PriceLabel 10

Integration Guide

Rev-20200705

1 Contents

2 API

Integration with other systems (programs, web sites) through the API allows you to print labels, open a template editor, get a list of templates, etc. directly from these programs and sites.

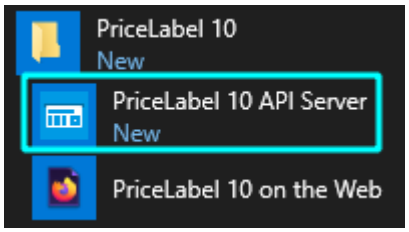
The program supports 3 API options:

1. REST API - commands and data are transmitted via HTTP
2. Clipboard API - commands and data are transmitted through the clipboard
3. Command line API - commands and data are transmitted in the program launch line

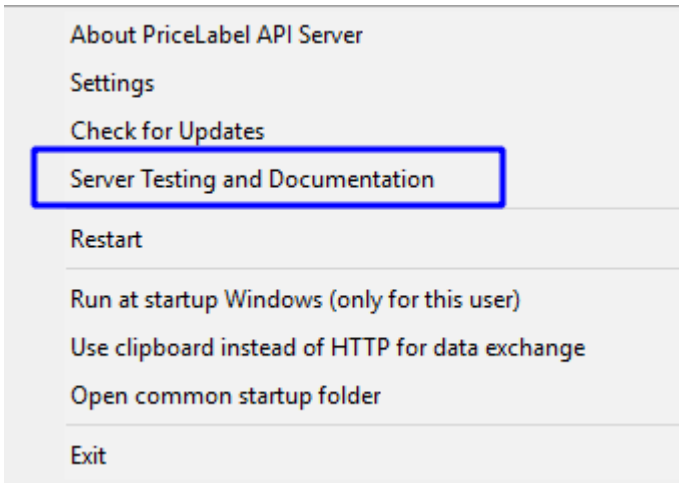
2.1 REST API

You can familiarize yourself with the REST API at the link https://pricelabel.app/docs/api

To work with this API, the program starts in the server API mode, for this the parameter in the "-apiserver" command line is used. Also, to run the program in server API mode, you can use the Windows Start menu:



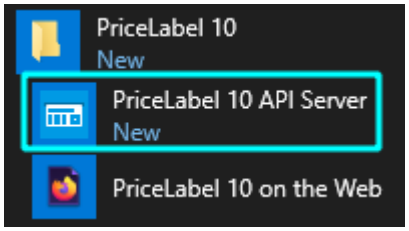
You can try the API in action by going to the local documentation page via the server API context menu in the tray:



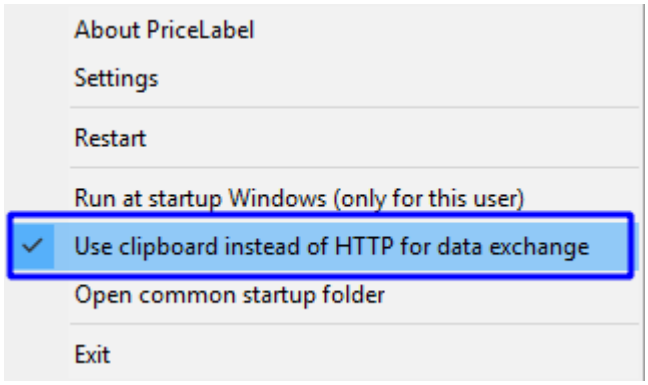
2.2 Clipboard API

This API allows you to configure interaction with third-party programs that work on a remote server through RDP and there is no way to install PriceLabel on a remote server. In this case, the data is transmitted through a clipboard shared between the local computer and the remote server.

To work with this API, the program starts in the server API mode, for this the parameter in the "-apiserver" command line is used. Also, to run the program in server API mode, you can use the Windows Start menu:



After the first launch, you must set the flag through the context menu of the program in the tray:



(this corresponds to the line "HTTPServer_UseClipboardInsteadHTTPForDataExchange = 1" in the config.ini program settings file).

In this API, the application receives requests through the clipboard and executes commands or sends data to the clipboard. To do this, the server with a frequency of 1 second views the contents of the clipboard. If the clipboard contains text data and this data is a request to the server, then the server executes the commands and places the response on the clipboard.
To transfer information to the program, text data of a special format is used in the form of a text block in the clipboard.

2.2.1 Request to server

To request the server to the clipboard you need to put the data, which also contains the program control parameters. The data format is described in the section Data format for transfer to the program..

2.2.2 Server response

The program places on the clipboard a response in which the first line is:

- "PriceLabelAnswer::OK" - in case of no error. Subsequent lines are data from the program, if the request meant receiving data from the program.
- "PriceLabelAnswer::ERROR" - in case of error. Subsequent lines are a description of the error.

2.2.3 Server connection test

To test the connection to the server, you need to put the string "PriceLabelTest" on the clipboard. In response, the server should return "PriceLabelAnswer :: OK".

2.3 Command line API

2.3.1 Interaction with a single file

-allinonefile <Path to the data file> -encoding <File encoding>

<Path to the data file>: the full path to the file with input data, which also contains the control parameters of the program.
The data format is described in the section Data format for transfer to the program..

<File Encoding>:

- **utf-8** (file with data in UTF-8 encoding)
- **utf-16** (file with data in UTF-16 encoding)
- **the number of the code page identifier** (https://msdn.microsoft.com/en-us/library/dd317756%28v=VS.85%29.aspx), for example 1252 (ANSI Latin 1; Western European (Windows)). Optional parameter. If not specified, the default encoding in the system will be used.

