will activate an action whenever the cart value exceeds the set price point. For this, we can use **Order value** condition with the following JSON parameters:

```
{"prices": "promotion", "consider_discounts": "TRUE", "exclude_products": ["green_point_flag"]}
```

In the above case promotion price of products and other promotional discounts will be counted.

For an Action, we have to use **Add or multiply loyalty points** action. This action requires to set if we want to add or multiply by percentage the loyalty points, so we can do that like in the example below:

```
{"type": "amount"}
```

In this case, we will add a set amount of additional FIDEN2 points to the customer.

For a promotion to work, you have to assign the above condition and action to the rule.

POWERPNT_zdCPCkg3aD.png

Visualization of the rule configuration of the promotion.

For an **Order value** condition to activate above, for example 499Kc, we have to set the logic operator to **>= (1)** and value to **499 (2)**. **Add or multiply loyalty points** action requires to specify the amount of points we want to add to the order. In our case as the action is set to the amount of points if we type for example **500** points, the customer will receive additional 500 FIDEN2 points with their order.

FIDEN2

Screenshot_2020-04-29_at_19.1 Getting started

FIDEN2 is a loyalty point system, where customers can earn FIDEN2 points by purchasing products on the e-commerce platform, reviewing products, or using configured promotions. With enough FIDEN2 points customers are able to use them to receive a one-per-order FIDEN2 voucher **(1)**, which discounts one, selected product in the cart by a selected amount **(2)**. When the customer has more than one of the selected products in the cart the price of the products will be reduced by the selected amount.

FIDEN2 works additionally in retail stores using physical loyalty cards thanks to the JANTAR and WINSHOP integrations, so customers are able to gain FIDEN2 points by purchasing products in the stores locally.

FIDEN2 is a system that works with four separate databases to ensure correct data flow:

- **TauCeti platform** - Stores e-commerce customer data and handles all e-commerce orders. Ensures that customers will receive points for all their e-commerce orders and reviews.
- **JANTAR** - Manages retail cards as well as stores customer card numbers and amount of points. sends to TauCeti platform data with the above-mentioned information.
- **ISAM** - Manages customer personal data assigned to the cards and connects customers to the cards according to their customer data on the TauCeti platform.
- **WINSHOP** - Manages retail transactions and point management whenever FIDEN2 points were used for a discount in the retail store. It additionally initiates retail cards whenever they are created.

All above mentioned systems work together, so we ensure that the FIDEN2 system is omnichannel, which allows customers to have multiple retail and e-commerce FIDEN2 cards and all of them will be assigned to certain customer and have the same amount of points.

POWERPNT_V8x4DbFs4A.png

Visualization of the FIDEN2 location in the cart (desktop)

POWERPNT_9Ums3DfdVR.png

Visualization of the FIDEN2 location in the cart (mobile)

POWERPNT_Mx8BjS6Jtv.png

Visualization of the adding of the discount through FIDEN2 functionality (desktop)

POWERPNT_tASPVHPWe5.png

Visualization of the adding of the discount through FIDEN2 functionality (mobile)

Table of contents:

1. [FIDEN2 system settings](#)
 1. [General](#)
 2. [Loyalty Points](#)
 3. [Webservices](#)
 4. [Balance operations export](#)
 5. [Free cards](#)
 6. [Vouchers](#)
 7. [Notifications](#)
 8. [ISAM file export](#)
 9. [FIDEN2 functions](#)
 10. [FIDEN2 loyalty card registration](#)
 11. [Card Balance Checking](#)
 12. [JANTAR API parameters](#)
 13. [ISAM Parameters](#)
2. [FIDEN2 Card management](#)
 1. [FIDEN2 search card](#)
 2. [FIDEN2 change balance](#)
 3. [FIDEN2 automatic balance operations](#)
 4. [FIDEN2 assign card](#)
 5. [FIDEN2 / Card initialize](#)
3. [FIDEN2 import and export operations](#)
 1. [Import points](#)
 2. [Export cards](#)
4. [FIDEN2 report](#)
5. [Configuring the FIDEN2 promotion](#)

Screenshot_2020-04-29_at_19.1

FIDEN2 system settings

All necessary tabs to configure the FIDEN2 functionality can be found in

System > System / Settings > FIDEN2 (1)

In the FIDEN2 tab, you can find the necessary settings (2). Each of them and their options are described in the chapters below.

POWERPNT_mUxeQgSfjq.png

<i>Visualization of the FIDEN2 tab in System / settings</i>

General

In the general tab, you can find general functions described below

POWERPNT_fa5HiCFn6t.png

<i>Visualization of the General tab</i>

- **Functionality enabled** - Switch that decides if the FIDEN2 function is enabled.
- **Debug mode** - Enables debug mode for testing purposes. NEVER use on the production environment, as this mode irreversibly anonymizes customer data.
- **Header icon enabled** - Enables a header icon that redirects customers to the NewLoyaltyClub page. The page URL can be set in [Loyalty Points](#) tab

Loyalty Points

In the Loyalty Points tab, you can set values connected to loyalty points and their earnings.

POWERPNT_upn7mWZnLn.png

Visualization of the Loyalty points tab

- **Amount of points for each 1Kc** - Amount of points customers will receive for every currency spent on the website. For example, if customers buys 700kc worth of products, they will receive 700 FIDEN2 points.
- **Currency base** - Low-level setting, that changes the number of points customers will receive per spent price. This value works with a mathematical operation:

$(\text{Total price} / \text{currency base}) * \text{multiplier}$

So, for example, changing this value to 2, and the multiplier to 1 will result in customers receiving a number of points equal to 50% of their order value.

- **Points Lifetime** - Value, which sets the lifetime of the points in days. If this setting is set to 365 it means that points earned in 01.01.2021 will disappear from the customer's account 01.01.2022.
- **Info page url** - URL of the page that customer is redirected to, by icon in the header.
- **Amount of points for review** - Amount of points customers will receive per product review.

Webservices

In the webservice tab, you are able to change the WSDL URL.

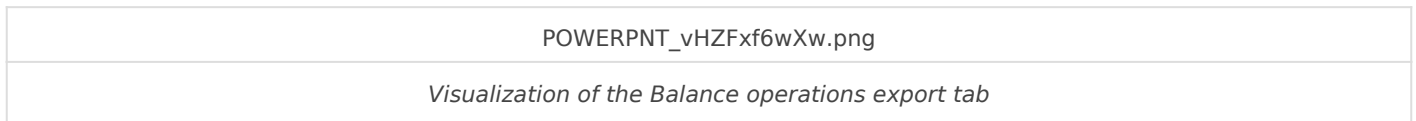
POWERPNT_ubKu9N4nKd.png

Visualization of the Webservices tab

- **WSDL URL** - Jantar service URL. This setting should be left on a default URL.

Balance operations export

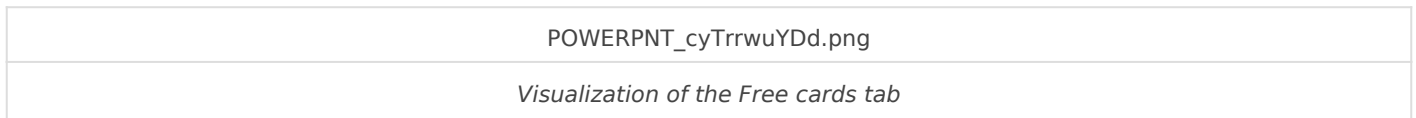
In the Balance operations export, you can change the operations settings described below



- **Delay for point gain since order placement** - Delays the points that customers will receive by X days from the date of order placement.
- **Count of tries if export fails** - The number of tries the system will try to assign FIDEN2 points to the customer before it stops the action.
- **Notify about failed operations** - A switch that when turned on sends daily e-mails (once per day) to all local administrators and business admins with information on how many FIDEN2 operations have failed in the last export. It doesn't send e-mails to GAC accounts.

Free cards

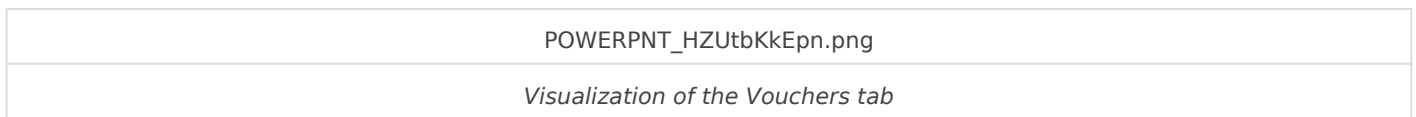
In the Free cards tab, you can set the following option:



- **Reserve size** - Count of cards that will be initialized and ready for assignment to customers.

Vouchers

In the Vouchers tab, you are able to configure the values and cost of the FIDEN2 promotion.



- **Configuration** - JSON field, which configures the FIDEN2 promotion visible in the cart. JSON must contain:
 - **"name"** - Visible name of the option.
 - **"value"** - How much of a currency will be discounted. For example "100.00"
 - **"points_cost"** - How many FIDEN2 points customers will lose by selecting the discount option in the cart.
 - **"description"** - description of the option. This field is not visible to the customer.

Example of the JSON:

```
[{"name": "150Kč", "value": 150.00, "points_cost": 2000, "description": "Věrnostní kupon"}, {"name": "400Kč", "value": 400.00, "points_cost": 4000, "description": "Věrnostní kupon"}]
```

- **Lifetime** - A lifetime in days of a coupon added to the cart. For example, if a customer adds a coupon to the cart and then leaves the website the coupon will be in the cart for another X days. After X days the coupon will get canceled and the customer will receive points back.

Notifications

The notifications tab allows configuring if customers should receive notifications regarding their FIDEN2 card.

POWERPNT_gm2bNEPzQ6.png

Visualization of the Notifications tab

- **Send SMS after card assignment in WAITING status** - Switch that decides if the customer receives an SMS message after their card receives WAITING status.
- **Send SMS after car assignment in ASSIGNED status** - Switch that decides if the customer receives an SMS message after their card receives ASSIGNED status.

ISAM file export

The ISAM file export settings allow configuring the following option:

POWERPNT_y3EvGQpBdJ.png

Visualization of the ISAM file export tab

- **Include agreements** - The setting that decides if ISAM customer export to ISAM should include their agreements.

FIDEN2 functions

In the FIDEN2 functions tab, you can set up the basic FIDEN2 functionalities described below

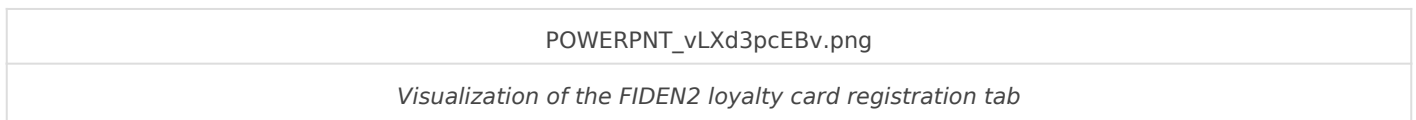
POWERPNT_d89Z7h9ZYT.png

Visualization of the FIDEN2 functions tab

- **Points enabled** - Switch, which enables the FIDEN2 points functionality. With this option disabled, customers won't be able to use FIDEN2 discounts and won't earn FIDEN2 points
- **Customer register card enabled** - Switch, which enables the creation of new FIDEN2 cards whenever a new account is created on the website. Without a FIDEN2 card, customers won't be use FIDEN2 functionalities.
- **Vouchers: exclude discounted products** - Switch, which when enabled, excludes already discounted products from the FIDEN2 discount functionality.
- **Fidelite CSV files import enabled** - Switch, which enables import of the Fidelite .csv file sent from ISAM.

FIDEN2 loyalty card registration

In the FIDEN2 loyalty card registration, you can turn ON/OFF an option for customers that will allow them to assign their retail FIDEN2 cards to their newly created accounts during the registration process in the mobile app.

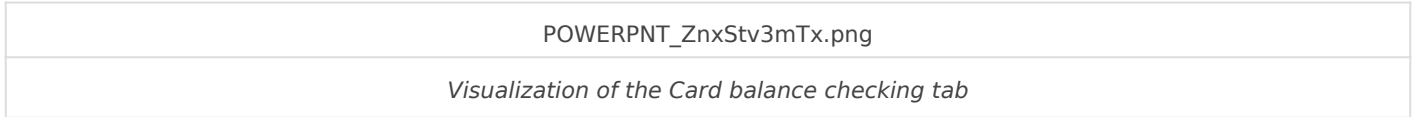


In this tab you can set up the following options:

- **Enable loyalty card registration on account creation form** - A switch that enables the functionality. When it is turned on customers will have an additional field in the registration form in the mobile app that will allow them to type their retail FIDEN2 card number and confirm it. The selected FIDEN2 card number must have a mobile number assigned to it in ISAM.
- **Max IP address requests per hour** - API setting, which defines how many IP address requests are allowed per hour before blocking the user.
- **Max IP address requests per day** - API setting, which defines how many IP address requests are allowed per day before blocking the user.
- **Max card number requests per hour** - API setting, which defines how many card number requests are allowed per hour before blocking the user.
- **Max card number requests per day** - API setting, which defines how many card number requests are allowed per day before blocking the user.
- **Max nonexistence card number requests per day** - API setting, which defines how many nonexisting in the database card number requests are allowed per day before blocking the user.
- **Max token check requests per day** - API setting, which defines how many token requests are allowed per day before blocking the user.
- **Max token check requests per hour** - API setting, which defines how many token requests are allowed per hour before blocking the user.

