



pCon.cloud  
pCon.ui – **Documentation**

# pCon.ui Documentation

## Content

1	Integration.....	3
1.1	iframe.....	3
1.2	iframe Resizer.....	3
2	pCon.ui for Websites - Lead.....	4
2.1	Integration in Websites (postMessage / Contact Button).....	4
2.2	postMessage.....	5
2.3	postMessage elements.....	6
2.4	additional postMessage elements.....	7
2.5	postMessage in Case of PECs.....	8
3	pCon.ui for Shops.....	9
3.1	Integration in Shop System (OCI Punchout).....	9
3.2	OCI Response.....	9
3.2.1	OCI 5.0 to OFML Mapping.....	11
3.2.2	OCI Example.....	13
3.2.3	OCI Example with Child Article.....	14
3.3	Reconfiguration.....	15
3.4	Price Validation.....	16
3.5	Checklist before Going Live.....	17
4	Individualization by URL Parameters.....	18
4.1	Basic Parameters for showing up an article.....	19
4.2	Parameters for changing behavior of 3D view.....	20
4.3	Parameters for changing Article Property view.....	20
4.4	other parameters.....	21
4.5	Helper Tool: Admin Panel.....	22
5	User Interaction Messages / User Tracking / Error Handling.....	24
5.1	General User Tracking.....	24
5.2	Error Handling.....	25
6	Price Update Webservice API.....	26
6.1	Concept.....	26
6.2	POST Request.....	27
6.2.1	Example incl. French ECO contribution.....	29
6.2.2	Example with child article:.....	29
6.3	Price Calculation.....	30
6.4	Response.....	31
6.4.1	Example without discount.....	32
6.4.2	Example with child articles.....	32
6.4.3	Example with visible discount.....	33
6.4.4	Example with visible discount (only discount percentage) without showing original price.....	34

- 7 PropertyService API..... 35
  - 7.1 Change available Properties..... 35
    - 7.1.1 POST Request..... 35
    - 7.1.2 Response..... 37
  - 7.2 Change available property values..... 38
    - 7.2.1 Post Request..... 38
    - 7.2.2 Response..... 38
- 8 Access Product Information Management (PIM)..... 39
  - 8.1 postMessage Response..... 39
  - 8.2 Supported Mediatypes..... 40
- 9 Adapt Variance of pCon.ui: Article Drawer and Property Editor..... 41
  - 9.1 Modular Article View..... 41
  - 9.2 Compact Property Editor View..... 41
  - 9.3 Combine the views..... 42
- 10 Adapt appearance of pCon.ui: MUI theme and CSS..... 43
  - 10.1 Color scheme..... 43
  - 10.2 Style Components..... 44
  - 10.3 Examples..... 45
    - 10.3.1 Change font..... 45
    - 10.3.2 Adapt cart button and hide number field in front of cart button..... 45
    - 10.3.3 Adapt request button and hide its icon..... 45
    - 10.3.4 Adapt price field..... 46
    - 10.3.5 Adapt original price and discount field..... 46
  - 10.4 Style Components for Modular View..... 47
- 11 Modify element labels..... 48
  - 11.1 overview changeable elements..... 48
  - 11.2 how to change label - prerequisites..... 48
  - 11.3 supported languages..... 49
  - 11.4 default label names and json file structure..... 49
- 12 Contact / Support..... 51
- 13 Notes on Data Protection (Cookies)..... 52
- 14 Footnotes..... 53

# 1 Integration

## 1.1 iframe

The individualized link, generated with the admin panel or manually, has to be integrated into the website as `iframe`<sup>3</sup>). We suggest to use an overlay since the potential design break is less noticeable to the user. Please note, that for the download of PDF reports or CAD models, the communication with our servers under a different domain needs to be allowed.

Every communication between pCon.ui and the `iframe` parent is a `postMessage`. Those messages can be checked with a little helper tool available via <https://ui.pcon-solutions.com/iframe.html>.

You can also attach all URL parameters directly to this tools URL, e.g. [https://ui.pcon-solutions.com/iframe.html?GATEKEEPER\\_ID=wcs\\_shop&sh=true](https://ui.pcon-solutions.com/iframe.html?GATEKEEPER_ID=wcs_shop&sh=true).

## 1.2 iframe Resizer

To adjust height of the `iframe` according to its width automatically, an `iframe resizer`<sup>1</sup> can be activated for pCon.ui. For large screens the height is set to 60% of the width and for small/mobile screens, 166% of the width are used. Please let us know, if you want to use this solution, so we can activate it for your pCon.ui instance. With this page you can test how your configuration affects the `iframe`. The test page also serves as an example for the correct usage of the `iframe resizer`.

For small screens, we also offer to set the height automatically adjusted to the content instead of using the 166% height. That means, there is no scrollbar inside the `iframe` even for long property lists. Please [contact us](#) if you want to use this option.

To integrate the `iframe resizer` in your website you have to add the following code to your website:

```
<iframe src="https://ui.pcon-solutions.com/" id="pcon-ui-iframe"></iframe>
<script type="text/javascript"
      src="https://ui.pcon-solutions.com/static/js/iframeResizer.min.js">
</script>
<script>
  iFrameResize({log:true}, '#pcon-ui-iframe');
</script>
```

---

<sup>1</sup> <https://github.com/davidjbradshaw/iframe-resizer#readme>

## 2 pCon.ui for Websites - Lead

### 2.1 Integration in Websites (postMessage / Contact Button)

The following describes the integration of pCon.ui into websites, which is also called the Lead scenario. In those scenarios it can be necessary to send the current configuration to the manufacturer in order to request further information. Often there exists already a contact form on the manufacturer's website. In the following section it is described how to connect pCon.ui with this contact form.

In the initial setup, i.e. in the configuration of your pCon.ui, you can choose to show or hide the "Send request"-button. Please note, that an activated "Send request"-button is only visible when the pCon.ui link is opened in an iframe (see [this section](#)).

#### Concept:



When the user presses the "Send request" button the pCon.ui sends a postMessage to the parent website. This postMessage includes a JSON<sup>2</sup> object with the article information. The parent website now evaluates the message, opens the contact form and prefills some of the form fields.

---

<sup>2</sup> see ext. wiki link providing [json syntax](#)

## 2.2 postMessage

Right after a click on the “Send request”-button there is a pre-inform postMessage<sup>3</sup>. This can be used to show, e.g., a contact form while pCon.ui collects the data for the contact-request postMessage. The JSON object in the pre-inform postMessage reads:

```
{
  "action": "contact-request-pre-inform",
  "data": null
}
```

After the processing of exports finished, the contact-request postMessage is send to the iframe parent. It contains a JSON object, whose "data"-field reads for a single article:

```
{
  "obxUrl": "<url_to_obx>",
  "reopenUrl": "<url_to_ui_with_obx>",
  "pdfUrl": "",
  "articles": [
    {
      "itemId": "<article_itemId>",
      "manufacturerId": "<string>",
      "seriesId": "<string>",
      "baseArticleNumber": "<string>",
      "variantCode": "<string>",
      "shortText": "<string>",
      "longText": "<string>",
      "featureText": "<string>",
      "articleImage": "<url_to_article_image>",
    }
  ]
}
```

The reopen- and OBX-URLs are valid for four weeks. In case you want to be able to reconfigure a product after this period, you need to download the OBX to your servers and store it. For reconfiguration replace the OBX link in the URL with the link of the OBX to your server.

**Note:** Please provide us with your server URL<sup>1</sup>, so we can allow the upload to our servers from your do-main.

The PDF- and article image URLs are valid for max. 24 hours. If you want to persist them, copy them to your own server, too.

**Note:** Per default, the PDF URL in the postMessage is empty. Please contact us, if you want to activate the generation of a product sheet for the request button.

<sup>3</sup> <https://developer.mozilla.org/en-US/docs/Web/API/Window/postMessage>

## 2.3 postMessage elements

Element	Value	Description
action	contact-request	Indicates that the postMessage has been sent by the "Send request" button.
obxUrl	url	URL to OBX file. The link is valid for four weeks.
reopenUrl	url	Iframe URL of the configured product. Can be used for reconfiguration. The link is valid for four weeks.
pdfUrl	url	If activated: URL to generated PDF productsheet. The link is valid for 24 hours.
itemId	itemId	Contains the unique id of the article.
manufacturerId	String	ID of the manufacturer
seriesId	String	ID of the product line
baseArticleNumber	String	Base article number of the product
variantCode	String	Configuration code of the article
shortText	String	Short description of the article. Language as set for pCon.ui.
longText	String	Long description of the product. Language as set for pCon.ui.
featureText	String	Describes the configuration of the article. Language as set for pCon.ui.
articleImage	url	Link to image with main article and its child articles. The link is valid for 24 hours.

## 2.4 additional postMessage elements

The data of the postMessage can also contain several products since the configuration allows adding of so-called child products. For example, a worktop can have a CPU holder. In that case the JSON is structured in the following way:

```
{
  "obxUrl": "<url_to_obx>",
  "reopenUrl": "<url_to_ui_with_obx>",
  "pdfUrl": "",
  "articles": [
    {
      "itemId": "<main_article_itemId>",
      "manufacturerId": "<string>",
      "seriesId": "<string>",
      "baseArticleNumber": "<string>",
      "variantCode": "<string>",
      "shortText": "<string>",
      "longText": "<string>",
      "featureText": "<string>",
      "articleImage": "<url_to_article_image_with_child>",
    },
    {
      "itemId": "<child_itemId>",
      "manufacturerId": "<string>",
      "seriesId": "<string>",
      "baseArticleNumber": "<string>",
      "variantCode": "<string>",
      "shortText": "<string>",
      "longText": "<string>",
      "featureText": "<string>"
    }
  ]
}
```



## 2.5 postMessage in Case of PECs

In case of a PEC, there can be several main articles. In this case, there is one obx containing all articles, but no reopen-URL. Instead, every main article has an additional obx and reopen-URL containing only this main article and its childs:

```
{
  "obxUrl": "<url_to_obx_all_articles >",
  "reopenUrl": "",
  "pdfUrl": "",
  "articles": [
    {
      "itemId": "<main_article1_itemId>",
      ... (see above)
      "articleImage": "<url_to_article1_image_with_child>",
      "obxUrl": "<url_to_obx1_with_article1_and_child>",
      "reopenUrl": "<url_to_ui_with_obx1>",
    },
    {
      "itemId": "<child_itemId>",
      ... (see above)
    },
    {
      "itemId": "<main_article1_itemId>",
      ... (see above)
      "articleImage": "<url_to_article2_image >",
      "obxUrl": "<url_to_obx2>",
      "reopenUrl": "<url_to_ui_with_obx2>",
    }
  ]
}
```

## 3 pCon.ui for Shops

### 3.1 Integration in Shop System (OCI Punchout)

The specific feature of pCon.ui for Shops is the “Add to cart”-button. When the user clicks this button in pCon.ui the article information will be sent back to the shop system as OCI. For this, the integrated pCon.ui-links have to contain the parameter HOOK-URL. Note that the button **“Add to cart” is not visible, if no HOOK-URL is provided**. The shop has to make sure that the HOOK-URL is unique for each user session, so the response of pCon.ui can be handled separately for each user in the shop basket.

Example URL with demo data and HOOK-URL for plain text output of OCI data:

```
https://ui.pcon-solutions.com/#GATEKEEPER_ID=wcs_shop&moc=EGROFFICE&ban=4520&ovc=Polster.Ausf%3D15%3BArmlehne.Visible%3D1%3BGestell.FootMat%3D83&lang=en&sh=false&sp=false&HOOK_URL=https://www.easterngraphics.com/print_r.php
```

### 3.2 OCI Response

To send back the article information the Open Catalog Interface (OCI) is used. The OCI specification can be found here: <https://wiki.scn.sap.com/wiki/display/SRM/OCI+Documentation>

**NOTE:** An OCI can contain several products since the configuration allows adding of so called child products. For example a worktop can have a CPU holder and a screen. In that case the OCI contains the information for three articles. See [this section](#) for an example. Please make sure to handle those articles as a group in your cart!

The naming convention for the fields in the OCI is as follows:

NEW\_ITEM-<Field name>[<index>]. The field type is always CHAR.

Field name	Length	Description
NEW_ITEM-DESCRIPTION[n]	40	Description of the item
NEW_ITEM-MATNR[n]	40	SRM product number of the item
NEW_ITEM-QUANTITY[n]	15	Item quantity (1.)
NEW_ITEM-UNIT[n]	3	Quantity unit for item quantity (3.)
NEW_ITEM-PRICE[n]	15	Price of an item per price unit (1.)
NEW_ITEM-CURRENCY[n]	5	Item currency (3.)
NEW_ITEM-PRICEUNIT[n]	5	Price unit of the item (if empty, 1 is used)(2.)
NEW_ITEM-LEADTIME[n]	5	Delivery time of the item in days (2.)
NEW_ITEM-LONGTEXT_n:132[]	∞	Long text for the item (4.)
NEW_ITEM-VENDOR[n]	10	SRM vendor number (business partner) for the item
NEW_ITEM-VENDORMAT[n]	40	Vendor product number for the item
NEW_ITEM-MANUFACTCODE[n]	10	SRM manufacturer number of the item
NEW_ITEM-MANUFACTMAT[n]	40	Item's manufacturer part number
NEW_ITEM-MATGROUP[n]	10	SRM material group for the item
NEW_ITEM-SERVICE[n]	1	Flag: the item is a service.
NEW_ITEM-CONTRACT[n]	10	SRM contract to which the item refers
NEW_ITEM-CONTRACT_ITEM[n]	5	Item within the SRM contract
NEW_ITEM-EXT_QUOTE_ID[n]	35	Number of an external bid for this item (as reference for a subsequent purchase order)
NEW_ITEM-EXT_QUOTE_ITEM[n]	10	Item of external bid
NEW_ITEM-EXT_PRODUCT_ID[n]	40	Unique database key for this item in the catalog
NEW_ITEM-ATTACHMENT[n]	255	URL of the attachment (the attachment must be accessible for downloading under this URL)
NEW_ITEM-ATTACHMENT_TITLE[n]	255	Title of the attachment (if this is empty the file name from the URL above is used)
NEW_ITEM-ATTACHMENT_PURPOSE[n]	1	Purpose of the attachment. C corresponds here to configuration.
NEW_ITEM-EXT_SCHEMA_TYPE[n]	10	Name of a schema via which it was imported in the SRM Server
NEW_ITEM-EXT_CATEGORY_ID[n]	60	Unique key for an external category from the schema above, <b>independent</b> of the version of the schema
NEW_ITEM-EXT_CATEGORY[n]	40	Unique key for an external category from the schema above, <b>dependent</b> on the version of the schema
NEW_ITEM-SLD_SYS_NAME[n]	60	Name of a system in the System Landscape Directory (SLD)
NEW_ITEM-CUST_FIELD1[n]	10	User-defined field
NEW_ITEM-CUST_FIELD2[n]	10	User-defined field
NEW_ITEM-CUST_FIELD3[n]	10	User-defined field
NEW_ITEM-CUST_FIELD4[n]	20	User-defined field
NEW_ITEM-CUST_FIELD5[n]	50	User-defined field
NEW_ITEM-ITEM_TYPE[n]	1	Item type Root, Outline, Leaf. Introduced since SRM 7.0 onwards.
NEW_ITEM-PARENT_ID[n]	5	An integer field that defines what the parent of the current item is in the Hierarchy structure. This field will be blank for an item which does not have any parent. Introduced since SRM 7.0 onwards.

### 3.2.1 OCI 5.0 to OFML Mapping

OCI field	OFML data value
[NEW_ITEM-DESCRIPTION]	shortText
[NEW_ITEM-QUANTITY]	quantity
[NEW_ITEM-UNIT]	ST
[NEW_ITEM-PRICE]	price value as shown by pCon.ui <sup>4</sup>
[NEW_ITEM-CURRENCY]	price currency
[NEW_ITEM-VENDORMAT]	baseArticleNumber
[NEW_ITEM-LONGTEXT_1:132]	longText
[NEW_ITEM-CUST_FIELD1]	empty
[NEW_ITEM-CUST_FIELD2]	empty
[NEW_ITEM-CUST_FIELD3]	empty
[NEW_ITEM-CUST_FIELD4]	empty
[NEW_ITEM-CUST_FIELD5]	empty
[NEW_ITEM-EXT_QUOTE_ID]	empty
[NEW_ITEM-SERVICE]	empty
[NEW_ITEM-LEADTIME]	empty
[NEW_ITEM-EXT_QUOTE_ITEM]	empty
[NEW_ITEM-CONTRACT_ITEM]	empty
[NEW_ITEM-MANUFACTCODE]	manufacturer/ Catalog ID
[NEW_ITEM-MANUFACTMAT]	empty
[NEW_ITEM-EXT_PRODUCT_ID]	baseArticleNumber
[NEW_ITEM-MATGROUP]	empty
[NEW_ITEM-VENDOR]	empty
[NEW_ITEM-CONTRACT]	empty
[NEW_ITEM-PRICEUNIT]	1
[NEW_ITEM-ATTACHMENT]	URL for a 512x512 px thumbnail (valid for max 24 hours)

<sup>4</sup> Please note that this price may differ from the net price in the OBX file determined from the OFML data.

[NEW_ITEM-ATTACHMENT_PURPOSE]	C (means thumbnail corresponds to configuration)
[NEW_ITEM-ITEM_TYPE]	R (main article), empty (child articles)
[NEW_ITEM-PARENT_ID]	empty (main article), ID of parent of the child (child article)
[NEW_ITEM-EGR_CUSTOMDATA_PRICESERVICE]	bool
[NEW_ITEM-EGR_CUSTOMDATA_TAXCODE]	country Code, see parameter tax in <a href="#">this section</a> . FR@ECO refers to France incl. the ECO contribution.
[NEW_ITEM-EGR_CUSTOMDATA_VAT]	0 (net prices), absolute value of total VAT (gross prices), absolute value or empty (priceservice connected)
[NEW_ITEM-EGR_CUSTOMDATA_ECO]	gross (gross prices)/net (net prices) ECO contribution or empty
[NEW_ITEM-EGR_CUSTOMDATA_ECO_INFO]	further information for French ECO contribution
[NEW_ITEM-EGR_CUSTOMDATA_EAN]	EndArticleNumber, may differ from baseArticleNumber
[NEW_ITEM-EGR_CUSTOMDATA_BASKET_ID]	basketId of the article in OBX file
[NEW_ITEM-EGR_CUSTOMDATA_FEATURETEXT]	featureText
[NEW_ITEM-EGR_CUSTOMDATA_VARCODE]	variantCode
[NEW_ITEM-EGR_CUSTOMDATA_REOPEN_URL]	reopenUrl (valid for 24h)
[NEW_ITEM-EGR_CUSTOMDATA_OBX_URL]	URL to OBX file (valid for 24h)
[NEW_ITEM-EGR_CUSTOMDATA_PRICEHASH]	JSON web token (jwt) for price validation please consider contained public key
[NEW_ITEM-EGR_CUSTOMDATA_SERIESID]	series ID
[NEW_ITEM-EGR_CUSTOMDATA_ARTICLE_IMAGE]	image with all articles (main and child)