Card Balance Checking

In the card balance checking, you can change the following option:



- **Enabled** - Disables or enables pooling checking balance for cards that needs to be checked as a result of registered Webhooks

JANTAR API parameters

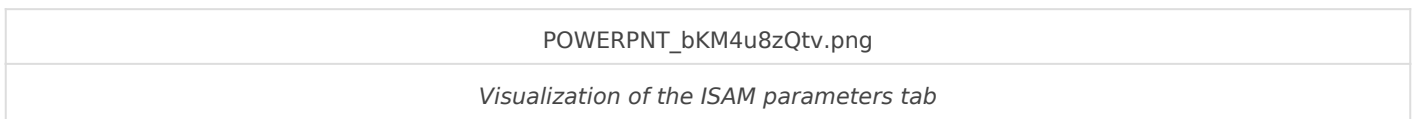
In the JANTAR API parameters, you can set up the following options:



- **Country ID** - ID of a country that is used in the JANTAR application.
- **Commercial Language ID** - ID of a commercial language that is used in JANTAR.

ISAM Parameters

In the ISAM parameters, you can set up the following options:

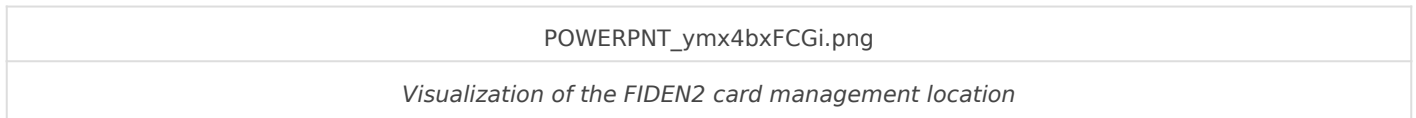


- **Origine shop nb** - A value that is placed in the TctoISAM file as origine shop nb.
 - **Export customers without address** - Switch, which decides if customers should have their main address exported to ISAM.
-

Screenshot_2020-04-29_at_19.1

FIDEN2 Card management

Customers cards can be verified, and their balance can be changed in the admin panel. Whole card management is located in the **Customers** tab **(1)**.

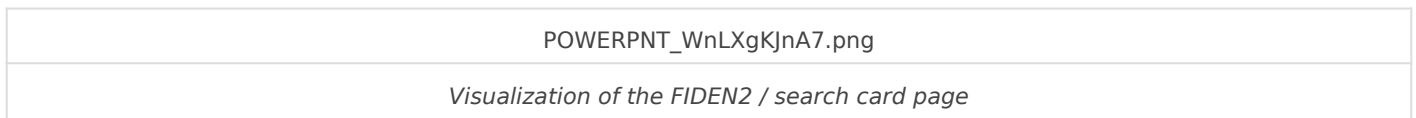


FIDEN2 search card

You are able to search for a card by its number in the FIDEN2 / Search card tab, which is located in:

Customers > FIDEN2 / Search card

Below you will find all possible fields in the FIDEN2 / Search card tab.



- **FIDEN2 / Search card** - a field where you have to type the FIDEN2 card number. Without the card number, other options won't be visible.
- **General information** - Contain general information about selected FIDEN2 card:
 - **Exists in FIDEN2 API** - A Yes/No field that returns the value if the selected card exists in the FIDEN2 API.
 - **Exists in e-shop database** - A Yes/No field that returns the value if the selected card exists in the e-shop database.
 - **Card type** - Type of a card. There are two types:
 - **Internet** - a card that has been assigned during the account creation on the e-shop.
 - **Retail** - a card that has been created and assigned in the physical retail store.
 - **API export status** - Status of the API export
 - **Is free?** - a yes/no field that shows if the selected card is assigned to any customer or not.
 - **Mobile (from ISAM)** - Customers mobile numbers imported from ISAM.
- **Data from API** - Contains card balance on FIDEN2 API.
- **Assigned customers** - List of customers that have selected cards assigned to them. This field contains columns:
 - **ID** - Customer internal ID
 - **E-mail** - Customers e-mail address

- **Mobile** - Customers mobile number
- **Status** - Assignment status.
- **Cards using the same loyalty account** - a list of FIDEN2 cards that assigned customers have also assigned to them.
- **Operations (from API)** - A list of operations taken from API.
- **Operations (from Fidelite Operation files)** - A list of operations taken from Fidelite import file provided by JANTAR.

Below you will see all fields in the Operations tabs.

POWERPNT_kXyd59UiMN.png
<i>Visualization of the operations tab</i>

- Select the **Show X entries(1)** field to set the number of operations, displayed on one page (min 10 / max 100).
- The names of the columns **(2)** that contain information about target operations. Each column is susceptible to sorting.

To sort the records, click on the title of the column in question.

- **Date** - A date and hour of an operation.
- **Card number**- The card number used in the operation
- **Point type** - A type of points. There are two point types:
 - **AMT** - Standard points for an order. This value is also visible with a point burn
 - **BON** - Bonus points. This value was used in the past. Currently, all values are set to AMT point type.
- **Operation type - Type of operation. There are two operation types:**
 - **BEN** - Gain points
 - **TKB** - Burn points
- **Points change** - The number of points added, or removed from the card.
- **Origin** - The channel from which the point transaction originated from. There are currently two origin types:
 - **SHP** - Retail stores
 - **NET** - E-commerce e-shop
- **Reference part 1** - A first reference number taken from JANTAR.
- **Reference part 2** - An second reference number taken from JANTAR.
- **Keypuncher** - A keypuncher value is taken from JANTAR. This value is usually constant.
- **Search box (3)** where you can search for featured products list by their card name, reference part 1, 2 and keypuncher.
- **Page buttons (4)**, which allow users to change the page of currently viewed operations.

FIDEN2 change balance

You are able to manually change certain card balances in:

Customers > FIDEN2 / Change card balance

Below you will find all possible fields in the FIDEN2 / Change card balance tab.



- **Card number** - a field where you have to type the FIDEN2 card number. Without the card number, other options won't be available.
- **Balance** - Current balance on the card. This field cannot be edited.
- **Waiting points** - The current amount of points that are waiting to be assigned to the card. This field cannot be edited.
- **Operation type** - Option that allows selection of if points should be added or removed from the current balance.
- **Points count** - The number of points that will get added / removed.
- **Change reason** - A reason why points have been added or removed.
- **Balance after change** - Displays a balance after adding / removing the points. This field cannot be changed.

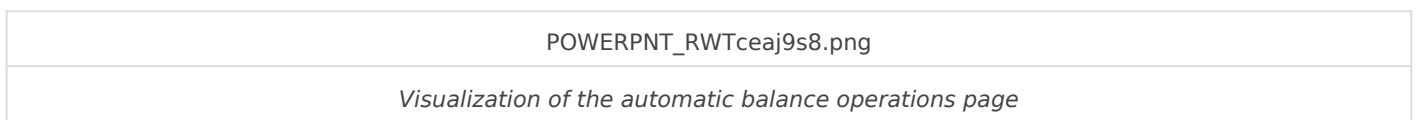
With all fields filled you are able to confirm the operation using the **Save points operation** button.

FIDEN2 automatic balance operations

Automatic balance operations contain all FIDEN2 automatic operations, like points gained because of the newly created order. You can access FIDEN2 automatic balance operations in:

Customers > FIDEN2 / Automatic balance operations.

Below you will find a description of all visible fields.



- Select the **Show X entries(1)** field to set the number of balance operations, displayed on one page (min 10 / max 100).
- The names of the columns **(2)** that contain information about target balance operations. Each column is susceptible to sorting.
To sort the records, click on the title of the column in question.
 - **ID** - An internal ID of the balance operation.
 - **Card number** - A card number associated with the selected operation.

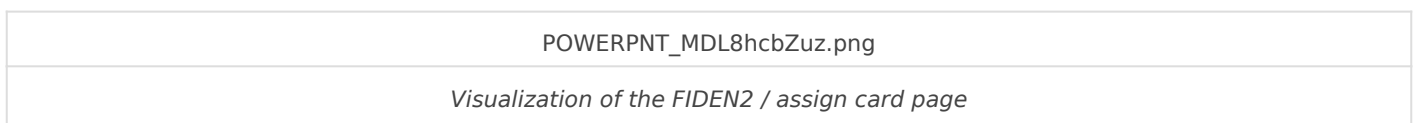
- **Reference** - Reference number where that operation occurred. It can be an e-shop order number, which can be entered, or a reference number for use in the retail store.
- **Type** - Type of operation. Available types are **Gain points(+)** and **Burn points(-)**.
- **Value change** - Value by how many points get added or removed from the card.
- **Status** - Status of the operation.
- **Operation time** - The time when the operation occurred.
- **Creation time** - The time when the operation was created.
- **Search box (4)** where you can search for balance operations by their card number and reference number.
- **Page buttons(5)**, which allow users to change the page of currently viewed operations.

FIDEN2 assign card

You are able to assign created FIDEN2 cards to selected customers in

Customers > FIDEN2 / Assign card

Below you will find a description of all visible fields.



- **Card number** - a field where you have to type the FIDEN2 card number.
- **Customer** - A search field for an e-shop customer. You can search for customers by their e-mail address, name, and surname or mobile number.

With both fields filled you can assign the selected FIDEN2 card to the selected customer by pressing the "**I confirm data validity**" button.

FIDEN2 / Card initialize

You are able to initialize a new FIDEN2 card number in the

Customers > FIDEN2 / Card initialize

Below you will find a description of all visible fields.



- **Card number** - a field where you have to type the FiDEN2 card number that is not in the system and FIDEN2 API.
- **Card status** - A status check, that shows if the selected card number is taken or not. If the number is taken by another card it cannot be initialized.

With a valid card number entered that isn't yet initiated you can create a new e-commerce FIDEN2 card using the **Create card** button.

Screenshot_2020-04-29_at_19.1

FIDEN2 import and export operations

Import points

You are able to import FIDEN2 points using the .xlsx file to the selected cards

You will find FIDEN2 points import in the:

Import/Export > Import / FIDEN2 points

POWERPNT_jgjuuVYU3z.png

Visualization of the FIDEN2 / card initialize page

In this tab you will find fields:

- **File** - button which allows to select the .xlsx file from the computer.
- **Reason** - Reason for a points change
- **Options** - optional test import tick box, which will test the imported file without doing any changes to the customer points.

With all necessary fields filled, you can import the FIDEN2 points using the **import file** button

POWERPNT_dzTERmCWHp.png

Visualization of the FIDEN2 points import file

Export cards

You are able to export all FIDEN2 cards to the .csv file in the:

Import/Export > Export / FIDEN2 cards

POWERPNT_UbDpA37URN.png

Visualization of the FIDEN2 / card initialize page

In the Export / FIDEN2 cards you can select a **card type**, and export all the cards with a selected type to the .csv file using the **Export** button.

POWERPNT_sTJL8gmK6y.png

Visualization of the FIDEN2 / card initialize page

The exported .csv file contains:

- **Card number** - Customers FIDEN2 card number
- **Type** - Type of a FIDEN2 card
- **Mobile (from vpm_customers)** - Customers mobile number taken from ISAM.
- **Status** - FIDEN2 card status
- **Customers** - e-mail address assigned to the selected FIDEN2 card.

Screenshot_2020-04-29_at_19.1 FIDEN2 report

FIDEN2 system has its own report, where you can see statistics about the voucher and points.

You can find FIDEN 2 report in the:

POWERPNT_N34vHHS3bs.png

Visualization of the FIDEN2 / card initialize page

In the FIDEN2 report you can find four tabs:

- **Generated vouchers** - the amount of generated available vouchers. In this tab you can find columns with:
 - **Overview** - The type of generated FIDEN2 vouchers.
 - **Number** - The number of generated FIDEN2 vouchers.
- **Already used vouchers** - The number of used vouchers. In this tab you can find columns:
 - **Overview** - The type of used FIDEN2 vouchers.
 - **Number** - The number of used FIDEN2 vouchers.
- **Points gains by origin (last 30 days)** - The origin from which customers gained points. In this tab you can find columns:
 - **Overview** - Type of origin. Available origins are **NET** and **SHP**.
 - **Number** - Number of points gained.
 - **Transactions** - Number of transactions.
 - **Avg Points/Transaction** - Average points gained per transaction.
- **VPI Points gains by status** - The number of points gained by their status. In this tab you can find the following columns:
 - **Overview** - Type of gain.
 - **Number** - Number of points gained
 - **Transactions** - Number of transactions
 - **Avg Points/Transaction** - Average points gained per transaction.

Screenshot_2020-04-29_at_19.1

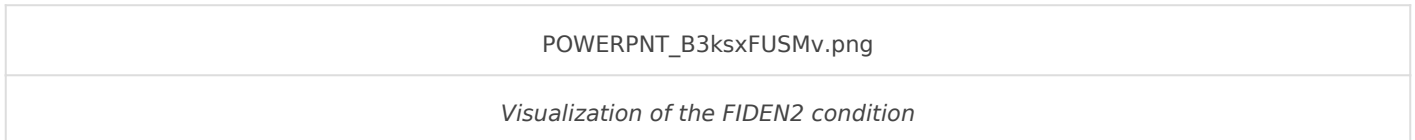
Configuring the FIDEN2 promotion

In order for a FIDEN2 system to work properly on the front end for the customers the target environment has to have configured promotion rule.

info2.svg **Remember**

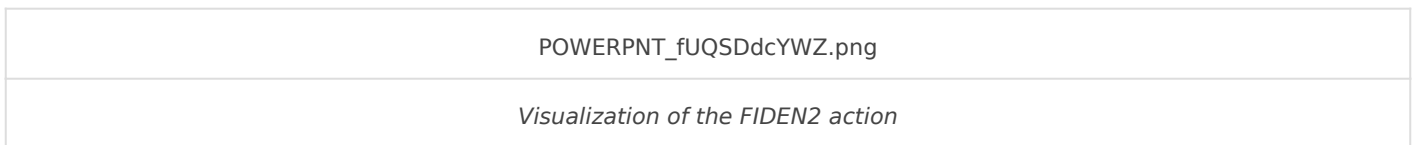
You can find more information about the promotional module and how to configure promotions in the [Promotion module introduction](#) article.