### 3.2.2 OCI Example

```
[NEW_ITEM-DESCRIPTION] => [1] => EA217 Soft Pad Group
[NEW_ITEM-QUANTITY] => [1] => 1
[NEW_ITEM-UNIT] => [1] => ST
[NEW_ITEM-PRICE] => [1] => 0.00
[NEW_ITEM-CURRENCY] => [1] => EUR
[NEW_ITEM-VENDORMAT] => [1] => 41238100
[NEW_ITEM-LONGTEXT_1:132] => [0] Soft Pad Chair EA 217
    office swivel chair with armrests
    back tilt mechanism
[NEW_ITEM-CUST_FIELD1] => [1] =>
[NEW_ITEM-CUST_FIELD2] => [1] =>
[NEW_ITEM-CUST_FIELD3] => [1] =>
[NEW_ITEM-CUST_FIELD4] => [1] =>
[NEW_ITEM-CUST_FIELD5] => [1] =>
[NEW_ITEM-EXT_QUOTE_ID] => [1] =>
[NEW_ITEM-SERVICE] => [1] =>
[NEW_ITEM-LEADTIME] => [1] =>
[NEW_ITEM-EXT_QUOTE_ITEM] => [1] =>
[NEW_ITEM-CONTRACT_ITEM] => [1] =>
[NEW_ITEM-MANUFACTCODE] => VI =>
[NEW_ITEM-MANUFACTMAT] => [1] =>
[NEW_ITEM-EXT_PRODUCT_ID] => [1] => 41238100
[NEW_ITEM-MATGROUP] => [1] =>
[NEW_ITEM-VENDOR] => [1] =>
[NEW_ITEM-CONTRACT] => [1] =>
[NEW_ITEM-PRICEUNIT] => [1] => 1
[NEW_ITEM-ATTACHMENT] => [1] => https://.../session-cache/ce8bf24ac2f65cb20fb.jpg
[NEW_ITEM-ATTACHMENT_PURPOSE] => [1] => C
[NEW_ITEM-ITEM_TYPE] => [1] => R
[NEW_ITEM-PARENT_ID] => [1] =>
[NEW_ITEM-EGR_CUSTOMDATA_PRICESERVICE] => [1] => false
[NEW_ITEM-EGR_CUSTOMDATA_TAXCODE] => [1] => DE
[NEW_ITEM-EGR_CUSTOMDATA_VAT] => [1] => 0
[NEW_ITEM-EGR_CUSTOMDATA_ECO] => [1] =>
[NEW_ITEM-EGR_CUSTOMDATA_ECO_INFO] => [1] =>
[NEW_ITEM-EGR_CUSTOMDATA_BASKET_ID] => [1] => 5207f1eb-8bf3-4976-997e-74d8a488d0f2
[NEW_ITEM-EGR_CUSTOMDATA_FEATURETEXT] => [1] => Base: chromed
    Cover material: leather
    Colour: dim grey
    Base on: castors hard, braked for carpet
    Packaging: standard, in plastic bag
[NEW_ITEM-EGR_CUSTOMDATA_VARCODE] => [1] => SOF_412381.SOFUNG010=0;
    SOF_412381.SOFBEZ010=4;SOF_412381.CLABZFLED010=21;
    SOF_412381.CLAFUS020=02;SOF_412381.STGVPEP010=ST
[NEW_ITEM-EGR_CUSTOMDATA_REOPEN_URL] => [1] =>
    https://ui.pcon-solu-tions.com/next/#GATEKEEPER_ID=demo_oci&
    lang=en&sp=true&HOOK_URL=https%3A%2F%2Fwww.easterngraphics.com%2F
    print_r.php&obx=https%3A%2F%2Fipa-eaiws1.easterngraphics.com%2F
    session-cache%2F9059a377-b288-4bfd-8427-29587a32fe93%2Fcut_buffer%2F
    00000002.obx
[NEW_ITEM-EGR_CUSTOMDATA_OBX_URL] => [1] =>
    https://ipa-eaiws1.easterngraphics.com/session-cache/
    9059a377-b288-4bfd-8427-29587a32fe93/cut_buffer/00000002.obx
[NEW_ITEM-EGR_CUSTOMDATA_PRICEHASH] => [1] => eyJ0eXAiOiJKV1Qi...
[NEW_ITEM-EGR_CUSTOMDATA_SERIESID] => [1] => Classic Collection
[NEW_ITEM-EGR_CUSTOMDATA_ARTICLE_IMAGE] => [1] => https://c-liveXa.pcon.eu/XXX.jpg
```

### 3.2.3 OCI Example with Child Article

```

[NEW_ITEM-DESCRIPTION] => [1] => example parent, a table
                        [2] => example child
[NEW_ITEM-QUANTITY] => [1] => 1
                        [2] => 1
[NEW_ITEM-UNIT] => [1] => ST
[2] => ST
[NEW_ITEM-PRICE] => [1] => 0.00
[2] => 0.00
[NEW_ITEM-CURRENCY] => [1] => EUR
[2] => EUR
[NEW_ITEM-VENDORMAT] => [1] => 0000001
[2] => 0000002
[NEW_ITEM-LONGTEXT_1:132] => [0] => some long description of the example parent
[NEW_ITEM-CUST_FIELD1] => [1] =>
[2] =>
[NEW_ITEM-CUST_FIELD2] => [1] =>
[2] =>
[NEW_ITEM-CUST_FIELD3] => [1] =>
[2] =>
[NEW_ITEM-CUST_FIELD4] => [1] =>
[2] =>
[NEW_ITEM-CUST_FIELD5] => [1] =>
[2] =>
[NEW_ITEM-EXT_QUOTE_ID] => [1] =>
[2] =>
[NEW_ITEM-SERVICE] => [1] =>
[2] =>
[NEW_ITEM-LEADTIME] => [1] =>
[2] =>
[NEW_ITEM-EXT_QUOTE_ITEM] => [1] =>
[2] =>
[NEW_ITEM-CONTRACT_ITEM] => [1] =>
[2] =>
[NEW_ITEM-MANUFACTCODE] => [1] => WI
[2] => WI
[NEW_ITEM-MANUFACTMAT] => [1] =>
[2] =>
[NEW_ITEM-EXT_PRODUCT_ID] => [1] => 0000001
[2] => 0000002
[NEW_ITEM-MATGROUP] => [1] =>
[2] =>
[NEW_ITEM-VENDOR] => [1] =>
[2] =>
[NEW_ITEM-CONTRACT] => [1] =>
[2] =>
[NEW_ITEM-PRICEUNIT] => [1] => 1
[2] => 1
[NEW_ITEM-ATTACHMENT] =>
[1] => https://c-live1b.pcon.eu/session-cache/.../objects/some_long_number.jpg
[2] => https://c-live1b.pcon.eu/session-cache/.../objects/some_long_number2.jpg
[NEW_ITEM-ATTACHMENT_PURPOSE] => [1] => C
[2] => C
[NEW_ITEM-ITEM_TYPE] => [1] => R
[2] =>
[NEW_ITEM-PARENT_ID] => [1] =>
[2] => 1
[NEW_ITEM-EGR_CUSTOMDATA_PRICESERVICE] => [1] => false
[2] => false
[NEW_ITEM-EGR_CUSTOMDATA_TAXCODE] => [1] => DE
[2] => DE
[NEW_ITEM-EGR_CUSTOMDATA_VAT] => [1] => 0
[2] => 0

```

```
[NEW_ITEM-EGR_CUSTOMDATA_ECO] => [1] =>
    [2] =>
[NEW_ITEM-EGR_CUSTOMDATA_ECO_INFO] => [1] =>
    [2] =>
[NEW_ITEM-EGR_CUSTOMDATA_BASKET_ID] => [1] => basket_ID_of_parent_article
    [2] => basket_ID_of_child_article
[NEW_ITEM-EGR_CUSTOMDATA_FEATURETEXT] =>
    [1] => more detailed text for parent article
    [2] => more detailed text for child article
[NEW_ITEM-EGR_CUSTOMDATA_VARCODE] =>
    [1] => TABLE22210.KAT01=KAT01;TABLE22210.GSN01=GSN01;TBLE22210.GS0222=GS051;
    TABLE22210.GAU100=GAU17;TABLE22210.SCF01=SCF99
    [2] => FILZ222.KAT01=KAT01;FILZ222.FILZ222=FILZ08
[NEW_ITEM-EGR_CUSTOMDATA_REOPEN_URL] => [1] => URL for parent article
    [2] =>
[NEW_ITEM-EGR_CUSTOMDATA_OBX_URL] => [1] => URL for parent article
    [2] =>
[NEW_ITEM-EGR_CUSTOMDATA_PRICEHASH] => [1] => jwt containing all articles
    [2] => jwt containing all articles
[NEW_ITEM-EGR_CUSTOMDATA_SERIESID] => [1] => 05
    [2] => 05
[NEW_ITEM-EGR_CUSTOMDATA_ARTICLE_IMAGE] => [1] => https://c-liveXa.pcon.eu/XXX.jpg
    [2] => https://c-liveXa.pcon.eu/XXX.jpg
[NEW_ITEM-LONGTEXT_2:132] => [0] => example long text child
```

### 3.3 Reconfiguration

Once the article has been transferred to the cart you can give the user the possibility for reconfiguration by using the reopenURL of the OCI file (NEW\_ITEM-EGR\_CUSTOMDATA\_REOPEN\_URL). With this URL the article is shown with the configuration set by the user.

**Note:** For article groups (parent and child articles) the reopenUrl is only available at the main article. The group can only be reconfigured as whole group. Make sure to replace the whole group when transferring the changed article group back to the cart.

**Note:** The reopenURL is only available for 24 hours.

In case you want to be able to reconfigure a product after 24 hours, you need to download the OBX from NEW\_ITEM-EGR\_CUSTOMDATA\_OBX\_URL to your servers and store it. For reconfiguration replace the OBX link in the URL with the link of the OBX to your server.

**Note:** Please provide us with your server URL<sup>1)</sup>, so we can allow the upload to our servers from your domain.



### 3.4 Price Validation

If you are using the price service API (see [this section](#)), you should ignore the following and use your price service for validation.

Without a price service you can use the json web token (jwt) contained in the field NEW\_ITEM-EGR\_CUSTOM-DATA\_PRICEHASH of the OCI to validate the price. The jwt leads to the following information:

```
{
  "items": [
    {"calculation": {
      "itemNetValue": {
        "formatted": "100,00 EUR",
        "currency": "EUR",
        "value": 100
      },
      "finalValue": {
        "formatted": "119,00 EUR",
        "currency": "EUR",
        "value": 119.00
      }
    },
    "basketId": "basketID",
    "quantity": 1,
    "articleNumber": "baseArticleNumber",
    "manufacturerName": "manufacturer",
    "seriesName": "seriesID",
    "finalValue": "price",
    "ofmlVarCode": "varCode"
  ]
}
```

The field itemNetValue contains the net price while finalValue is the gross price for the country in your configuration. This price can be compared with the price in the OCI field NEW\_ITEM-PRICE and the OBX field itemPrice. Over all three formats the articles can be related with the basketID.

**Note:** The OBX file always contains net prices while the OCI contains net or gross prices. The jwt always contains both.

In case of child articles or a PEC, the list in items has more than one entry:

```
{
  "items": [
    {
      "calculation": ...,
      "basketId": "basketID1",
      ...
    },
    {
      "calculation": ...,
      "basketId": "basketID2",
      ...
    },
    ...
  ]
}
```

The jwt signature can be verified with the following public key:

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEASlq93R1JBeHZbE5pq0qt
DZff9aWU5LqCBe5SvuzBgEGfvkYYQrwhFGHH8C11iPGh0tWmdCqFf41dN49Wl/U4
VsoLrkxYdKazy1T993n63NdRjXGoaKGG1BoYJMKrJHq10Rf8pA1Nvoo5sAg1iWwM
nCEYZMYUtKbyN9KIq+yXdD9rdev/bhG2fkNHv7ExT0+iHRcUE05ST+ksADC5vcP5
S7966Dc3S/Ex2U/YS6i+HBA4tbCQ24y++/belvTeKHn5wNaOKexFuqIbKL9FfUPq
XdQWy0hiddaFIo5Q9KQCWygxD1WwW8X6wcrv41q+yemGJPjbWRDQUHHZv9B9tXcN
zwIDAQAB
-----END PUBLIC KEY-----
```

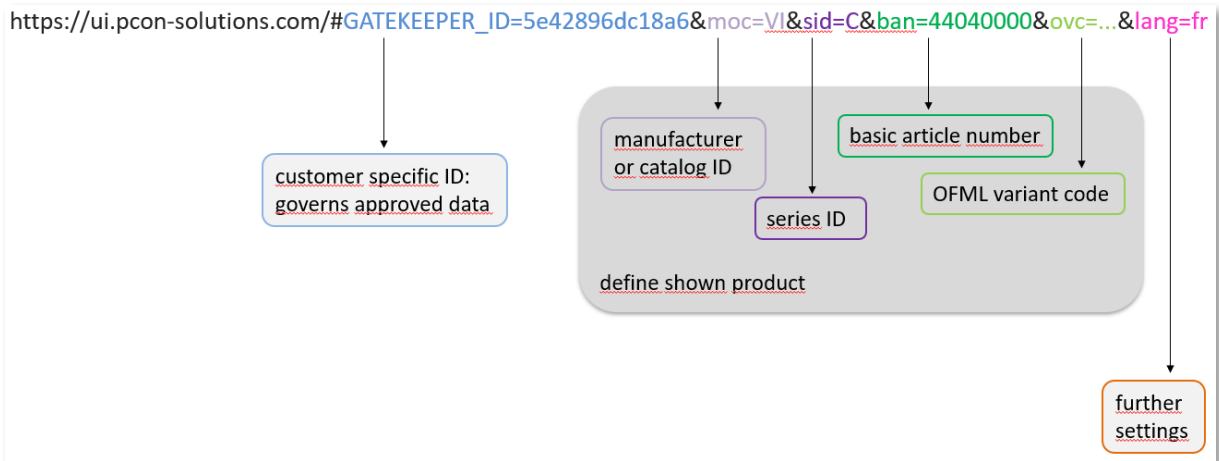
### 3.5 Checklist before Going Live

Before launching your new shop solution we suggest to check your implementation at least for the following topics.

- Iframe is shown in correct size
- All article information are transferred correctly to the shopping cart
  - Image
  - Description
  - Price
  - Quantity
  - Child article including their article information
- Link for reconfiguration is available in the cart.
- The reconfigured product replaces the original product.
- In the shopping cart an article group should only be allowed to be deleted and reconfigured as whole group. Furthermore the quantity should be only allowed to be changed for the whole group.
- Reconfiguration for article group replaces whole article group in the cart

## 4 Individualization by URL Parameters

After setting up your personalized pCon.ui a base URL will be provided. By adding different URL parameters you can control the layout of pCon.ui as well as the entry point. The final URL will then be used as iframe source on the website. The links for articles are constructed as follows:



Note that some of the URL parameter can override the default settings from the initial setup. For instance, you can hide the price with the URL parameter "sp=false" even if your pCon.ui is configured to show prices.

**Addition:** In case of articles with metaproperties or PECs (pCon Exchange Container) URLs to predefined articles differing from the default can **not** be generated by URL parameters. Therefore you have to use an obx-reopen-url.

## 4.1 Basic Parameters for showing up an article

Parameter	Presence	Description	Value
GATEKEEPER_ID	mandatory	personalized ID	provided by EasternGraphics
moc	optional	manufacturer or catalog can be changed, if you have access to more than one catalog or manufacturer	e.g. EG
ban	optional	base article number must be a valid article number for the chosen catalog (manufacturer)	e.g. 4520
sid	optional	series id must be available if a base article number is used in several series	e.g. OFFICE2
ovc	optional	OFML variant code requires the corre- sponding base article number (ban)	e.g. 4520
cep	optional	catalog entry point*	e.g. egroffice%3A0
crp	optional	catalog root path*, defines reachable level of the catalog, needs to be part of cep, deactivates the search field	e.g. egroffice%3A0
obx	optional	Load obx file from URL**	URL to obx file

## 4.2 Parameters for changing behavior of 3D view

Parameter	Presence	Description	Value
fso	optional	show preview in full screen mode, hides property editor at right side	Boolean: true or false
alpha	optional	sets the alpha rotation (x-axis) for custom camera view (needs to be set with beta)	number
beta	optional	sets the beta rotation (y-axis) for custom camera view (needs to be set with alpha)	number
lft	optional	lock article at center of 3D view toggles the default	Boolean: true or false
rrz	optional	Restrict rotation to z-axis, automatically activates lft	Boolean: true or false
sb	optional	sets scene background toggles the default	white, radiantgrey, black
sibd	optional	Show OAP interactors by default	Boolean: true or false

## 4.3 Parameters for changing Article Property view

Parameter	Presence	Description	Value
emp	optional	Enable Metatypes toggles the default; should be true Attention: breaks some OFML-data and deactivates OAP interactors	Boolean: true or false
lang	optional	controls the language depends on the OFML data, the GUI itself supports	one of the following: da, de, en, es, fr, it, nl, pt, sv, pl, cs, tr, ro
ofmlLang	optional	Controls language of OFML data by default the GUI language is used	
sh	optional	show breadcrumb navigation header toggles the default	Boolean: true or false
msi	optional	Show manufacturer in subinfo toggles the default	Boolean: true or false
sph	optional	Show names of property groups as headlines in property list	Boolean: true or false

Parameter	Presence	Description	Value
sup	optional	Show uneditable properties toggles the default	Boolean: true or false
ssi	optional	Show series in subinfo toggles the default	Boolean: true or false
san	optional	Show article name toggles the default	Boolean: true or false
asi	optional	Show article number in subinfo toggles the default	Boolean: true or false
hde	optional	hide description below property list, toggles the default	Boolean: true or false
sp	optional	show price, toggles the default	Boolean: true or false
tax	optional	country code for VAT, toggles the default Only applicable if gross prices are used	Country code according to ISO 3166-1 alpha-2 (e.g. DE)

#### 4.4 other parameters

Parameter	Presence	Description	Value
cad	optional	activates cad download function (only if CAD Formats have been ordered)	Boolean: true or false
adv	optional	switches the Article Drawer Variant	single or modular ( <a href="#">this section</a> )
pev	optional	Switches Property Editor Variant	full or compact ( <a href="#">this section</a> )
customFields	optional	transfer custom information to service APIs	encoded string e.g. user%3Dtrue

\* **Note:** Deep links to folders are unstable and can break with each update or reinstallation of OFML-data. Hence, you should use the parameters cep and crp with care and only for high catalog folder levels.

\*\* **Note:** The domain of the obx URL needs to be whitelisted by EasternGraphics. Please [send us](#) the domains from which you want to load obx files

## 4.5 Helper Tool: Admin Panel

In addition, there is a helper tool available to determine the correct values for the parameters and to generate the URLs. The admin control panel can be activated by adding the URL parameter "ap=true" to the base URL.

See for example [https://ui.pcon-solutions.com/#GATEKEEPER\\_ID=wcs\\_architecture&lang=en&ap=true](https://ui.pcon-solutions.com/#GATEKEEPER_ID=wcs_architecture&lang=en&ap=true).

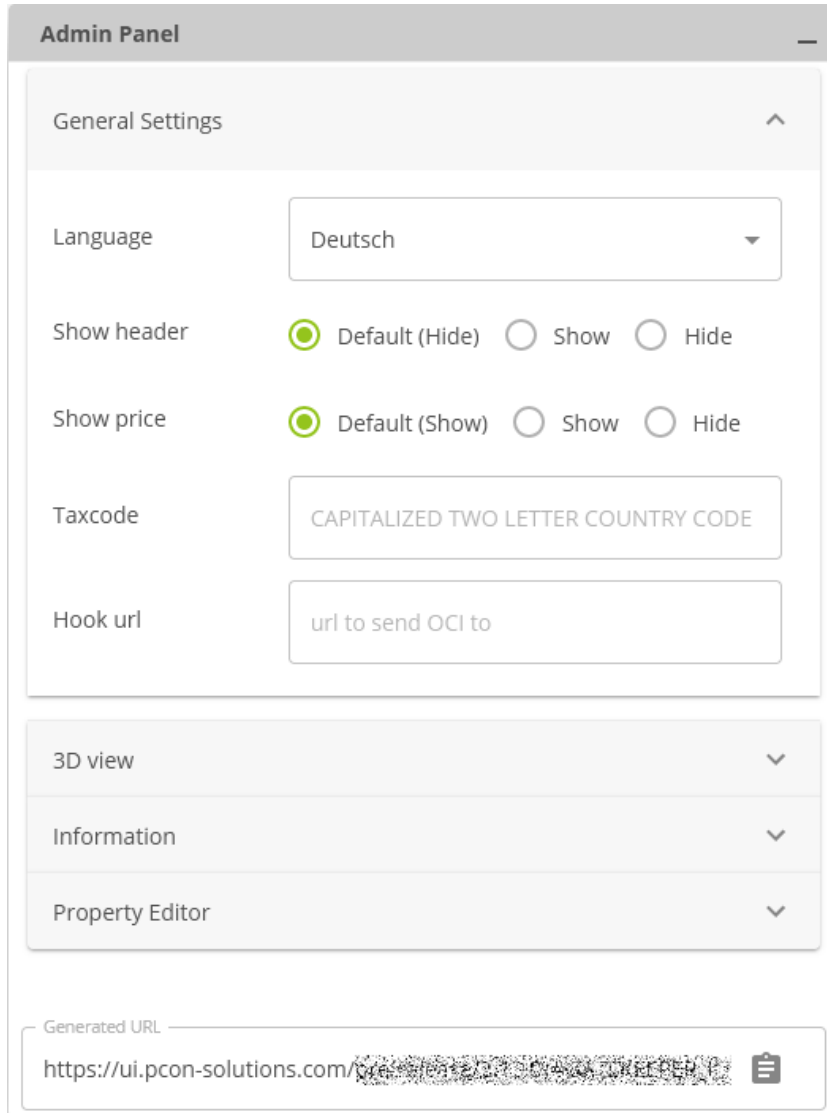


Figure 1 Admin Panel on catalog level

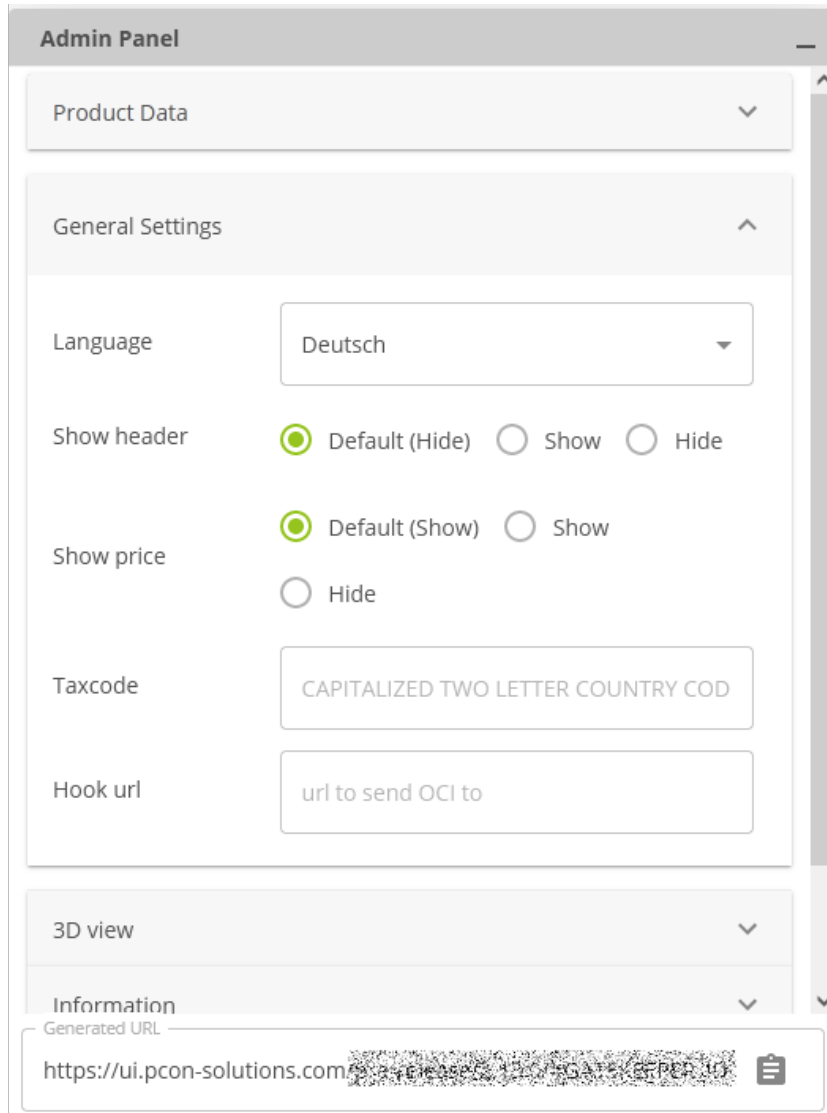


Figure 2 Admin Panel on product level

The field "Use current camera" in "3D View" allows you to include the current perspective, i.e. viewing angle and zoom, to the generated link (by adding the URL parameter alpha and beta). If you want to alter the standard camera settings for your pCon.ui instance, send us the generated link by e-mail please.

The last field Generated URL shows the URL with all settings made in the field above. This URL can be copied with the clipboard icon and used as iframe source in your website.



## 5 User Interaction Messages / User Tracking / Error Handling

### 5.1 General User Tracking

This feature needs to be activated for your pCon.ui instance. If so, pCon.ui sends postMessages<sup>5</sup> to the parent window of the iframe with every user action. The receiving website, i.e. your website, is free to handle this event as needed. For instance, the information can be used to alter the surrounding web page or to track the user actions.

The postMessage sends a json object with the following information:

```
{
  "action": "message",
  "data": {
    "type": " ",
    "subtype": " "
  }
}
```

The following types and subtypes are currently supported:

Parameter	Value	Description
type	articleInsert	pCon.ui was started and an article was inserted
type	propertyChanged	The article was configured.
type	cadDownload	CAD file was downloaded. File format is given in subtype "typeValue"
type	pdfDownload	PDF downloaded
type	snapshot	Snapshot made
subtype	manufacturerId	e.g. EG
subtype	seriesId	e.g. OFFICE2
subtype	baseArticleNumber	e.g. 4520
subtype	variantCode	
subtype	ofmlVariantCode	
subtype	typeValue	File format, e.g. "DWG". Only available for type "cadDownload"

If you want to test out the user interaction messages you can use our iframe tester tool:  
<https://ui.pcon-solutions.com/iframe.html>

<sup>5</sup> <https://developer.mozilla.org/en-US/docs/Web/API/Window/postMessage>

## 5.2 Error Handling

When integrated in an iframe pCon.ui sends postMessages to the parent window in case of errors (e.g. article not found). The receiving website, i.e. your website, is free to handle this event as needed (e.g. show error message to the user, write log or email notification to administrator).

The HTML postMessage sends a json object with the following information:

```
{
  "action": "error",
  "data":
  {
    "type": " ",
    "subtype": " "
  }
}
```

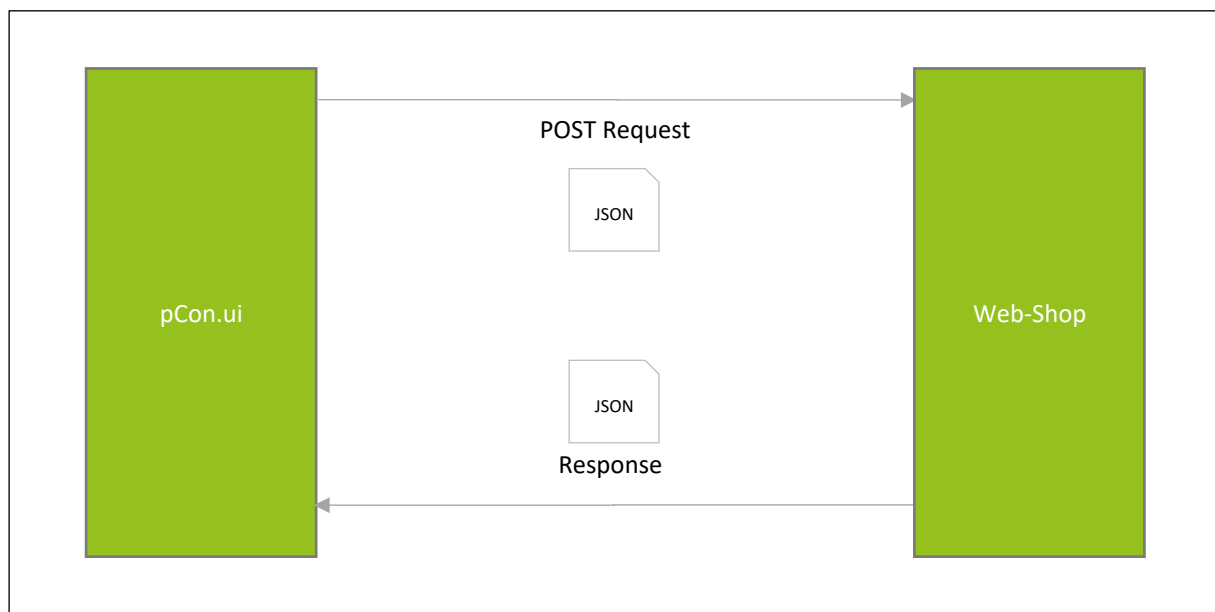
The following types and subtypes are currently supported:

Parameter	Value	Description
type	article	error in connection with an article
subtype	not-found	The article does not exist in the product data.
subtype	insert-failed	The article could not be inserted. Usually due to an error in the OFML data.
type	catalog	
subtype	not-found	The catalog node does not exist in the product data.
type	configuration	The insertion of a configuration by the provided OBX failed.
type	unlicensed	The gatekeeper-ID is not allowed to be opened in pCon.ui (generally or with the used parent URL).
type	priceservice	
subtype	not-available	The price-service is not available.
subtype	empty-response	The answer does not contain price data.
type	articledata	A property change could not be done.
subtype	manufacturer   articleNumber   message	Details about the article and the occurred error.

## 6 Price Update Webservice API

One important use case for pCon.ui is the integration into shop systems, but often the shops don't want to use the product prices available in the OFML data. For example, there are no prices available for different currencies or promotion discounts of the shops cannot be applied. Therefore, a functionality is needed that allows the shop to overwrite the prices coming from the OFML data in pCon.ui. The provided API can also be used with pCon.ui for Websites (Lead).

### 6.1 Concept



When the user inserts a product or configures it, a POST request with a JSON including a link to an OBX file is sent to a webservice of the shop. The OBX file contains all information of the product (manufacturer, article code, configuration, ...). The webservice of the shop downloads and parses this XML file and gets the required information to determine the price. The new price, currency and the article ID are sent back to pCon.ui as JSON file. pCon.ui reads this file and displays the new price to the user.

**NOTE:** The URL<sup>1)</sup> of the webservice<sup>2)</sup> has to be sent to EasternGraphics, so it can be set as parameter in the corresponding GatekeeperID configuration on the server.

## 6.2 POST Request

The POST request sent by pCon.ui looks like the following:

```
{
  "obxUrl": "<url>"
  "customFields": ""
}
```

The URL leads to an OBX file. This file contains all information of the inserted products. Note that the link expires after 24 hours. If you desire to persist the obx file, please store it on your own server.

If customFields parameter (see section about Custom Fields) was set in pCon.ui URL, the JSON contains this as well.

**NOTE:** An OBX file can contain several products since the configuration allows adding of so called child products. For example a worktop can have a CPU holder and a screen. In that case the OBX contains the information for 3 articles.

If required, a detailed OBX specification can be found [here](#). The elements important for the price service are:

Element	Attribute	Description
bskArticle		The bskArticle element contains all information about the product like manufacturer, article code, variant code, description and price.
	basketId	Contains the unique ID for the basket article. The id is important when sending back the price information to pCon.ui.
	itemType	BasketArticle – Products of this type have to be used for price evaluation in the shop system. BasketAggregate – Products of this type have to be used for price evaluation in the shop system. BasketPartialPlanning – those elements can be ignored.
manufacturer		This element contains information about the manufacturer including the name of the manufacturer in different languages.
	id	ID of the manufacturer
series		The series element contains information about the product line like the name in different languages.
	id	ID of the product line
artNr		This element contains the information about the article code.
	type	base - Base article number final - Final article number varcode - Configuration of the article
description		Description of the article
	type	short – short description of the article in different languages long – long description of the article in different languages feature – describes the configuration of the article in different languages
itemPrice		This element contains the price information of the article. Use for a web shop the element with type='sale' and pd='1'. This is always the net price from the OFML-data.
	type	purchase – purchase price of the article sale – sales price of the article
	currency	Currency of the price
	value	Price
taxInfo		Contains special tax information, e.g. for the French ECO contribution. This does not include any information regarding a VAT.
	typeld	e.g. ECO_FR
	categoryId	Type of French ECO contribution, e.g. workplace_wood

### 6.2.1 Example incl. French ECO contribution:

```
<bskArticle basketId='c45f23d5-8e3f-43a5-b542-f5bddcb14194' itemType='BasketArticle'>
  <manufacturer id='EGR'>
    <name lang=' ' >EasternGraphics GmbH</name>
  </manufacturer>
  <series id='SE'>
    <name lang=' ' >Series Name</name>
  </series>
  <artNr type='base'> NU3456 </artNr>
  <artNr type='final' default='1'> NU3456-TE </artNr>
  <artNr type='varcode'>CLASS1.PROP1=01;CLASS1.PROP2=06;CLASS2.PROP1=ABC</artNr>
  <description type='short' default='0'>
    <text lang='de'>Example Product</text>
  </description>
  <itemPrice type='purchase' currency='EUR' value='103.0' />
  <itemPrice type='sale' pd='1' currency='EUR' value='103.0' />
  <taxInfos>
    <taxInfo typeId='ECO_FR' categoryId='workplace_wood' />
  </taxInfos>
</bskArticle>
```

### 6.2.2 Example with child article:

```
<bskArticle basketId='main_article_ID' itemType=' BasketAggregate' >
  ... (see example above)
  <bskArticle basketId=child1_ID' itemType='BasketAggregate' subItem='1'>
    ... (see example above)
  </bskArticle>
</bskArticle>
```

## 6.3 Price Calculation

Prices can be manipulated in many different ways:

manipulation Type	description	things to consider
change price	manipulate price given by manufacturer to your needs (eg. give discount)	take given price, do calculations with it, send new price back with mandatory fields (see <a href="#">this section</a> )
visualize discount	manipulate price given by manufacturer eg. with given discount percentage or set new price	take given price, set discount, send default and new price back with optional fields (see <a href="#">this section</a> )
gross price	show gross price of articles in pCon.ui	given price is by default net price so you will have to do the correct calculation of gross price on your side considering the vat tax for your country and send new price back with mandatory fields (see <a href="#">this section</a> ) there will be no conversion from net to gross price on our side if you use a price service
change currency	change the currency given by manufacturer	default currency given by the manufacturers is mostly EUR. If you want to display a different currency you have to do the conversion from the default currency to the desired one by yourself and send the new price and currency back with mandatory fields (see <a href="#">this section</a> ) there will be no conversion from EUR to other currencies on our side if you use a price service

## 6.4 Response

### JSON

After the shop system determined the price, it sends the information back to pCon.ui as JSON<sup>6</sup>. The JSON file needs to contain the following elements:

Element	Presence	Description
currency	mandatory	currency of the price(s)
taxCode	optional	code for tax country
articles	mandatory	list that contains information and prices for all articles
basketId	mandatory	ID of the product as defined in the OBX file.
salesPrice	mandatory	New sales price of the article. Should include the French eco fee, if applicable.
vat	optional	VAT for this article
eco	optional	French eco fee for this article
eco_info	optional	additional information for French eco fee
originalPrice	optional	original Price
discount	optional	given discount (as number) percentage towards the original price
compositePrice	optional	compositePrice will get the new salesPrice

**NOTE:** If the OBX file contained more than one product the JSON response has to provide the new price information for all articles.

<sup>6</sup> see ext. wiki link providing [json syntax](#)



#### 6.4.1 Example without discount

```
{
  "currency": "EUR",
  "taxCode": "DE",
  "articles": [
    {
      "basketId": "f7d06042-5cda-49b7-a2f1-47a4901e3fa9",
      "salesPrice": "1.0",
      "vat": "0.19",
      "eco": "0.1",
      "eco_info": "eco info text"
    }
  ]
}
```

#### 6.4.2 Example with child articles

```
{
  "currency": "EUR",
  "taxCode": "DE",
  "articles": [
    {
      "basketId": "main_article_id",
      "salesPrice": "2.0",
      "vat": "0.4",
      "eco": "0.2",
      "eco_info": "eco info text"
    },
    {
      "basketId": "child_article_id",
      "salesPrice": "1.0",
      "vat": "0.19",
      "eco": "0.1",
      "eco_info": "eco info text"
    }
  ]
}
```

### 6.4.3 Example with visible discount

```
{
  "currency": "EUR",
  "taxCode": "DE",
  "articles": [
    {
      "basketId": "f7d06042-5cda-49b7-a2f1-47a4901e3fa9",
      "salesPrice": "135.75",
      "vat": "0.19",
      "eco": "0.1",
      "eco_info": "eco info text"
    }
  ],
  "originalPrice": "181.0",
  "discount": "25",
  "compositePrice": "135.75"
}
```

Leads to:

You have the opportunity to change the style,  
e.g. to this:

EUR 181,00 -25%  
**Gesamt: EUR 135,75**  
inkl. EUR 1,26 EUR Eco Abgabe, zzgl. MwSt

EUR 181,00 **-25%**  
**Gesamt: EUR 135,75**  
inkl. EUR 1,26 EUR Eco Abgabe, zzgl. MwSt

For style customization see [this section](#)

#### 6.4.4 Example with visible discount (only discount percentage) without showing original price

```
{
  "currency": "EUR",
  "taxCode": "DE",
  "articles": [
    {
      "basketId": "f7d06042-5cda-49b7-a2f1-47a4901e3fa9",
      "salesPrice": "135.75",
      "vat": "0.19",
      "eco": "0.1",
      "eco_info": "eco info text"
    }
  ],
  "discount": "25",
  "compositePrice": "135.75"
}
```

Leads to:

-25%

**Gesamt: EUR 135,75**

inkl. EUR 1,26 EUR Eco Abgabe, zzgl. MwSt

## 7 PropertyService API

The PropertyService API allows customers to decide, which properties should be shown to the user. This can be useful, for instance, if a shop does not offer all variants of a product. Technically the API works similar to the PriceService API, see [this section](#). Please also consider the requirements to your server settings<sup>2)</sup>.

**NOTE:** The URL<sup>1)</sup> of the webservice<sup>2)</sup> has to be sent to us, so it can be set as parameter in the corresponding GatekeeperID configuration on the server.

The properties can be adapted on two levels: the properties themselves and their values. For both levels the post requests are sent to the same webservice URL. In any case, the properties are not “removed” from the article, they are just not shown to the user.

### 7.1 Change available Properties

#### 7.1.1 POST Request

The POST request sent by pCon.ui on every event that may changes the property list looks like the following:

```
{
  "type": "property",
  "manufacturerId": "<manufacturerId>",
  "seriesId": "<seriesId>",
  "baseArticleNumber": "<baseArticleNumber>",
  "properties": Array<property>
}
```

The field properties contains all available properties in the format <property>, which is defined as

```
{
  "propClass": "<propClass>",
  "propKey": "<propKey>",
  "customFields": ""
}
```

If customFields parameter was set in pCon.ui URL, the JSON contains this as well.

Example:

```
{
  "type": "property",
  "manufacturerId": "BJ",
  "seriesId": "EV",
  "baseArticleNumber": "EV151",
  "properties": [
    {
      "propClass": "MetaProperties",
      "propKey": "GCOLLTYPE"
    },
    {
      "propClass": "MetaProperties",
      "propKey": "GCOLL"
    },
    {
      "propClass": "EV152",
      "propKey": "SM03"
    },
    {
      "propClass": "EV152",
      "propKey": "SM05"
    }
  ]
}
```

### 7.1.2 Response

The response can contain a whitelist as well as a blacklist. If both are sent, the blacklist will be ignored. If none is returned, all properties are shown.