To create a FIDEN2 promotion first you have to create a FIDEN2 condition. Condition name and description can be set to whatever fits the user as it's not visible on the front end. You can learn more about promotion module conditions in the [Conditions](#) article.



FIDEN2 condition has to be set to type **Voucher in the cart**. Parameters can be left empty, as FIDEN2 parameter is a default one.

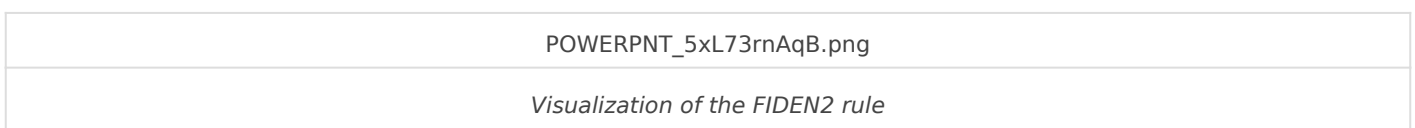
With the created condition, we have to create an action for a FIDEN2 promotion. Action name can be set to whatever fits the user as it's not visible on the front end. You can learn more about promotion module actions in the [Actions](#) article.



In the FIDEN2 action we have to set the type of action to **Apply voucher discount**. Here you can configure parameters according to your needs. Available parameters are:

- **use_default_excluded_list:** TRUE (default) / FALSE - use the global excluded SKU list
- **include_payed_gifts:** FALSE (default) / TRUE - apply discount also to paid gifts
- **exclude_products:** array, f.x. ["12345","12346"] - list of SKUs that should be excluded from this discount. You can use "green_point_flag", "bestseller_flag", "limited_edition_flag", "set_flag" and "outlet_flag" as SKU which will include all SKUs with selected flag into an excluded list. F.x. ["green_point_flag", "set_flag", "01234"].
- **allowed_products:** array, f.x. ["12345","12346"] - list of SKUs which CAN be discounted, it disabled excluded products settings (only products from this list can be discounted).

With created condition and action we can make them work together by creating a new rule. You can learn more about creating promotion module rules in the [Rules](#) article.



In the FIDEN2 rule, you have to set **Start date** and **End date** within a broad range of dates, so the promotion doesn't expire anytime soon. Please remember to also be in the **Alt shops** list to enable

the rule in all desired altshops, which customers have access to.

It is important to set **Sort order** to 0, so the FIDEN2 discount will be calculated and applied before any other promotion in the system. FIDEN2 is the only promotion type that can have sort order set to 0.

In the Conditions tab select your created FIDEN2 condition and set it to = **TRUE**, then in the actions tab select your FIDEN2 action, which doesn't take any other input.

With a properly configured rule, you can save it using the "**Save**" button, and the promotion rule should work right away on an e-commerce platform allowing users to use their FIDEN2 points on products in the cart.

Discount codes

Screenshot_2020-04-29_at_19.1 Getting started

In this article, you will learn about the discount codes. Discount codes are codes that are used by the customers in the cart on the website **(1)** in order to get benefits from the configured promotional rules. We distinguish two types of discount codes:

1. **Main code** - a custom-made code created by the administrator with unlimited amount of uses (unless specified otherwise in the promotional rule).
2. **Personal code** - a code assigned inside the main code with one time use. When the main code has assigned at least one personal code, the main code itself doesn't work, and only the personal codes can be distributed among customers for it to work.

info2.svg **Remember**

Imported discount codes that were imported without assigning them to the main codes are added to the public personal code pool, which can be used to add personal codes to the selected main codes. You can check the current amount of public personal codes in the Reports > personal discount codes. You can find more about assigning public personal codes in the [Personal codes](#) chapter.

POWERPNT_LKXEPV5x53.png

Visualization of discount code field in the cart

To open the Featured products / list view, in the admin panel, navigate to:

Promotions > Discount codes / list

All possible operations to perform on the **Discount codes / list** view can be done in the **Filters (2)**, and **Discount codes / list (3)** tabs.

POWERPNT_TXgbuhNkLZ.png

Visualization of the discount code / list

Table of contents:

1. [Filters section](#)
2. [Discount codes / list](#)
3. [Creating a new discount code](#)
4. [Personal codes](#)
 1. [Generating new personal codes](#)
 2. [Importing of the personal codes](#)
5. [Discount codes cart translations](#)

Screenshot_2020-04-29_at_19.1

Filters section

The **Filters** section allows sorting of the discount codes in the promotion module, based on the available filters. Below you will find a description of available filters

POWERPNT_bjebc8Kxtr.png

Visualization of the filters tab in the discount codes / list module

- **Is active?** - Featured products list activity status.
- **Personal code search** - A tick box that allows to search for personal codes inside the main codes, when using the search functionality in [discount codes / list](#).

After defining the above filters, select the **Apply Filters** button to get a list of sites according to the applied filters.

Screenshot_2020-04-29_at_19.1

Discount codes / list

The **Discount codes / list** section allows organizing the filtered discount codes, edit them and create new ones, delete them, and search for the discount codes.

Below you will find a description of all possible actions you can perform in the discount codes / list section.

POWERPNT_w6B54vjvKH.png

Visualization of the discount codes / list section in the discount codes / list

- Select the **Show X entries (1)** field to set the number of discount codes, displayed on one page (min 10 / max 100).
- The names of the columns **(2)** that contain information about a target discount codes. Each column is susceptible to sorting.
To sort the records, click on the title of the column in question.
 - **ID** - An internal ID of the main code.
 - **Name** - The main code name. This name can be used to apply the promotion, if the promotion is set to use a main code, unless the selected main code has a personal code assigned to it. When a main code has assigned personal codes to it by writing the main code into the promotion, it will take personal codes from that main code.
 - **Is active?** - Activity status of the main code.

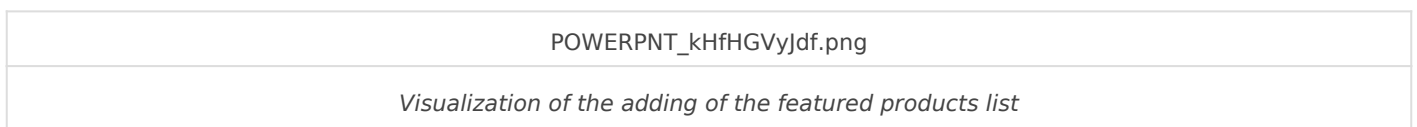
- **Actions** - List of buttons with possible actions that can be done to the desired main code:
 - **Edit** - Edit desired discount code. As editing has exactly same actions as creating the new code, you can find more about editing in the [Creating a new discount code](#) chapter.
 - **View** - Allows to view the content of the main code without the ability to edit any fields.
 - **Import** - A button that redirects to **import / export > import / discount codes**. In this module, it is possible to import personal codes to the selected main code. More about importing codes can be found in the [Importing of the personal codes](#) chapter.
 - **Delete** - Deletes the main code without the ability to restore it.
- Select the **Add (3)** button to go to the adding the new main code. You can learn more about adding of the discount codes in the [Creating a new discount code](#) chapter.
- **Search box (4)** where you can search for discount codes by their system name, or if the **Personal code search** tick box is selected in the [Filters section](#), it also searches for a personal codes assigned to the main codes.
- **Page buttons (5)**, which allows user to change the page of currently viewed discount codes.

Screenshot_2020-04-29_at_19.1

Creating a new discount code

Add button allows users to create a new main code. During the creation of the main code, you won't have access to the Personal codes tab **(1)**. This tab can be only accessed while editing the main code. You can read more about personal codes in the [Personal codes](#) chapter

Below, you will find a description of all possible fields that can be edited while creating a new main code.



- **Discount code** - The main code name. This name can be used to apply the promotion, if promotion is set up to use main code, unless the selected main code has a personal code

assigned to it.

- **Mailing code** - Value of the discount code. This field is exported to ISAM inside MAILING_KEY field. It can contain up to 6 characters.
- **Is active?** - Activity status of the main code.

info2.svg **Remember**

ISAM is an external system that manages customer personal data used in the FIDEN2 system, and connects customers to the cards according to their customer data on the TauCeti platform. You can find more information about FIDEN2 functionality in [FIDEN2](#) article.

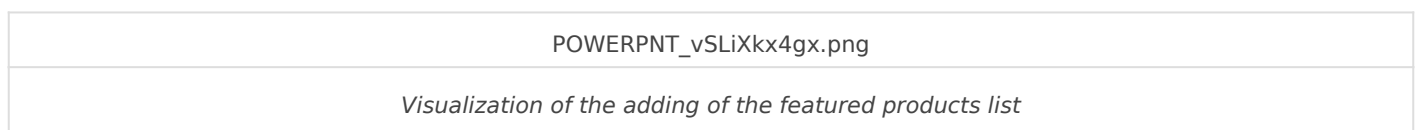
With all necessary fields filled, you can save the changes using the **save** button.

Screenshot_2020-04-29_at_19.1

Personal codes

Personal codes are codes assigned inside the main code with one time use. When the main code has assigned at least one personal code, the main code itself doesn't work, and only the personal codes can be distributed among customers for it to work.

Below you will find a description of all possible actions you can perform in the personal codes section and available fields.



In the **Assign new personal code** tab we can find information about main code how many personal codes are assigned to it, how many are reserved by the customers and how many of them are used. Except that we can assign personal codes from the public personal code pool by typing the quantity **(1)**, which we want to assign and press the **Assign** button **(2)**.

In the **Personal codes (main code)** tab, you can find:

- Select the **Show X entries (3)** field to set the number of discount codes, displayed on one page (min 10 / max 100).
- The names of the columns **(4)** that contain information about a target discount codes. Each column is susceptible to sorting.

To sort the records, click on the title of the column in question.

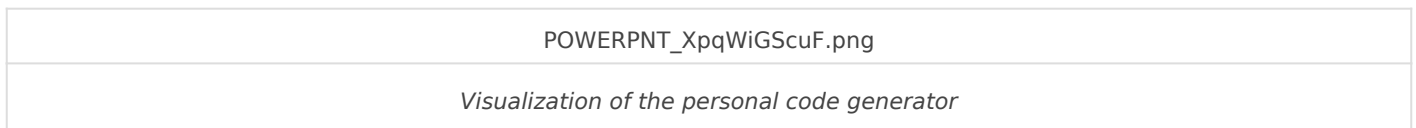
- **ID** - An internal ID of the main code.
- **Discount code** - The personal code value. This is a code that customer uses in the cart in order to get the promotion.
- **Reserved** - status that shows if the selected personal code has been reserved to the customer or not. Reserved personal codes cannot be assigned to other customer.
- **Used on** - A date when the selected personal code was used.
- **Customer** - Customers e-mail addresses, that code is assigned to.
- Select the **Export (5)** button to export all personal codes assigned to the selected main code.
- **Search box (6)** where you can search for discount codes by their name.
- **Page buttons (7)**, which allows user to change the page of currently viewed discount codes.

Generating new personal codes

In the Generator / discount codes module, you are able to generate personal codes and download them in the .txt file.

Personal codes can be generated in a high volume in

Import / export > Generator / discount codes



In this tab, you can select:

(1) Personal codes number - Number of generated personal codes.

(2) Personal code min length - The minimum length of the generated code

(3) Personal code max length - The maximum length of the generated code

With all fields filled, you can generate the codes by pressing the **Generate and export** button **(4)**. You will be able to download the .txt file with all the codes in it. These codes can be imported to the public pool of the codes, or to the specific main code. More about importing of the personal codes can be found in the [Importing of the personal codes](#) chapter.

Importing of the personal codes

Generated personal codes, located in the .txt file can be imported to the public pool of personal codes, or to the selected main code. You can import personal codes in the:

Import / export > Import / discount codes

POWERPNT_nmijBU8D5L.png
<i>Visualization of the personal code import</i>

In the import / discount codes section, you can:

- Select a .txt file to import by pressing the **browse...** button **(1)**.
- Select a main code to import the personal codes into **(2)**. If this field is empty the codes will be imported to the public personal code pool.
- Select the tick box to do a test import **(3)**. Test import checks the correctness of the file without adding the selected personal codes.

By pressing the **Import file** button **(4)**, you will import the .txt file.

Screenshot_2020-04-29_at_19.1

Discount codes cart translations

When customers adds, or tries to add a discount code to the cart he will see one of the pop-ups with a message adequate to current promotion or discount code state. The content of these pop-ups can be set in the:

System > Translations / list

info2.svg **Remember**

You can learn more about translations in the **[UNDER CONSTRUCTION]** article.

Currently, there are four pop-up messages that customer will be able to see in the cart:

1. **Correct discount code** - This pop-up will appear when promotion is active, has correct start and end dates and not used personal code. This message can be found under phrase **Discount code added to cart**

POWERPNT_B7GVZfhPK9.png

Visualization of the correct discount code message

2. **Incorrect discount code** - This pop-up occurs as the promotion is not active, has a start date in the future or is inactive in the selected altshop. This message can be found under phrase **Improper discount code removed from cart**

POWERPNT_dVV2h3xsFZ.png

Visualization of the incorrect discount code message

3. **Inactive discount code** - It occurs as the end date of the promotion is in the past, so promotion is no longer valid. This message can be found under phrase **Promotion with this discount code is no longer valid**

POWERPNT_d6YCfm5ujj.png

Visualization of the inactive discount code message

4. **Used personal code** - This message occurs as the given personal code has already been used and the promotion is still active. This message can be found under phrase **You provided improper discount code**

POWERPNT_26ZrGWJN6Z.png

Visualization of the used personal code message

Promotion pages

Screenshot_2020-04-29_at_19.1 Getting started

Promotion pages are easy to make and maintain pages with a list of products, where customers are able to select the desired products and add them separately to the cart on the website.

Promotion pages are usually created in order to hold categories of products that have discounts assigned in the [Products / list - browsing and managing products in altshop module](#).

info2.svg **Remember**

You can learn about the promotion module in the [Promotion module introduction](#) article.

info2.svg **Remember**

In order for the promotion page to be visible on the website, it needs to be activated in the desired altshops. You can find more about activating the promotion pages in the [Promotion pages / list - browsing and managing promotion pages in altshop module](#) article

POWERPNT_7CndCRysHV.png

Visualization of the promotion page (desktop view)

POWERPNT_LMNvFDedhV.png

Visualization of the promotion page (mobile view)

To open the Promotion pages / list view, in the admin panel, navigate to:

Promotions > Promotion pages / list.

In the Actions / list view, we can find two tabs: **Filters (1)** and **Actions / list (2)**

POWERPNT_UZ7T5k2JuN.png

Visualization of the promotion pages / list section in the administrator panel

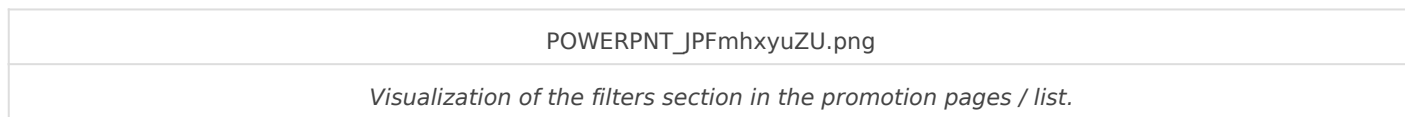
Table of contents:

1. [Filters section](#)
 2. [Promotion pages / list section](#)
 3. [Adding a new promotion page](#)
 4. [Editing of a promotion page](#)
 5. [Assigning the CMS block to the promotion page](#)
-

Screenshot_2020-04-29_at_19.1

Filters section

The **Filters** section allows sorting promotion pages by using the available filters.



- **Is active?** - Filters by the "**is active?**" status of the promotion page.

After defining the above filters, select the **Apply Filters** button to get a list of pages according to the applied filters.

Screenshot_2020-04-29_at_19.1

Promotion pages / list section

The **Promotion pages / list** section allows organizing the filtered pages, edit them, and add the new ones.

Below you will find a description of all possible actions you can perform in the promotion pages / list section.

- Select the **Show X entries (1)** field to set the number of promotion pages, displayed on one page (min 10 / max 100).
- The names of the columns **(2)** that contain information about a target promotion page. Each column is susceptible to sorting. To sort the records, click on the title of the column.
 - **ID** - An internal ID of the promotion page.
 - **Name** - The name of the promotion page. The value is displayed on the e-commerce platform for the customer in the breadcrumb.
 - **Is active?** - Promotion page activity status.
 - **Start date** - date from which the promotion page will be visible on the e-commerce platform for the customers.
 - **End date** - date from which the promotion page will not be visible on the e-commerce platform for the customers.
 - **Actions** - List of buttons with possible actions that can be done to the desired page:
 - **Edit** - Edit desired promotion page. Editing allows editing the promotion page parameters and changing the products visible on the promotion page. You can find more about editing the promotion page in the [Editing of a promotion page](#) chapter.
 - **View** - Allows to view the content of the promotion page without the ability to edit it.
 - **Delete** - Deletes the promotion page without the ability to restore it.
- Select the **Add (3)** button to go to the adding the new promotion page. You can learn more about adding of the CMS blocks in the [Adding a new promotion page](#) chapter.
- **Search box (4)** where you can search for desired promotion page by using their name.
- **Page buttons (5)**, which allows user to change the page of the promotion pages list.

Screenshot_2020-04-29_at_19.1

Adding a new promotion page

Add button allows users to create a new promotion page. While creating a new promotion page, you can only set key features about the promotion page, without the ability to add or remove products. In order to add or remove products to the promotion page, you have to edit the promotion page. More about editing of the promotion pages can be found in the [Editing of a promotion page](#) chapter.

Below, you will find a description of all possible fields that can be edited while creating a new promotion page.

POWERPNT_cP53DfMpvx.png

Visualization of the promotion page creation.

- **Name** - The name of the promotion page. The value is displayed on the e-commerce platform for the customer in the breadcrumb.
- **Friendly URL** - URL address where the promotion page is located.
- **Is active?** - Promotion page activity status.
- **Start date** - date from which the promotion page will be visible on the e-commerce platform.
- **End date** - date from which promotion page will not be visible on the e-commerce platform.
- **Short description** - No longer supported field, as the CMS blocks have replaced it. You can find more about adding the CMS blocks to the promotion pages in the [Assigning the CMS block to the promotion page](#) chapter.
- **Display colors as separate products** - A switch that decides if the products with multiple colors should be displayed separately or not. This switch can only be set when creating a promotion page, and cannot be changed later.
- **Default product sort order** - Default sort order of the products in the list. You can select:
 - **Newest** - Products will be sorted by their add date.
 - **Customer order** - Products will be sorted by their sort order in the products field.You can read more about products field in the [editing of a promotion page](#) chapter.
- **Meta title** - Title of the page used in the metadata.
- **Meta keywords** - Keywords for a specific promotion page used in the metadata.
- **Meta Description** - Description of the page used in the metadata
- **Description** - No longer supported field, as the CMS blocks have replaced it. You can find more about adding the CMS blocks to the promotion pages in the [Assigning the CMS block to the promotion page](#) chapter.

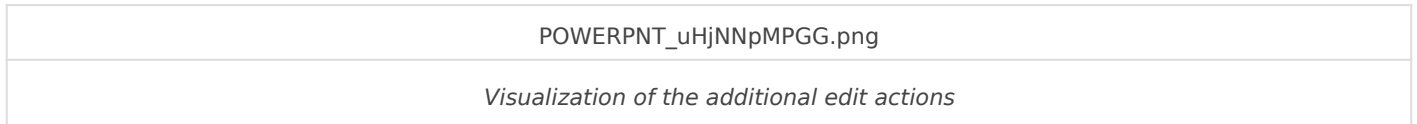
With filled necessary fields, users can save the changes using the **save** button to save the changes and go back to the pages list, or **save and continue** button which will save the changes and redirect the user to the [editing of a promotion page](#).

Screenshot_2020-04-29_at_19.1

Editing of a promotion page

As products cannot be added during the creation of the promotion page they have to be configured using the edit functionality.

Below, you will find a description of all possible fields.



- **Add product** - in this tab, you can search for a single product by its SKU and add it to the promotion pages product list.
- **Add many products** - In this tab you can add multiple products to the promotion page product list by typing their SKUs separated by comma (for example 12345,54321,55443)
- **Remove many products** - In this tab you can remove multiple products to the promotion page product list by typing their SKUs separated by comma (for example 12345,54321,55443)
- **Products** - A list of products that are currently assigned to the promotion page. They can be moved by using the "**Drag and drop**" functionality, and their order will be visible on the website when the "**customer order**" sort order is selected. Products that are out of stock or inactive will have corresponding information next to their name. You are able to remove the products using the **remove** button.

In order to save the changes, press the **Save** button above the add product tab.

Screenshot_2020-04-29_at_19.1

Assigning the CMS block to the promotion page

Promotion pages by default do not have banners - they have to be added by assigning the CMS block into a promotion page.

In order for a CMS block to be visible on a promotion page, you have to set its title to "**promotion banner {ID}**" where ID is a promotion's page internal ID. For example, correctly assigned CMS blocks have names:

- promotion banner 1
- promotion banner 552
- promotion banner 1066

As with every CMS page and CMS block, these CMS blocks contain code for both desktop and mobile banners using the responsive web design.

It is possible to add a dedicated category navigation, that is generated on all available categories from added products. Category Navigation CMS block has to be named in the following format:

promotion category navigation {ID}

Where ID is a promotion page's internal ID. For example, correctly assigned CMS blocks have names:

- promotion category navigation 1
- promotion category navigation 552
- promotion category navigation 1066

You can find an example code used in promotion category navigation at the bottom of this article

info2.svg **Remember**

You can learn about the CMS blocks and their creation in the [CMS blocks](#) article.

Example promotion category navigation {ID} code:

```
<link href="https://cdnjs.cloudflare.com/ajax/libs/tiny-slider/2.9.3/tiny-slider
.css"
rel="stylesheet" />

<style type="text/css">.selected {

    color: #ffffff !important;

    background-color: #8b9620 !important;

}

.tns-outer [aria-controls],
```

```
.tns-outer [data-action] {  
  
  display: none;  
}
```

```
.onglet-slider a:hover {  
  
  color: #8c951e;  
  
  border-color: #8c951e;  
}
```

```
.onglet-slider a {  
  
  box-sizing: border-box;  
  
  text-transform: uppercase;  
  
  text-decoration: none;  
  
  color: #000000;  
  
  padding: 7px 10px;  
  
  border-radius: 25px;  
  
  margin: 5px;  
  
  text-align: center;  
  
  font-size: 12px;  
  
  line-height: 12px;  
  
  margin: 1px 10px 0 0;
```

```
box-shadow: 0 2px 6px 0 rgba(0, 0, 0, 0.2);

border: solid 1px #868686;

background-color: #ffffff;

justify-content: center;

align-items: center;

display: flex;

width: 100%;

font-weight: bold;

height: 40px;
}

@media (min-width: 780px) {

  .onglet-slider {

    width: 100% !important;

  }
}

.onglet-slider {

  display: flex;

  gap: 10px;

  margin: 10px 0px;

}
```

```
.onglet {  
  
    min-width: 100px;  
  
}  
  
</style>  
  
<script src="https://cdnjs.cloudflare.com/ajax/libs/tiny-slider/2.9.2/min/  
tiny-slider.js">  
</script><script>  
  
    var slider = tns({  
  
        container: '.onglet-slider',  
  
        navAsThumbnails: false,  
  
        "edgePadding": 15,  
  
        "swipeAngle": false,  
  
        slideBy: 1,  
  
        "mouseDrag": true,  
  
        autoplay: false,  
  
        gutter: 1,  
  
        nav: true,  
  
        items: 3,  
  
        center: false,  
  
        loop: false,  
  
        responsive: {  
  
            480:{  
  
                items: 5,  

```

```
        loop: true
    },
    640: {
        "edgePadding": 25,
        loop: true
    },
    780: {
        items: 7
    },
    1024: {
        items: 10
    }
}
});
```

```
</script>
```

Actions

Screenshot_2020-04-29_at_19.1 Getting started

From this article, you will learn about promotion module actions. Actions are promotions that are applied to the products after selected conditions in the action are met.

info2.svg **Remember**

You can learn about the promotion module in the [Promotion module introduction](#) article.

To open the Actions / list view, in the admin panel, navigate to:

Promotions > Actions / list.

In the Actions / list view, we can find two tabs: **Filters (1)** and **Actions / list (2)**



POWERPNT_4pvgkjGCmx.png

Visualization of the actions / list module in the administrator panel

Table of contents:

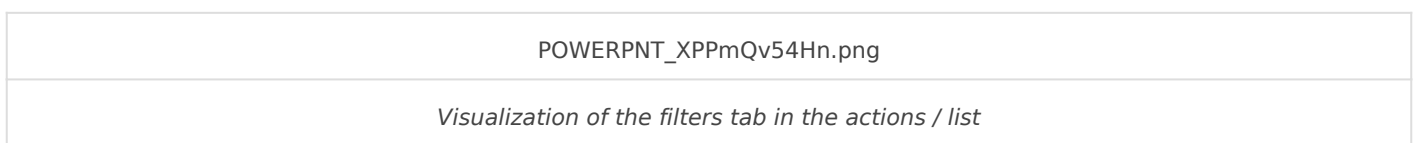
1. [Filters section.](#)
2. [Actions list section.](#)
3. [Adding a new action.](#)
4. [Action types.](#)
 1. [Discount from order value](#)
 2. [Discount the cheapest product](#)
 3. [Discount the most expensive product](#)
 4. [Gift](#)
 5. [Gift of choice](#)
 6. [Payed gift](#)
 7. [Free delivery](#)
 8. [Free payment](#)

9. [Show rule description](#)
 10. [Two products for one](#)
 11. [Enrollment set](#)
 12. [Product as gift](#)
 13. [Prima port discount](#)
 14. [Apply voucher discount](#)
 15. [Add or multiply loyalty points](#)
 16. [Set of products](#)
 17. [Product value multiplication](#)
 18. [Product value multiplication - most expensive](#)
 19. [Product value multiplication - cheapest](#)
-

Screenshot_2020-04-29_at_19.1

Filters section

The **Filters** section allows sorting of the actions in the promotion module, based on the available filters. Below you will find a description of each filter.



- **Is active?** - Defines the activity date of the visible list of actions.
- **Type** - Type of the action. More about rule types can be found in the [Action types](#) chapter.

After defining the above filters, select the **Apply Filters** button to get a list of actions according to the applied filters.

Screenshot_2020-04-29_at_19.1

Actions list section

The **Actions list** section allows organizing the filtered actions, as well as editing, duplicating and deleting the desired actions.

Below you will find a description of all possible actions you can perform in the actions list section.