```
{
  whitelist: Array<property>,
  blacklist: Array<property>
}
```

Element	Presence	Description
whitelist	optional	List of <property>s that should be visible
blacklist	optional	List of <property>s that should be hidden

Example:

```
{
  "whitelist": [
    {
      "propClass": "MetaProperties",
      "propKey": "GCOLLTYPE"
    },
    {
      "propClass": "EV152",
      "propKey": "SM03"
    }
  ]
}
```

## 7.2 Change available property values

### 7.2.1 Post Request

The POST request sent by pCon.ui on every event that may changes the property list looks like the following:

```
{
  "type": "choice",
  "manufacturerId": "<manufacturerId>",
  "seriesId": "<seriesId>",
  "baseArticleNumber": "<baseArticleNumber>",
  "propClass": "<propClass>",
  "propKey": "<propKey>",
  "currentChoice": "<propValue>",
  "choices": Array<propValue>
}
```

The field choices contains all available property values as a list of string values.

```
{
  "type": "choice",
  "manufacturerId": "BJ",
  "seriesId": "EV",
  "baseArticleNumber": "EV151",
  "propClass": "EV152",
  "propKey": "SM01",
  "currentChoice": "CI",
  "choices": [
    "CI",
    "ER",
    "MH",
    "S58",
    "KI",
    "RO"
  ]
}
```

### 7.2.2 Response

The response has to contain a whitelist. If the current choice is not part of this list, it will be automatically added in order to avoid empty choice lists. In case of an erroneous response, all property values will be shown.

```
Array<propValue>
```

Example:

```
["CI", "ER", "MH"]
```

## 8 Access Product Information Management (PIM)

You can decide whether you want to get PIM Data (Product Information Management) from the articles of the manufacturer. To activate this feature please [contact us](#).

### 8.1 postMessage Response

Product information from the following categories may be provided via the interface. The order of the categories in the table below corresponds with the standard presentation order of the categories. However, the presentation of the categories may vary in the individual pCon products depending on the intended use of the respective pCon application.

Category	Description	Preferred media types
PRODUCT_BROCHURE	Product brochure	PDF
PRODUCT_INFORMATION	General product information	Text, PDF
CUSTOM_	Prefix for user-defined categories (for example, CUSTOM_DATA_SHEET)	Any of the supported
SPECIFICATIONS	Specifications, tender specifications	PDF
CERTIFICATE	Certificate, e.g. ISO 9001, TÜV GS, FSC	PDF with image as thumbnail, Text
ENVIRONMENTAL_INFORMATION	Environmental information	Text, PDF
MATERIAL_INFORMATION	Material images, material documents, material description	PDF, Text
USER_INSTRUCTIONS	Instructions for use	PDF
CARE_INSTRUCTIONS	Care instructions	PDF
OPERATING_INSTRUCTIONS	Operating instructions	PDF
ASSEMBLY_INSTRUCTIONS	Assembly instructions	PDF



## 8.2 Supported Mediatypes

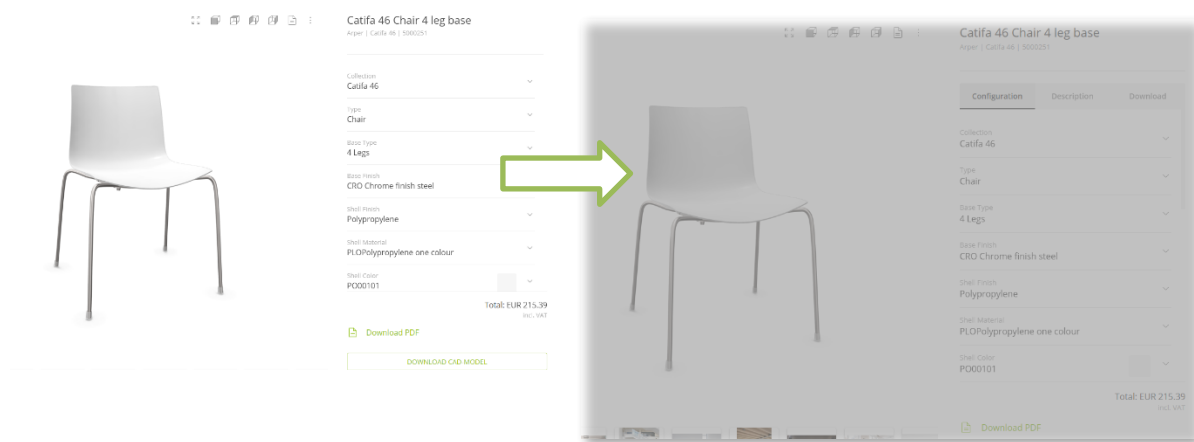
Media type	Description
text/plain	Simple text (UTF-8)
text/markdown	Markdown text (UTF-8), see below
application/pdf	PDF

## 9 Adapt Variance of pCon.ui: Article Drawer and Property Editor

You can change the Variance of the Article Drawer and the Property Editor in pCon.ui.

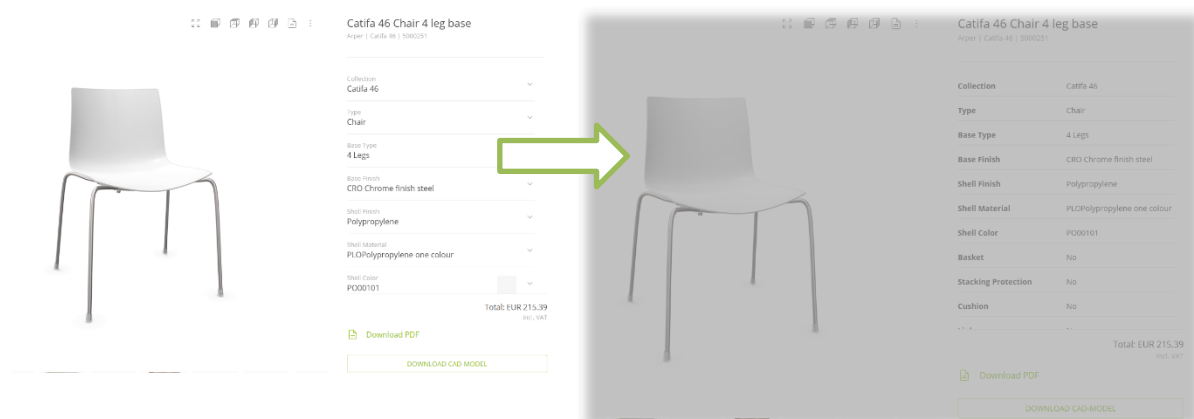
### 9.1 Modular Article View

If you set the parameter `adv=modular` the pCon.ui will switch to a look like this



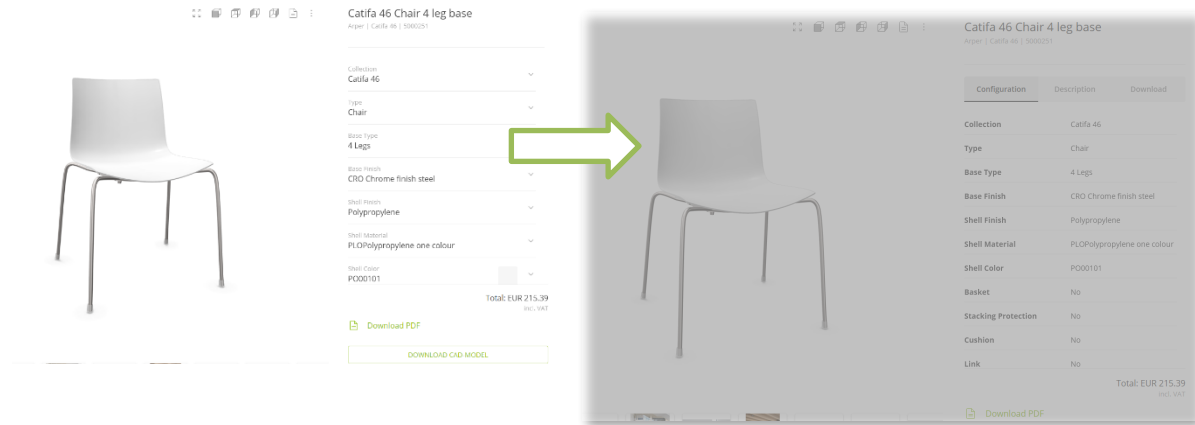
### 9.2 Compact Property Editor View

if you set the parameter `pev=compact` the pCon.ui will switch to a look like this:



### 9.3 Combine the views

if you set the parameter `adv=modular&pev=compact` the pCon.ui will switch to a look like this:



## 10 Adapt appearance of pCon.ui: MUI theme and CSS

If you want pCon.ui to adapt optically to your website, you can manipulate the colors by providing new values for material ui colors and adapt some components by providing a CSS file.

### 10.1 Color scheme

Your custom colors should be gathered in a JSON file on your server. Then, you need to send a publicly available URL to this file to EasternGraphics. We will then connect this URL with your pCon.ui instance.

**NOTE:** The URL<sup>1)</sup> of the color scheme has to be sent to us, so it can be set as parameter in the corresponding GatekeeperID configuration on the server.

The JSON file can contain the following fields:

Part	Kind	Description
primary	main	Highlight color for hover effect and filled buttons
text	primary	Text color for title, property values, property group names
text	secondary	Text color for subtitle, description, property names
background	default	Background color of iframe-region
background	paper	Background color of page (only for usage without iframe)
action	active	Color of icons in 3D view
divider		Color of lines above and below property list

Example:

```
{
  "primary": {
    "main": "#868686"
  },
  "text": {
    "primary": "#000000",
    "secondary": "#000000"
  },
  "background": {
    "default": "#ffffff"
  },
  "divider": "#000000"
}
```

## 10.2 Style Components

The commands to adapt the style of your pCon.ui instance should be in a CSS file on your server.

**NOTE:** The URL<sup>1)</sup> of the CSS file has to be sent to us, so it can be set as parameter in the corresponding GatekeeperID configuration on the server.

The following components can be customized:

Component	Description
egr_add_to_basket	“Add to cart”-button
egr_add_to_basket_x	x between number field and “Add to cart”-button
egr_add_to_basket_number	number field in front of “Add to cart”-button egr_add_to_basket_number fieldset and egr_add_to_basket_number input are also available
egr_send_request	“Send request”-button
egr_send_request .egr_footer_icon	Icon of “Send request”-button
egr_footer_cad_button .egr_footer_icon	Icon of “CAD Download” if enabled and in footer
egr_footer_price (3)	Price field below property list (label and value)
egr_footer_price_label (4)	Label in front of price (Standard “Total:”)
egr_footer_price_value	Value of price consisting of number and currency
egr_footer_price_discount_original (1)	Value of original price when discount is given via <a href="#">Price Update Webservice API</a>
egr_footer_price_discount_value (2)	Value of percentage discount given via <a href="#">Price Update Webservice API</a>



## 10.3 Examples

### 10.3.1 Change font

```
@font-face {
  font-family: 'name of font';
  font-style: normal;
  font-weight: normal;
  src: url("path/to/font");
}

body {
  font-family: "name of font", sans-serif;
}
```

### 10.3.2 Adapt cart button and hide number field in front of cart button

```
#egr_add_to_basket {
  background-color: black;
  color: white;
  border-radius: 0px;
}

#egr_add_to_basket_x {
  display: none;
}

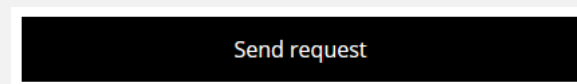
#egr_add_to_basket_number {
  display: none;
}
```



### 10.3.3 Adapt request button and hide its icon

```
#egr_send_request {
  background-color: black;
  color: white;
  border-radius: 0px;
  text-align: center;
}

#egr_send_request .egr_footer_icon {
  display: none;
}
```



### 10.3.4 Adapt price field

```
#egr_footer_price {  
  font-weight: bold  
}  
  
#egr_footer_price_label {  
  background: yellow  
}  
  
#egr_footer_price_value {  
  text-decoration-line: overline underline;  
  text-decoration-color: blue;  
}
```



Gesamt: EUR 181,00  
zzgl. MwSt

### 10.3.5 Adapt original price and discount field

```
#egr_footer_price_discount_original {  
  text-decoration: line-through  
}  
  
#egr_footer_price_discount_value {  
  background-color: #71c419;  
  border-radius: 5px;  
  padding: 2px;  
  color: black  
}
```



EUR 181,00 -25%  
Gesamt: EUR 135,75  
inkl. EUR 1,26 EUR Eco Abgabe, zzgl. MwSt

## 10.4 Style Components for Modular View

if modular view is activated (see [this section](#)) you can change the style of the following additional components.

Component	Description
egr_tabs button	parent element which involves all buttons
egr_tab_configuration (1)	id for the configuration tab
egr_tab_description (2)	id for the description tab
egr_tab_download (3)	id for the download tab
egr_tab_indicator (4)	indicator for active tab

**Note:** you have to use the !important rule to the style attribute.





# 11 Modify element labels

## 11.1 overview changeable elements

You are able to modify e.g. the following **element labels**:

## 11.2 how to change label - prerequisites

basically you need a folder with the wanted translation json-files<sup>7</sup> on your server which is reachable from outside without any login restrictions<sup>2</sup>). The path to this folder has to be send to easterngraphics (see [Contact/ Support](#)). Each language you want to change the labels for needs a json file with language short-cut as filename.

Example – you want to change the labels in English, German and Polish:

```

└─ example.com/pcon_translations/
   ├── en.json
   ├── de.json
   └── pl.json
    
```

<sup>7</sup> see ext. wiki link providing [json syntax](#)

## 11.3 supported languages

Following languages are currently supported:

language	shortcut
English	en
German	de
Czech	cs
Danish	da
Spanish	es
French	fr
Italian	it
Dutch	nl
Polish	pl
Portuguese	pt
Romanian	ro
Swedish	sv
Turkish	tr

## 11.4 default label names and json file structure

the corresponding json file has to contain the default label name and the wanted translation in json syntax like this:

```
"Open catalog": "back to catalog"
```

Labels and languages you don't list will be translated by default translation of our servers language files.

Following labels are supported for translation (case sensitive), you have always to use the English label names for your translation:

label	description
"Open catalog"	open catalog breadcrumb navigation label
"Configuration"	label for configuration tab (only in modular article view)
"Description"	label for description tab (only in modular article view)
"Download"	label for download tab (only in modular article view)
"Add to Cart"	Add to cart button (shop only)
"Send request"	send request button (Lead only)
"Download PDF"	Download pdf label for button if in footer or legend text on icon in 3D
"Download CAD-Model"	Download CAD Model label for button if in footer or legend text on icon in 3D
"Create snapshot"	Create Snapshot label for button if in footer or legend text on icon in 3D
"Total_Short form, do not include words like sum, price etc."	label for total before price
"excl. VAT"	excl. VAT label below the price (if net prices are activated)
"incl. VAT"	incl. VAT label below the price (if gross prices are activated)
"incl. VAT and {{value}} eco contribution"	incl. VAT label below the price (if Price Service API is used and eco is set)
"incl. {{value}} eco contribution, excl. VAT"	excl. VAT label below the price (if Price Service API is used and eco is set)

Example de.json for only this two labels, other labels will be translated with default german values:

```
{
  "Open catalog": "zurück",
  "Download PDF": "PDF herunterladen"
}
```

## 12 Contact / Support

### **In case of providing information / change settings in you instance**

If you want to provide us with new information regarding your pCon.ui instance settings please write an mail to [support@easterngraphics.com](mailto:support@easterngraphics.com).

### **In case of questions / issues / feature requests**

If you have any questions and need support please contact us via mail to [support@easterngraphics.com](mailto:support@easterngraphics.com).

**Note:** Our support team is working with so called SupportBudget. To order additional SupportBudget please contact your distribution partner at easterngraphics.

### **In case of Order new Services (additional cost)**


If you would like to add options that require additional cost please contact your distribution partner at easterngraphics

## 13 Notes on Data Protection (Cookies)

pCon.ui saves only information to the browsers local and session storage (no cookies) those are technically relevant.

Key Name	Storage	Purpose
i18nextLng	Local Storage	Is used to handle translations in the User Interface.
[GatekeeperID]	Session Storage	Used to identify the session on the server and use it across the page.

## 14 Footnotes

1)	Please make sure that your URL is using certificate-encrypted protocol (HTTPS). HTTP (unsecured) is <b>not</b> supported.
2)	<div style="display: flex; align-items: flex-start;">  <div> <p>CORS Header (Cross-Origin)</p> <p>Since pCon.ui will get information from a different origin than its own (prices from webshop) the same origin policy is violated. Therefore, CORS header need to be set in the webservers of the webshop to allow loading of those information in pCon.ui as follows</p> </div> </div> <div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px; background-color: #f9f9f9;"> <pre>Access-Control-Allow-Headers: Content-Type Access-Control-Allow-Method: POST Access-Control-Allow-Origin: https://ui.pcon-solutions.com</pre> </div>
3)	<p>Using AR share functionality on desktops with chromium based browsers:  the copy button of the AR Share function in the chromium bases browsers is not working unless you set the iframes access rights to access the clipboard and write to it.  The corresponding attribute to the iframe is <code>allow="clipboard-write"</code> - example:  <code>&lt;iframe allow="clipboard-write" src="***"&gt;</code></p>

### **Legal remarks**

© EasternGraphics GmbH | Albert-Einstein-Straße 1 | 98693 Ilmenau | GERMANY

This work (whether as text, file, book or in other form) is copyright. All rights are reserved by EasternGraphics GmbH. Translation, reproduction or distribution of the whole or parts thereof is permitted only with the prior agreement in writing of EasternGraphics GmbH.

EasternGraphics GmbH accepts no liability for the completeness, freedom from errors, topicality or continuity of this work or for its suitability to the intended purposes of the user. All liability except in the case of malicious intent, gross negligence or harm to life and limb is excluded.

All names or descriptions contained in this work may be the trademarks of the relevant copyright owner and as such legally protected. The fact that such trademarks appear in this work entitles no-one to assume that they are for the free use of all and sundry.