POWERPNT_VDHzVUbfwS.png
<i>Visualization of the Actions / list tab in the Actions / list.</i>

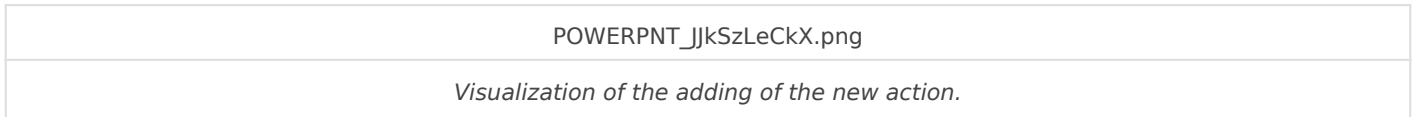
- Select the **Show X entries(1)** field to set the number of actions, displayed on one page (min 10 / max 100).
 - The names of the columns **(2)** that contain information about target actions. Each column is susceptible to sorting. To sort the records, click on the title of the column in question.
 - **ID** - Internal ID of the action.
 - **Type** - Type of the action. You can read more about types in the [Action types](#) chapter.
 - **Name** - Name of the action.
 - **Is active?** - Activity status of the action.
 - **Actions** - List of buttons with possible actions that can be done to the desired action:
 - **Edit** - Edit desired action. Editing has exactly the same actions as adding a new action, so the list of actions can be looked up at [Adding a new action](#) chapter.
 - **Duplicate** - Duplicates the actions with all its properties and content.
 - **View** - Allows to view the content of the action without the ability to edit any fields in it.
 - **Delete** - Deletes the action without the ability to restore it.
 - Select the **Add (3)** button to go to the adding the action. You can learn how to add an action in the [Adding a new action](#) chapter.
 - **Search box (4)** where you can search for the desired action by its name.
 - **Page buttons (5)**, which allows user to change the page of currently viewed actions.
-

Screenshot_2020-04-29_at_19.1

Adding a new action

Add button allows users to create a new promotion action, which can be configured and prepared for very specific needs.

Below, you will find a description of all possible actions you can perform.



- **Type** - Type of the action. You can see more about action types in the [Action types](#) chapter
- **Name** - Internal name of the promotion action.
- **Description** - Description of the promotion action. This field is visible to the customer if the action "**Show description**" is assigned to the action.
- **Parameters** - JSON code used for parametrization of the action.
- **Is active?** - Activity status of the action.

info2.svg **Remember**

You can learn about promotion conditions in the [Conditions](#) article.

info2.svg **Remember**

You can learn about promotion rules in the [Rules](#) article.

Screenshot_2020-04-29_at_19.1

Action types

The type of action defines what activity will apply to the products when the condition is met.

Discount from order value

Discounts entire order using percentage or amount.

Parameters:

- **base_prices** - promotion (default) / catalog - which prices should be affected by discount.
- **type**: percentage (default) / amount - the value of the discount is percentage or amount to remove from order value?
- **use_default_excluded_list**: TRUE (default) / FALSE - use global SKUs that have been excluded globally in **Promotions > Products excluded from promotion (global)**
- **include_paid_gifts**: FALSE (default) / TRUE - apply discount also to paid gifts
- **exclude_products**: array, f.x. ["12345","12346"] - list of SKUs that should be excluded from this discount. You can use "green_point_flag", "bestseller_flag", "limited_edition_flag", "set_flag" and "outlet_flag" as SKUs which will include all SKUs with selected flags in excluded list. F.x. ["green_point_flag", "set_flag", "01234"].
- **allowed_products**: array, f.x. ["12345","12346"] - list of SKUs which CAN be discounted, it disabled excluded products settings (only products from this list can be discounted)
- **required_quantity**: "all" (default) / any number - number of SKUs from list which has to be present in the order, "all" means all SKUs from list has to be present
- **multiplier_sku_list**: array, e.g. ["12345","12346"] - list of SKUs which quantities should be summed; result number will multiply discount value
- **multiple**: FALSE (default) / TRUE - if discount value will be multiplied

JSON example:

```
{"type":"amount","base_prices":"promotion","allowed_products":["J0322","J0323","J0324","J0325"],
```

Discount the cheapest product

Discounts only the cheapest products in the cart. In case there are two cheapest products in the cart with exactly the same price the promotion module chooses the first one on the list.

Parameters:

- **type**: percentage (default) / amount - the value of the discount is percentage or amount to remove from order value?
- **prices**: "promotion" (default) or "catalog" (use one of the values) - define which product price (catalog or promotion) is to be granted a discount
- **use_default_excluded_list**: TRUE (default) / FALSE - use global SKUs that have been excluded globally in **Promotions > Products excluded from promotion (global)**
- **exclude_products**: array, f.x. ["12345","12346"] - list of SKUs which should be excluded from this discount
- **allowed_products**: array, f.x. ["12345","12346"] - list of SKUs which CAN be discounted, it disabled excluded products settings (only products from this list can be discounted)
- **quantity_limit** (default 1): integer - quantity which should be discounted. It can be 3 of 5 pieces of the cheapest product or 3 cheapest products (every with 1pcs) depending on the cart.

JSON example:

```

{"type":"percentage", "prices":"promotion","exclude_products":["I1695","I1696","BB003","BB004"],

```

Discount the most expensive product

Discounts for the most expensive single or multiple products are currently added to the cart. In case there are two most expensive products in the cart with exactly the same price the promotion module chooses the first one on the list.

Parameters:

- **type**: percentage (default) / amount - the value of the discount is percentage or amount to remove from order value?
- **prices**: "promotion" (default) or "catalog" (use one of the values) - define which product price (catalog or promotion) is to be granted a discount
- **use_default_excluded_list**: TRUE (default) / FALSE - use global SKUs that have been excluded globally in **Promotions > Products excluded from promotion (global)**
- **exclude_products**: array, f.x. ["12345","12346"] - list of SKUs which should be excluded from this discount
- **allowed_products**: array, f.x. ["12345","12346"] - list of SKUs which CAN be discounted, it disabled excluded products settings (only products from this list can be discounted)
- **quantity_limit** (default 1): integer - quantity which should be discounted. It can be 3 of 5 pieces of the most expensive product or 3 most expensive products (every with 1pcs) depending on the cart.

JSON example:

```
{"type": "percentage", "prices": "catalog", "exclude_products": ["BB012", "BB011", "I1695", "I1696", "BE
```

Gift

Adds gift to the cart. Product that will be added to the cart must have the **GIFT** flag activated. If the product has at the same time activated **GIFT** and **GIFT OF CHOICE** flags product won't be visible.

This action doesn't have any parameters. The price of the gift is taken from the product price.

You can find more about product flags in the **[Under construction]** article

Gift of choice

Adds to the cart the list of the gifts, that customers can choose one from.

Those gifts must have **GIFT** and **GIFT OF CHOICE** flags. The price of the gift of choice is taken from the product price.

Parameters:

- **sku_list**: array, e.g. ["12345", "12346"] - list of gifts SKUs (they have to be gifts). The list may also contain SKU "EMPTYGIFT", which allows the configuration of variants: gift or discount.
- **min_cart_value_for_description**: float (default: 0.00) - minimum cart value to activate these gifts, used to inform the customer and to group gifts of choice to one set (box)
- **condition_description**: string (default: "") - condition description used instead of minimum cart value information
- **show_inactive**: 0|1 (default: 0) - show inactive gifts group in cart
- **multiplier_sku_list**: array, e.g. ["12345", "12346"] - list of SKUs which quantities should be summed; result number will multiply this list of choice
- **multiplier_divider**: integer (default: 1) - divider used in calculation of how many lists should be presented, result number is always rounded down
- **empty_gift_description**: if in **sku_list** there is a gift that is not available in the catalog anymore (i.e. got removed, wrong SKU number was typed) this description will be displayed instead of the product.

- **empty_gift_picture**: if in **sku_list** there is a gift that is not available in the catalog anymore (i.e. got removed, wrong SKU number was typed) this picture will be displayed instead of the product.
- **show_missing_amount_when_active_is_rule: RULE_ID** - the rule will be visible (without the option to select, with the missing amount to be achieved) when the RULE_ID rule is active. For example **show_missing_amount_when_active_is_rule: 6** will result in a GOC promotion to be grayed out when promotion rule 6 is active when the promotion itself is not active yet.
- **sku_list_stock_track: QUANTITY** - quantity of products from sku_list below which warning email about low stock should be sent (if service of informing about stock levels of promotional products is enabled). This functionality needs to be enabled and e-mail address has to be set in **System > System / Settings > Admin > Promotion stock warning**. Stock values are set in **System > System / Settings > E-commerce > Stocks**.
- **fallback_sku_list : array** - array with list of SKUs that will be used as fallback when product from sku_list is not available. Products from fallback list are not associated with specific products from sku_list, they will be used in order they are on the list and in quantity corresponding to quantity of products from sku_list, which stock level dropped to 0.

JSON example:

```
{"sku_list": ["F06881", "F94615", "F44972", "250PT", "NOGIFT"], "min_cart_value_for_description": "296"}
```

Payed gift

Shows in the cart product that can be optionally bought by the customer usually in the promotional price **(1)**. The price of the payed gift is taken from the product price.

Parameters:

- **sku_list**: array, e.g. ["12345", "12346"] - list of gifts SKUs
- **min_cart_value_for_description**: float (default: 0.00) - minimum cart value to activate these gifts, used to inform the customer
- **condition_description**: string (default: "") - condition description used instead of minimum cart value information

JSON example:

```
{"sku_list": ["0I0811"], "min_cart_value_for_description": "899.00"}
```

POWERPNT_CLyPZnzJw9.png

Visualization of the payed gift in the cart

Free delivery

Makes that every delivery option is free of charge. This action doesn't have any parameters.

Free payment

Makes that every payment option is free of charge. This action doesn't have any parameters.

Show rule description

Shows the description of the promotion from **rules / list** in the cart **(2)**.

POWERPNT_8Brg7Vb6Yr.png

Visualization of the show rule description action.

info2.svg **Remember**

You can learn about promotion rules in the [Rules](#) article.

Two products for one

Takes the product from the allowed_products parameter, and if it is in the cart, adds another same product, but for free

Parameters:

- **use_default_excluded_list:** TRUE (default) / FALSE - use global SKUs that have been excluded globally in **Promotions > Products excluded from promotion (global)**
- **exclude_products:** array, f.x. ["12345","12346"] - list of SKUs that should be excluded from this discount
- **allowed_products:** array, f.x. ["12345","12346"] - list of SKUs which CAN be discounted, it disabled excluded products settings (only products from this list can be discounted)
- **catalog_price:** TRUE / FALSE (default) - set catalog price for main product
- **limit:** 0 (default) - limit of actions per cart
- **cheapest:** TRUE / FALSE (default) - priority actions from the cheapest product

JSON example:

```
{"catalog_price":"TRUE","exclude_products":["green_point_flag","set_flag"],"limit":"50"}
```

Enrollment set

Exact same functionality as **Gift of Choice**, but with bigger product images. Adds a list of products that customers can choose one.

Parameters:

- **sku_list:** array, e.g. ["12345","12346"] - list of gifts SKUs (they have to be gifts)
- **must_to_buy:** TRUE (default) | FALSE - one product must be purchased
- **min_cart_value_for_description:** float (default: 0.00) - minimum cart value to activate these gifts, used to inform the customer and to group gifts of choice to one set (box)
- **condition_description:** string (default: "") - condition description used instead of minimum cart value information

JSON example:

```
{"sku_list": ["P26885","P27300","P27714","P28172"],"must_to_buy": "TRUE"}
```

Product as gift

Creates the button with text from the **“text”** parameter and puts it in the product catalog right next to the **“add to cart”** button. With this button, customers can buy selected products from the

“**sku_list**” parameter in the promotional price from the “**price**” parameter.

 **Warning**

Only one action of this type should exist in the altshop. Having more than one product as gift action active in the same altshop may result in unexpected behaviour.

Parameters:

- **sku_list**: array, e.g. ["12345","12346"] - list of product SKUs that can be chosen as a gift
- **limit**: integer (default: 1) - how many products can be chosen as a gift
- **price**: float (default: 0.00) - the price of gift added to cart
- **text**: string (default: "Choose as gift") - text on a button

JSON example:

```
{ "sku_list" : ["36454","87816","77108","37086","26091","20410","87476","27899","70215","66318",
```

Prima port discount

This type of action is no longer supported since the Prima port is no longer used.

Apply voucher discount

Applies the FIDEN2 Voucher to the product. This functionality only works when FIDEN2 functionality is turned on in **system / settings**. You can learn more about FIDEN2 in the [FIDEN2](#) article.

Parameters:

- **use_default_excluded_list**: TRUE (default) / FALSE - use global SKUs that have been excluded globally in **Promotions > Products excluded from promotion (global)**
- **include_payed_gifts**: FALSE (default) / TRUE - apply discount also to payed gifts
- **exclude_products**: array, f.x. ["12345","12346"] - list of SKUs that should be excluded from this discount. You can use "green_point_flag", "bestseller_flag", "limited_edition_flag", "set_flag" and "outlet_flag" as SKUs which will include all SKUs with selected flags in the excluded list. F.x. ["green_point_flag", "set_flag", "01234"].

- **allowed_products**: array, f.x. ["12345","12346"] - list of SKUs which CAN be discounted, it disabled excluded products settings (only products from this list can be discounted)

JSON example:

```
{"exclude_products":["85949","set_flag"]}
```

Add or multiply loyalty points

Replaces the amount of points per product set in FIDEN2 system / settings with the new value. You can learn more about FIDEN2 in the [FIDEN2](#) article.

Parameters:

- type: "amount" (default) / "percentage" - add points by the amount or multiply by the selected percentage

JSON example:

```
{"type":"percentage"}
```

Set of products

Allows to create sets of products. In those sets, it can be set that if for example, we have three products from the “**allowed product**” parameter the third product will be for free as a gift or for reduced price. It is possible to set “**set_size**” and “**discounted_amount**” to the same value, so every product in the set will be discounted to the “**discounted_price**” value.

Parameters:

- **prices**: "promotion" (default) or "catalog" (use one of values) - define which product price (catalog or promotion) is to be granted a discount
- **discounted_price** (default "0.01"): string - discounted product price
- **allow_same_sku**: FALSE (default) / TRUE - allow to create pair of products with the same SKU

- **sticker** (default none): string - cms block name
- **quantity_limit** (default 999): integer - max sets that can be created by the customer within the promotion in the cart.
- **exclude_products**: array, f.x. ["12345","12346"] - list of SKUs that should be excluded from this discount. You can use "green_point_flag", "bestseller_flag", "limited_edition_flag", "set_flag" and "outlet_flag" as SKUs which will include all SKUs with selected flags into the excluded list. F.x. ["green_point_flag", "set_flag", "01234"].
- **allowed_products**: array, f.x. ["12345","12346"] - list of SKUs which CAN be discounted, it disabled excluded products settings (only products from this list can be discounted)
- **include_payed_gifts**: FALSE (default) / TRUE - apply discount also to payed gifts
- **use_default_excluded_list**: TRUE (default) / FALSE - use global SKUs that have been excluded globally in **Promotions > Products excluded from promotion (global)**
- **set_size** (default 2): integer - The size of the set
- **discounted_amount** (default 1): integer - The amount of products that will be discounted when the set is created.
- **cmsblock** (default ""): optional title of cms block with the content of promotion description pop-up

JSON example:

```
{ "prices": "catalog", "discounted_price": "75", "allow_same_sku": "TRUE", "quantity_limit": "1", "allowe
```

Product value multiplication

Product value multiplication discounts the product according to a multiplication of its current catalog or promotional price i.e. if the multiplication is set to 10 and discount to 2 euro, the promotion will discount 2 euro for every 10 euro in the product, so a product that costs 50 euro will have 10 euro of discount.

Parameters:

- **multiplication_value**: value from which multiplication discount will be calculated
- **every_sku_separately**: TRUE/ FALSE (default) - value defines if the discount should be calculated separately for every SKU or summed up from all allowed SKUs
- **prices**: "promotion" (default) or "catalog" (use one of values) - define which product price (catalog or promotion) is to be granted a discount
- **use_default_excluded_list**: TRUE (default) / FALSE - use global SKUs that have been excluded globally in **Promotions > Products excluded from promotion (global)**
- **exclude_products**: array, f.x. ["12345","12346"] - list of SKUs that should be excluded from this discount. You can use "green_point_flag", "bestseller_flag", "limited_edition_flag", and "set_flag" as SKUs which will include all SKUs with selected

flags into the excluded list. F.x. ["green_point_flag", "set_flag", "01234"].

- **allowed_products**: array, f.x. ["12345","12346"] - list of SKUs which CAN be discounted, it disabled excluded products settings (only products from this list can be discounted)

JSON example:

```
{"multiplication_value":10,"prices":"catalog","allowed_products":["36454","87816","77108","3708
```

Product value multiplication - most expensive

Product value multiplication discounts the most expensive product according to the multiplication of its current catalog or promotional price i.e. if the multiplication is set to 10 and discount to 2 euro, the promotion will discount 2 euro for every 10 euro in the product, so a product that costs 50 euro will have 10 euro of discount.

Parameters:

- **multiplication_value**: value from which multiplication discount will be calculated
- **prices**: "promotion" (default) or "catalog" (use one of values) - define which product price (catalog or promotion) is to be granted a discount
- **use_default_excluded_list**: TRUE (default) / FALSE - use global SKUs that have been excluded globally in **Promotions > Products excluded from promotion (global)**
- **exclude_products**: array, f.x. ["12345","12346"] - list of SKUs that should be excluded from this discount. You can use "green_point_flag", "bestseller_flag", "limited_edition_flag", "set_flag" as SKU which will include all SKUs with selected flags into the excluded list. F.x. ["green_point_flag", "set_flag", "01234"].
allowed_products: array, f.x. ["12345","12346"] - list of SKU which CAN be discounted, it disabled excluded products settings (only products from this list can be discounted)

JSON example:

```
{"multiplication_value":10,"prices":"catalog","allowed_products":["36454","87816","77108","3708
```

Product value multiplication - cheapest

Product value multiplication discounts the cheapest product according to a multiplication of its current catalog or promotional price i.e. if the multiplication is set to 10 and discount to 2 euro, the promotion will discount 2 euro for every 10 euro in the product, so a product that costs 50 euro will have 10 euro of discount.

Parameters:

- **multiplication_value**: value from which multiplication discount will be calculated
- **prices**: "promotion" (default) or "catalog" (use one of values) - define which product price (catalog or promotion) is to be granted a discount
- **use_default_excluded_list**: TRUE (default) / FALSE - use global SKUs that have been excluded globally in **Promotions > Products excluded from promotion (global)**
- **exclude_products**: array, f.x. ["12345","12346"] - list of SKUs that should be excluded from this discount. You can use "green_point_flag", "bestseller_flag", "limited_edition_flag", "set_flag" as SKU which will include all SKUs with selected flags into the excluded list. F.x. ["green_point_flag", "set_flag", "01234"].
- **allowed_products**: array, f.x. ["12345","12346"] - list of SKUs which CAN be discounted, it disabled excluded products settings (only products from this list can be discounted)

JSON example:

```
{"multiplication_value":10,"prices":"catalog","allowed_products":["36454","87816","77108","3708"]}
```

Rules

Screenshot_2020-04-29_at_19.1 Getting started

From this article, you will learn about promotion module rules. Rules are whole promotions that contain in themselves conditions and actions and make them work together in selected altshops and date ranges. They are essential parts for a promotion to work.

info2.svg **Remember**

You can learn about the promotion module in the [Promotion module introduction](#) article.

To open the Rules / list view, in the admin panel, navigate to:

Promotions > Rules / list.

In the Rules / list view, we can find two tabs: **Filters (1)** and **Rules / list (2)**

POWERPNT_GgMEH2m3fj.png

Visualization of the rules / list module in the administrator panel

Table of contents:

1. [Filters section](#)
2. [Rules list section](#)
3. [Adding a new rule](#)

Screenshot_2020-04-29_at_19.1

Filters section

The **Filters** section allows sorting of the rules in the promotion module, based on the available filters. Below you will find a description of each filter.

POWERPNT_f6VCsxqMBY.png

Visualization of the filters tab in the rules / list

- **Is active?** - Defines the activity date of the visible list of rules. By default, "Yes" is selected.
- **Alt shop** - Defines to which alt shop, you want to limit the list of rules.

After defining the above filters, select the **Apply Filters** button to get a list of rules according to the applied filters.

Screenshot_2020-04-29_at_19.1

Rules list section

The **Rules list** section allows organizing the filtered rules, as well as editing, duplicating, and deleting the desired rules.

Below you will find a description of all possible actions you can perform in the rules list section.

POWERPNT_7WugyYXTJK.png

Visualization of the Rules / list tab in the Rules / list.

- Select the **Show X entries(1)** field to set the number of rules, displayed on one page (min 10 / max 100).
- The names of the columns **(2)** that contain information about target rules. Each column is susceptible to sorting. To sort the records, click on the title of the column in question.
 - **ID** - Internal ID of the rule.
 - **Name** - Name of the rule.
 - **Start / end date** - Range of dates where the rule is active.
 - **Is active?** - Activity status of the rule.
 - **Sort order** - Sort order of the rule. Sort order defines the order in which the rules will be applied in the customer cart, starting from 1.
 - **Actions** - List of buttons with possible actions that can be done to the desired rule:
 - **Edit** - Edit desired rule. Editing has exactly the same actions as adding a new rule, so the list of actions can be looked up in [Adding a new rule](#) chapter.
 - **Duplicate** - Duplicates the rules with all their properties and content. By default, duplicates are **inactive**.

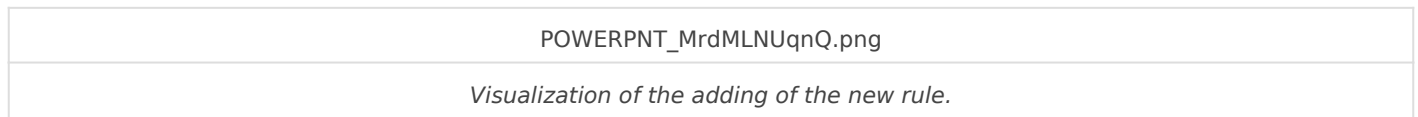
- **View** - Allows to view the content of the rule without the ability to edit any fields in it.
- **Delete** - Deletes the rule without the ability to restore it
- Select the **Deactivate expired rules button (3)** to set the **is active?** status on all the rules that have **start / end date** range outside the current date.
- Select the **Add list (4)** button to go to the adding the rule. You can learn how to add a rule in the [Adding a new rule](#) chapter.
- **Search box (5)** where you can search for desired rule by its name.
- **Page buttons (6)**, which allows user to change the page of currently viewed rules.

Screenshot_2020-04-29_at_19.1

Adding a new rule

Add button allows users to create a new promotion rule, which can be configured to very specific needs.

Below, you will find a description of all possible actions you can perform while creating a new rule.



- **Name** - Internal name of the promotion rule.
- **Rule code** - Value that is visible in the API. This value doesn't change anything, how the selected rule works on the website.
- **JOY club member discount percentage** - Value sent to JOY at the order line level. This value doesn't change anything, how the selected rule works on the website.
- **Description** - Description of the promotion rule. This field is visible to the customer if the action "**Show description**" is assigned to the rule.
- **Rule type** - Type of the rule. There are two types to choose from:
 - **Standard** - standard rule type that doesn't make any specific changes. This is the default type that should be used in most cases.
 - **ETD** - is used for backward analytics of ETD (End of ticket discount) discounts and how many of them were added. Selecting this type requires **JOY club member discount percentage** to be filled.
- **Start / end date** - Range of dates where the rule is active.
- **Is active?** - Activity status of the rule.

- **Sort order** - Sort order of the rule. Sort order defines the order in which the rules will be applied in the customer cart, starting from 1.
- **Altshops** - List of altshops where the rule will be active. You can assign the altshop by selecting them from the list on the left side, and pressing the arrow buttons. The double arrow button will move all the altshops
- **Conditions** - In the conditions field you can assign created conditions to the rule, assign operators to them, and the values. It is possible to change the order of the conditions using the arrows on the left side of the condition. Additionally, you are able to create a new condition from this window using the "**Create new condition**" button **(1)** and edit existing, selected conditions using the "**Edit**" button **(2)** next to the desired condition. At the bottom of the conditions list you will find brackets and logic connectors. They can be used to create more complicated promotions
 - **Brackets** - Brackets can be used with logic connectors to separate certain conditions, and run them together. For example, conditions can look like: **X or {Z and Y}**, where **X** condition will run the promotion **OR** combination of **Z AND Y**.
 - **Logic connectors** - Logic connectors allow you to create complicated promotions where multiple conditions must be fulfilled by using **AND** connector, or to select that the rule will be activated by multiple, independent conditions using the **OR** connector.
- **Actions** - In the actions field, you can select multiple actions from the list of available ones. Actions will run, whenever selected conditions are met. You are additionally able to create a new action from this window using the "**Create new action**" button **(3)** and edit existing, selected action using the "**Edit**" button **(4)** next to the desired action.

POWERPNT_EtpRitCDNZ.png

Visualization of the condition creating window

POWERPNT_ygnpa7kjpg.png

Visualization of the action editing window

info2.svg **Remember**

You can learn about promotion conditions in the [Conditions](#) article.

info2.svg **Remember**

You can learn about promotion actions in the [Actions](#) article.

In the example below, you can see a promotion with:

- Conditions that need to be fulfilled: Order value (catalog price) more or equal 50 euro **OR** { Discount code in basket = CODE **AND** Order items count more or equal 2 }.

- If the conditions are met, the customer will receive 50 euro discount on the most expensive product and 25% discount on the cheapest one.

POWERPNT_VdCR2KLMJs.png

Visualization of the example promotion rule configuration

Conditions

Screenshot_2020-04-29_at_19.1

Getting started

This article will teach you about conditions, their types, and how to configure them. Conditions are used to determine whether a customer is eligible to get a promotion.

Conditions can be found in the

Promotions > Conditions / list

At the **Conditions / list** window **Filters (1)** can be found which will help search for a specific type of conditions, and the list of **Conditions (2)** itself, which allows for searching conditions by name, editing, deleting, and adding new ones.

Screenshot_2022-09-14_154342.png

Visualization of the Promotions, Conditions / list

info2.svg **Remember**

Conditions and **Actions** are integral to **Promotion Rules**. All three can't be used individually. You can learn more about actions and rules in the [Actions](#) and [Rules](#) articles

Table of contents:

1. [Conditions / list](#)
 1. [Filters](#)
 2. [Conditions](#)
2. [Creating new condition](#)
3. [Condition types](#)
 1. [Discount code in cart](#)

2. [Discount code is empty in cart](#)
 3. [Order value](#)
 4. [Check if SKU\(s\) exist](#)
 5. [Active alt shop](#)
 6. [Sum of quantities from SKU list](#)
 7. [Prima port ID in cart](#)
 8. [Count of orders](#)
 9. [Days from last order](#)
 10. [Voucher in cart](#)
 11. [Empty gift of choice in the cart](#)
 12. [Delivery method](#)
 13. [Payment method](#)
 14. [Customer title](#)
 15. [Regular account](#)
 16. [Order items count](#)
 17. [Promotion rules active in cart](#)
-

Screenshot_2020-04-29_at_19.1

Conditions / list

Filters

Filters are the first tab in **Conditions / list**. It is used to narrow the results in the **Conditions** list.

(1) Is active? - Shows only active or inactive conditions.

(2) Type - Shows only active or inactive conditions.

(3) Two buttons:

- **Apply filters** - Applies chosen filters to **Conditions**
- **Clear filters** - Clears chosen filters and refreshes **Conditions**

Conditions

The conditions list is the main tab in this module. Here new conditions can be added **(1)**, below there is a search field **(2)** that will filter out conditions names to typed letters. The table list itself **(3)** contains several columns, these are:

- **ID** - The number given after the creation of the condition, it cannot be changed. Used only for database listing.
- **Type** - Type of condition that determines its behavior (more about types in this section [\[LINK\]](#))
- **Name** - Name of the condition, used as the main identification. Its name is used for rules and in the search bar.
- **Is active?** - Mark that shows whether a condition is active or inactive
- **Actions** - Four buttons:
 - **Edit** - Edit condition, its name, description, whether it's active, and parameters.
 - **Duplicate** - Duplicates condition with exact same parameters, the name will be edited by gaining (copy) as a suffix and changing its activation status to inactive.
 - **Show** - Shows details of the condition without permission to edit.
 - **Delete** - Erase condition, it won't be recoverable.

Screenshot_2020-04-29_at_19.1

Creating new condition

The condition has three fixed options that will stay the same **(1)**, these are:

- **Name** - Name of the condition, used as the main identification. Its name is used for rules and in the search bar.

- **Description** - Description serves as an additional source of information about the condition and its mechanics
- **Is active?** - ON/OFF - switch to turn the condition active or inactive.

There are also two **(2)** configurable options that determine its behavior, these are:

- **Type** - Determines the main target of verification. Types are:
 - **Discount code in cart (link do każdego, scrollujący do danego typu)**
 - **Discount code is empty in cart**
 - **Order value**
 - **Check if SKU(s) exist**
 - **Active alt shop**
 - **Sum of quantities from SKU list**
 - **Prima port ID in cart**
 - **Count of orders**
 - **Days from last order**
 - **Voucher in the cart**
 - **Empty gift of choice in the cart**
 - **Delivery method**
 - **Payment method**
 - **Customer title**
 - **Regular account**
 - **Order items count**
 - **Promotion rules active in cart**
- **Parameters** - JSON code used for parametrization of the condition.

Screenshot_2022-09-15_152600.png

Visualization of Filters, Conditions / list

info2.svg **Remember**

JSON is a light-weighted format to exchange data. More about this format you can learn from [here](#).

Screenshot_2020-04-29_at_19.1

Condition types

The type of condition defines what it's supposed to look for when calculating cart items and values. Here are described all types of conditions and their parameters

Discount code in cart

This condition checks if there is a discount code in the cart (more about discount codes can be found in the [Discount codes](#) article).

Parameters:

- **None**

Logic operators:

- =
- !=

Logic values:

- **Discount code name**

info2.svg **Remember**

Since this condition type is so general and lacks any parameters. There is usually one of this type in the environment.

Discount code is empty in cart

This condition checks if there isn't a discount code in the cart.

Parameters:

- **Discount code name** - it is not used in the condition itself since adding this type of condition to the rule will require naming it.

Logic operators:

- =
- !=

Logic values:

- **TRUE**
- **FALSE**

info2.svg **Remember**

Since this condition type is so general and lacks any parameters. There is usually one of this type in the environment.

Order value

This condition checks the current cart value.

Parameters:

- **prices:**
 - **promotion** - used by default, counts the value of products by promotion prices
 - **catalog** - counts values of products by their standard prices
- **consider_discounts:**
 - **FALSE** - used by default, counts the value before all discounts
 - **TRUE** - calculate order value after all discounts
- **use_default_excluded_list:**
 - **TRUE** - used by default, uses global excluded SKU list
 - **FALSE** - excluded SKUs will also be counted toward the cart value
- **include_payed_gifts:**
 - **TRUE** - consider the value of payed gifts
 - **FALSE** - used by default, do not count payed gifts
- **include_opensets:**
 - **TRUE** - opensets count into the cart value
 - **FALSE** - used by default, do not count value of opensets
- **exclude_products:**
 - **Array** - SKU numbers, these will be excluded from the order value. This parameter is above and will exclude products regardless of **use_default_excluded_list**

parameter.

- **exclude_fid_discounted_products:**
 - **TRUE** - Products discounted by the fidelity program will not be counted.
 - **FALSE** - used by default, do not count value of products discounted by fidelity points

Logic operators:

- =
- !=
- >
- <
- >=
- <=

Logic values:

- **Number** - cart value

Example:

- {"prices":"catalog","consider_discounts":"TRUE","exclude_products":["12345","54321"]}

Check if SKU(s) exist

This condition checks if the product SKUs from the selected array are present in the cart.

Parameters:

- **SKU:**
 - **Array** - SKU numbers, without this parameter the condition won't work.
- **required_quantity:**
 - **all** - All SKUs from the **SKU** parameter have to be present in the cart
 - **number** - Number of SKUs from the SKU parameter that has to be present.
- **consider_quantity:**
 - **TRUE** - The number of pieces counts towards the **SKU** parameter (for example if the **SKU** parameter has 3 SKU's in the list then one of these SKU's added 3 times to the cart will return **TRUE** value in the promotion rule)
 - **FALSE** - Used by default, doesn't count the number of pieces

Logic operators:

- =
- !=

Logic values:

- **TRUE**
- **FALSE**

Example:

- {"SKU":["12345","54321"],"required_quantity":"5"}

Active alt shop

This condition checks the altshop. This type is deprecated and unused. Promotion rules already are able to filter altshops.

Sum of quantities from SKU list

This type sums up all pieces from the SKU list set in the parameters.

Parameters:

- **SKU:**
 - **Array** - SKU numbers, without this parameter the condition won't work.

Logic operators:

- =
- !=
- >
- <
- >=
- <=

Logic values:

- **Number** - Quantity of products

Example:

- {"SKU":["54321","12345"]}

Prima port ID in cart

This type of condition is no longer supported since Prima port is no longer used.

Count of orders

This type count the number of orders in the customer account.

Parameters:

- **orders:**
 - **all** - All orders from the customer account will be counted (still regulated by the rest of the parameters)
 - **all_mc** - All orders are counted but since the date of gaining a title. You can learn more about customer titles in the [Campaigns and titles](#) article.
 - **current_title** - Orders are counted since the date of gaining the current customer title.
 - **active_rule** - Orders are counted only if they used certain promotion rules (list from parameter **rule id**)
 - **campaigns:**
 - **Array** - List of campaigns short names (more about campaigns in this article **[UNDER CONSTRUCTION]**)
 - **date_from:**
 - **date** - The format for the date is YYYY-MM-DD (for example 2018-12-31). Orders will be counted from this date onwards.
 - **date_to:**
 - **date** - The format for the date is YYYY-MM-DD (for example 2018-12-31). Counted will be orders placed only before this date.
 - **titles:**
 - **Array** - List of title short names. Orders will be counted only if they were placed while the customer had these titles. Only used when parameter **"orders":"current_title"**
 - **rule_id:**
 - **Array** - List of **Rules** id's. This parameter only works with **"orders":"active_rule"**
 - **only_paid:**
 - **TRUE** - Only COD or fully paid orders are counted.
 - **FALSE** - Orders with pending payment are also counted
 - **altshop_id:**
 - id of altshop only from which orders are considered. The parameter value can only be one altshop/one ID. You can't use an array of altshop ID's.
- Syntax example:**
- {"altshop_id":"122"}

Logic operators:

- =
- !=
- >

- <
- >=
- <=

Logic values:

- **Number** - Orders

Example:

- {"orders":"all_mc","campaigns":["Campaign1","Campaign2"],"date_from":"2023-01-12","date_to":"2023-02-22"}

Days from last order

This condition count passed days from the last order.

Parameters:

- **orders:**
 - **all** - All orders from the customer account will be counted (still regulated by the rest of the parameters)
 - **all_mc** - All orders are counted but since the date of gaining a title. You can learn more about customer titles in the [Campaigns and titles](#) article.
 - **current_title** - Orders are counted since the date of gaining the current customer title.
 - **active_rule** - Orders are counted only if they used certain promotion rules (list from parameter **rule id**)
- **campaigns:**
 - **Array** - List of campaigns short names (more about campaigns in this article **[UNDER CONSTRUCTION]**)
- **date_from:**
 - **date** - The format for the date is YYYY-MM-DD (for example 2018-12-31). Orders will be counted from this date onwards.
- **date_to:**
 - **date** - The format for the date is YYYY-MM-DD (for example 2018-12-31). Counted will be orders placed only before this date.

Logic operators:

- =
- !=
- >
- <
- >=
- <=

Logic values:

- **Number** - Orders

Example:

- {"orders":"all_mc","campaigns":["Campaign1","Campaign2"],"date_from":"2023-01-12","date_to":"2023-02-22"}

caution.svg **Warning**

If there are no order-matching rules defined by parameters, the condition will return 999 days. We strongly recommend pairing this condition with "Count of closed orders > 0"

Voucher in the cart

This condition verifies whether there is applied promotion code in the cart.

Parameters:

- **type:**
 - **1** - Fiden2 code, this type is set as default.
 - **2** - Promotion code from **Promotions > Discount codes / list**. You can learn more about discount codes in the [Discount codes](#) article.

Logic operators:

- =
- !=

Logic values:

- **TRUE**
- **FALSE**

Example:

- {"type":"2"}

Empty gift of choice in cart

This condition will check if the customer chooses a certain, empty gift from the **Gift of choice** action type. You can learn more about actions in the [Actions](#) article.

Parameters:

- **rule_id:**
 - **1** - Fiden2 code, this type is set as default.
 - **2** - Promotion code from **Promotions > Discount codes / list** (More about codes in this article[**UNDER CONSTRUCTION**])

Logic operators:

- =
- !=

Logic values:

- **TRUE**
- **FALSE**

Example:

- {"rule_id":"2"}

Delivery method

This type checks chosen delivery method.

Parameters:

- **name:**
 - **Array** - List of delivery method names.

Logic operators:

- =
- !=

Logic values:

- **TRUE**
- **FALSE**

Example:

- {"name":["Delivery1","Delivery2"]}

Payment method

This type checks chosen payment method.

Parameters:

- **name:**
 - **Array** - List of delivery method names.

Logic operators:

- =
- !=

Logic values:

- **TRUE**
- **FALSE**

Example:

- {"name":["Payment1","Payment2"]}

Customer title

This type checks current customer title.

Parameters:

- **include_title:**
 - **Array** - List of **shortcut title names** that will count into the **TRUE** value during calculation.
- **exclude_title:**
 - **Array** - List of **shortcut title names** that will count into the **FALSE** value during calculation.

Logic operators:

- =
- !=
-

Logic values:

- **TRUE**
- **FALSE**

Example:

- {"include_title":["T0","TS"]}

Regular account

This type checks if a customer is logged in to a registered account (not one-time).

Parameters:

- **None**

Logic operators:

- =
- !=

Logic values:

- **TRUE**
- **FALSE**

Orders item count

This type checks the number of items in the cart.

Parameters:

- **consider_quantity:**
 - **TRUE** - Set by default, every piece in each row will be counted.
 - **FALSE** - Only unique SKUs will be counted.
- **consider_gifts:**
 - **TRUE** - Gifts will be counted to this condition.
 - **FALSE** - Set by default, gifts won't be counted to this condition.
- **consider_sets:**
 - **TRUE** - All sets will be counted.
 - **FALSE** - Set by default, sets won't be counted.
- **consider_outlet:**
 - **TRUE** - Discounted products with the flag outlet will count toward this condition.
 - **FALSE** - Set by default, products with the flag outlet won't be counted.
- **consider_greenpoint:**
 - **TRUE** - Products with the flag greenpoint will count toward this condition.
 - **FALSE** - Set by default, products with the flag greenpoint won't be counted.

Logic operators:

- =
- !=
- >
- <
- >=

- <=

Logic values:

- **Number** - Products in a cart

Example:

- {"consider_quantity":"FALSE","consider_greenpoint":"TRUE"}

Promotion rules active in cart

This type checks if a set of rules is active in a cart. A rule is considered an active one when its conditions are met. A rule with no conditions is always considered as active. It is important to remember to set a sort order in a rule with this condition to a high value, so it is read last by the promotion module.

Parameters:

- **rule_ids:**
 - **Array**- list of rules' IDs that are checked for activity. Set by default, every piece in each row will be counted.
- **mode (Default "all"):**
 - **"all"** means *all of the rules must be active*,
 - **"any"** means *any of the rules must be active*.

Logic operators:

- =
- !=

Logic values:

- **TRUE**
- **FALSE**

Examples:

- **Rule 123 is active:**
 - {"rule_ids": [123]}
- **All the rules 123 and 321 are active:**
 - {"rule_ids": [123, 321]}
- **Any of the rules 123 and 321 is active:**
 - {"rule_ids": [123, 321], "mode": "any"}

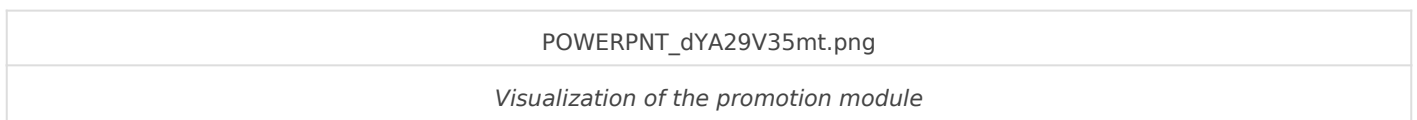
Promotion module introduction

Screenshot_2020-04-29_at_19.1 Getting started

In this article, you will learn about a promotion module, and what can you do by using that module.

Promotion module is a complex group of categories that together create a base for all the promotions on the website. By using the promotion module, you can create and configure your own promotions, or create a ready promotion pages and opensets for customers to select from.

Promotion module consists of many parts, described below:



- Promotion logic **(1)**, which is a main, and most important part of promotion module, which consists of conditions, actions, and rules that together create a promotion that is usable on the website **(2)**.
- Discount codes
- Promotion pages
- Opensets
- Excluded products tab
- Push messages
- FIDEN2 vouchers

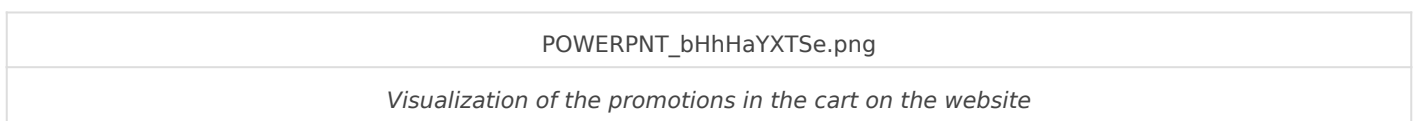


Table of contents:

1. [Promotion logic](#)
 1. [Excluded products](#)
2. [Discount codes](#)
3. [Promotion pages](#)
4. [Opensets](#)
5. [FIDEN2 vouchers](#)

Screenshot_2020-04-29_at_19.1

Promotion logic

Promotion logic allow creating custom promotions that will be applied according to the customer's cart value on the website. As an example of the promotion rules, we can distinguish:

- 20% catalog price discount on all products in the cart when customer's cart value is greater than 50 euro.
- Two for one promotion on selected products
- Buy 30 euro of selected products and get a gift product for free
- Select a product for free from a list when cart value is over 20 euro.
- When a discount code is applied in the cart, the most expensive and cheapest products are discounted by 50% of the promotional product value.

Promotion logic consist of three parts:

1. **A condition**
2. **An action**
3. **A rule**

Condition is a backbone of the promotion, as it dictates what conditions need to be fulfilled in order for the promotion to work. For example, in this section the "Order has to be greater than 50 euro" condition will be located.

An **action** applies the selected promotional action to the customer's cart, when the selected condition is met. In this section, the "discount 20% off catalog price of all the products in the cart" action will be located.

Rules are whole promotions that contain in themselves conditions and actions and make them work together in a selected altshops and date ranges.

You can find articles about these three parts in the following articles:

- [Conditions](#)
- [Actions](#)

- [Rules](#)

Excluded products

Products excluded from promotion (global) is a tab where users are able to exclude the list of SKU's globally, so these products won't be affected by any created promotions until they are excluded in this list.

POWERPNT_PUnATmux5q.png
<i>Visualization of the products excluded from promotion (global) tab.</i>

Screenshot_2020-04-29_at_19.1

Discount codes

Discount codes are codes that are used by the customers in the cart on the website in order to get benefits from the configured promotional rules. We distinguish two types of discount codes:

1. A main code
2. A personal code

Main code is a custom-made code created by the administrator with unlimited amount of uses (unless specified otherwise in the promotional rule).

Personal code is a code assigned inside the main code with one time use. When the main code has assigned at least one personal code, the main code itself doesn't work, and only the personal codes can be distributed among customers for it to work.

You can find more about discount codes in the [Discount codes](#) article

Screenshot_2020-04-29_at_19.1

Promotion pages

Promotion pages are easy to make and maintain pages with a list of products, where customers are able to select the desired products and add them separately to the cart on the website.

Promotion pages are usually created in order to hold categories of products that have discounts assigned in the [Products / list - browsing and managing products in altshop module](#).

POWERPNT_pCJSLs4a6S.png

Visualization of the promotion page

You can learn more about promotion pages in the [Promotion pages](#) article.

Screenshot_2020-04-29_at_19.1

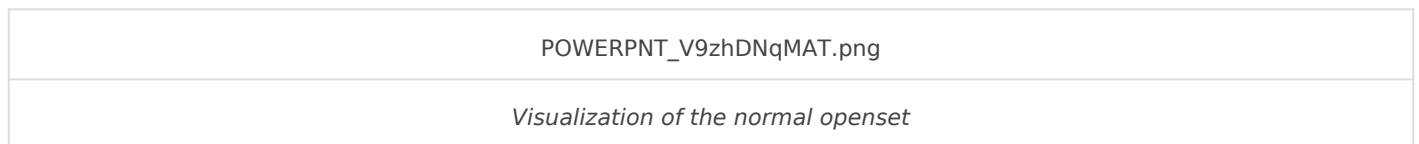
Opensets

Opensets are a set of products that are bundled into a single package.

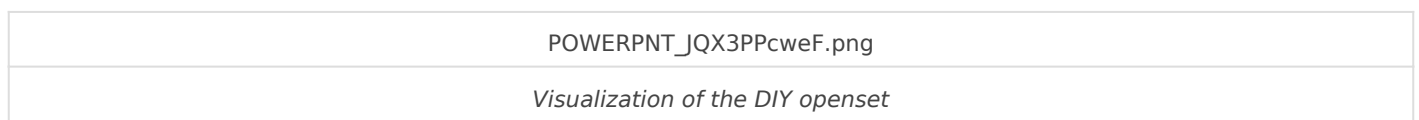
We distinguish three types of opensets:

1. Normal openset
2. DIY openset
3. DIY catalog price openset

Normal openset is a webpage where all the products are visible and selectable. The customer is able to select a configured amount of products and then bundle them together into an openset and added to the cart.



DIY (Do it yourself) openset and **DIY catalog price openset** are webpages that use the JSON of products and categories and shows them this way on the openset page. Customer is able to select a product from a category and is being moved automatically to another one. Once customer select at least one product in all available categories, the products will get bundled into an openset and added to the cart.



You can find more about opensets in the [Opensets](#) article.

Screenshot_2020-04-29_at_19.1

FIDEN2 vouchers

FIDEN2 vouchers are vouchers that use FIDEN2 points and assign a configured discount to the selected product and remove the configured amount of points from the customer's account.

Customer is able to earn FIDEN2 points by buying products on the website, creating reviews or inviting its friends to create an account on the website and creating a new order on it.



POWERPNT_QJrMu9Bkj3.png

Visualization of the adding of the discount through FIDEN2 functionality

You can learn more about FIDEN2 in the [FIDEN2](#) article.

Screenshot_2020-04-29_at_19.1

Push messages

Push messages functionality allows creating and send push messages to the customers. This functionality is no longer used and maintained as for YR5 version of the web application

Opensets

Screenshot_2020-04-29_at_19.1 Getting started

This article describes openset promotion functionality, which allows users to create a web page with set of products, where the customer is able to select desired products from the list, and add them to the cart as a set.

We distinguish three types of opensets - “**Normal**”, “**DIY**” and “**DIY catalog price**”, which are described in the following chapters.

POWERPNT_weGuVk7X9h.png
<i>Visualization of the DIY openset</i>

Table of contents

1. [Creating an Openset](#)
 1. [Creating a normal openset \(tab “General”\)](#)
 2. [Creating a normal openset \(tab “CMS blocks”\)](#)
 3. [Creating a DIY or DIY catalog prices openset](#)
 1. [JSON construction](#)

Screenshot_2020-04-29_at_19.1 Creating an Openset

In order to create an Openset, in the administration panel navigate to the **Admin panel > Promotions > Opensets / list** section and click **Add** button **(1)**, that will redirect you to the Openset creation page.

POWERPNT_jjmTvPbvN.png

Visualization of the opensets / list tab

Creating a normal openset (tab “General”)

Below you will find a description of all parameters needed to create an Openset.

POWERPNT_q8fYm8Qjvh.png

Visualization of the normal openset creation.

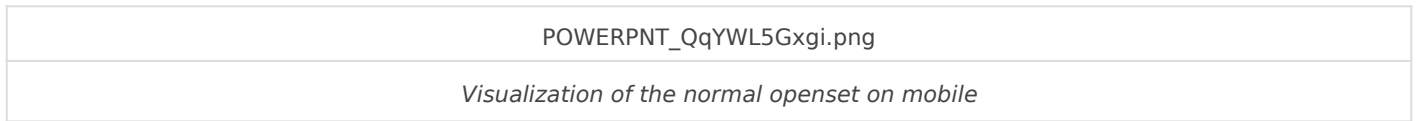
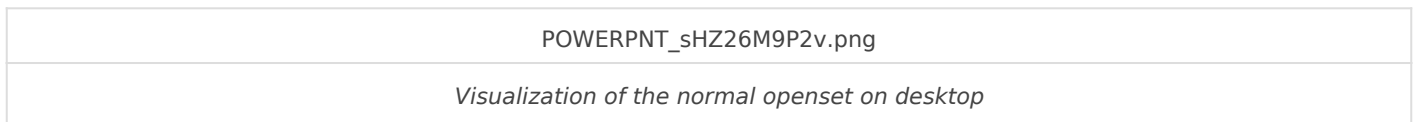
- **SKU** - automatically generated unique openset product code.
- **Name** - displayed name of the openset. The value is displayed on the e-commerce platform for the customer.
- **Friendly URL** - URL address where openset is located.
- **Type** - type of openset.
- **Price** - Target price for the created set. The superior prices of products placed in the openset (**Products SKU list** field) are not taken into account.
- **Products number** - number of products you can choose from the available list (**Products SKU list** field). After selecting 3 products, you will receive a popup with the possibility of adding the chosen set to the cart.
- **Allow to choose single product quantity** - A switch that enables customers to select more units than 1 of the same product, instead of the need to select different products.
- **Products SKU list** - list of products (SKU product code). Must be separated by comma.
- **Cart minimum value** - archive field, currently not used.
- **Start date** - date from which the openset will be visible on the e-commerce platform.
- **End date** - date from which openset will not be visible on the e-commerce platform.
- **Is active** - a switch that determines whether the openset is active or not on the e-commerce platform.

After filling in all the above parameters, select **Save** button to save the changes or **Cancel** button to cancel adding an Openset.

info2.svg **Remember**

In order for the element to be fully activated and visible on the e-commerce platform, remember to activate the openset in the Altshops module.

Below you'll find the visualization of normal openset on the e-shop website.



Creating a normal openset (tab “CMS blocks”)

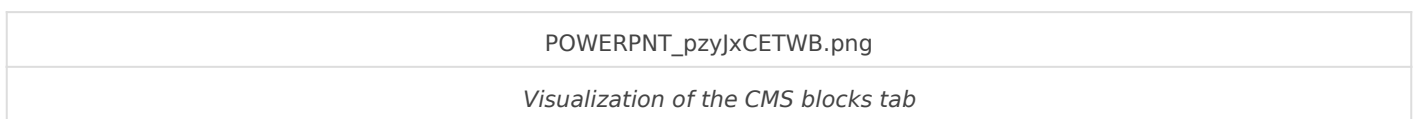
CMS Blocks tab allows us to see header CMS blocks (automatically created by the application) that will be used by the openset.

CMS block is required to have openset ID in its name, for example:

openset / header / 47 (where 47 is openset ID)

We can press “**add**” button, which will create a new CMS block for us with the appropriate block name.

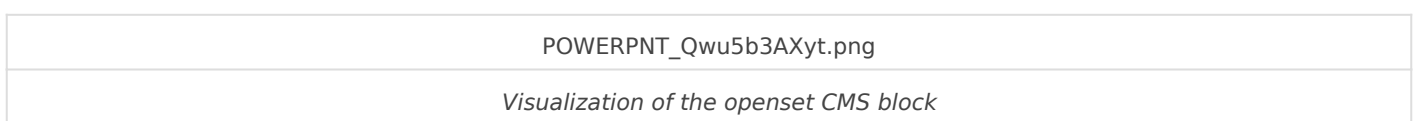
Search button will search the CMS blocks in order to find blocks that meet the required name (openset / header / ID). Mobile openset header is no longer used as the website is in a responsive design, so desktop and mobile headers should be located in the **openset / header / ID** CMS block



info2.svg **Remember**

The header of the openset cannot be wider than 1248px.

Section CMS block is the same for Normal as well as DIY opensets. And it looks as in the example displayed below **(1)**.



Creating a DIY or DIY catalog prices openset

The process of creating a DIY opener is identical to a [Normal type openset](#). The only differences between normal and DIY sets are:

- **Type** field, where we have to indicate **DIY** or **DIY catalog prices** openset.
- **Product SKU list** construction where we have a JSON record.
- Product price value indicated in the JSON is superior value in relation to **Price**. In case of **DIY catalog prices** openset the price value takes the catalog prices of the product instead of promotion prices.
- **Products number** - means the number of steps of which openset DIY is composed of.

POWERPNT_bQ3pEBGkWm.png

Visualization of the DIY openset in the admin panel

JSON construction

Below you can find an example JSON that can be used in the openset

```
[{"name": "Krem do rąk", "products": [{"sku": "27300", "price": "8.60"}, {"sku": "27714", "price": "8.60"}, {"sku": "28172", "price": "8.60"}, {"sku": "26885", "price": "8.60"}], {"name": "Mydło", "products": [{"sku": "05764", "price": "7.15"}, {"sku": "44059", "price": "7.15"}, {"sku": "42871", "price": "7.15"}, {"sku": "43332", "price": "7.15"}], {"name": "Balsam do ust", "products": [{"sku": "00174", "price": "7.15"}, {"sku": "00392", "price": "7.15"}, {"sku": "00599", "price": "7.15"}, {"sku": "01030", "price": "7.15"}]}
```

- **Name** parameter sets the name of the category. **(1)**
- **Products** array sets the product SKU and product price **(2)** in the set name .

The number of categories **(3)** is taken from the **Products number** field of the openset.

Below you'll find the visualization of DIY openset on e-shop website.

POWERPNT_5wzPVB4huF.png
<i>Visualization of the DIY openset</i>