2.3.2 Other interaction options

2.3.2.1 Print, preview, save labels to a file (PDF, Excel), open print settings, open the template editor

<Action> <Parameter 1> <Parameter 2> ... <Parameter N>

Action:

-print
printing of labels.

-preview
preview of the labels.

-printsettings
opens the print settings.

-openinprog
opens a list of products in the program in the *Printable Data* for further action.

-templateseditor
opens the template editor.

-apiserver
launch API server.

Parameters:

-file <Path to the data file>
<Path to the data file> - the full path to the file with input data.
The data format is described in the section Data format for transfer to the program.

-extdataid <ID>
<ID> - source ID from the catalog *External data sources.*

-template <Name of the primary label template>
<Name of the primary label template> is the name of the template from the template editor. Specifies the primary label template. This template will be used for all products, except those that have a different template.

-templateid <ID of the primary label template>
<ID of the primary label template> - template ID from the template editor. Specifies the primary label template. This template will be used for all products, except those that have a different template.

-format <Data format in file>
<Data format in file>: **txt** - plain text, **json** - in JSON format. Optional parameter. If not specified, the format of the data file is considered to be a plain text file.

-encoding <File encoding>
<File encoding>: **utf-8** (file with data in UTF-8 encoding), **utf-16** (file with data in UTF-16 encoding) or **code-page identifier number** (https://msdn.microsoft.com/en-us/library/dd317756%28v=VS.85%29.aspx), for example 1252 (ANSI Latin 1; Western European (Windows)). Optional parameter. If not specified, the default encoding in the system will be used.

-printername <Printer Name>
<Printer Name> - the name of the printer in the system or **printtopdf** for saving to a PDF file, **printtoexcel** for saving to an Excel file, **printtoimages** to save labels as PNG images. If the printer is not specified or the printer is specified, but there is no such printer in the system, then will be used the printer is bound to the template and the current user, and if there is no such binding, the default printer will be used. The printer is bound to the template and the current user in the print settings.

-username <Username>
<Username> is the user name in the *Users* catalog. Optional parameter. If the password is set to

enter the program, if it is not specified or specified, but there is no such in the program, the login window will be displayed.

-userpass <User password>
<User password> is the user's password in the Users catalog. Optional parameter. If not specified, but the user's password is set, the login window will be displayed.

Example:

C:\Program Files (x86)\PriceLabel 10\PriceLabel.exe **-preview -file** "C:\products.json" **-template** "Simple" **-format** json **-encoding** utf-8 **-printername** "Office printer 1"

The command will open a preview form, which will display labels according to the "Simple" template with the data from the file "C:\products.json".

2.3.2.2 Getting a list of label templates

-gettemplatesnames <Parameter 1> <Parameter 2> ... <Parameter N>

Parameters:

-file <The path to the file where the template list will be uploaded>
<The path to the file where the template list will be uploaded> - the full path to the file to which the list of label templates will be uploaded.

-encoding <File encoding>
<File encoding>: **utf-8** (file with data in UTF-8 encoding), **utf-16** (file with data in UTF-16 encoding) or **code-page identifier number** (https://msdn.microsoft.com/en-us/library/dd317756%28v=VS.85%29.aspx), for example 1252 (ANSI Latin 1; Western European (Windows)). Optional parameter. If not specified, the default encoding in the system will be used.

-favorites
upload only "favorites". Optional parameter.

-detailed
upload detailed information. Optional parameter.

The format of the downloaded file with a list of label templates is described in the section Format of data received from the program.

2.3.2.3 Opening the User Guide

-help

2.4 Data format for transfer to the program

Used in the *Clipboard API* and *Command line API*.

The format is used in several variants:

- **Ready-made data:** products, their properties and other parameters. Such data the program can directly use for printing or when importing data. Also can contain the parameters of the program management.
- **Data to connect to the data source:** a description of the external data source. Having received such data, the program will connect to the external data source described in this data, will receive the products, their properties and other parameters, which will then be used for printing or when importing data. Also can contain the parameters of the program management.
- **Data for program management:** contain only program management parameters and are used for various control options, where no ready-made data or description of an external data source is required. For example, printing using an external data source described in the *External Data Sources* catalog.

The JSON format is used to represent the data.

The name of the format used in the program: *PriceLabel JSON*.

2.4.1 Ready-made data

2.4.1.1 Data structure

The name of the key in the JSON steam structure (key: value)	Value type	Value	Compulsory	Note
PriceLabelData	String	Empty String	Yes	A label indicating that the data is for PriceLabel.
version	String	3	Yes	The version of the format.
runparameters	Object	The list of pairs key:value, each of which is the program control parameter and its value.	No	Required only when the program is started with a parameter in the command line **allinonefile** or when requesting the HTTP protocol with the command **allinonefile**
items	Array of objects	A list of objects, each of which is a product and its parameters as a list of pairs key:value.	Yes	Ignored if the program control parameter is specified, which does not require ready-made data about the products, for example **extdataid**

2.4.1.2 Value Formats

Strings and numbers are passed as is.

Strings can use special characters:

\b	Backspace (ascii code 08)
\f	Form feed (ascii code 0C)
\n	New line
\r	Carriage return
\t	Tab
\"	Double quote
\\	Backslash character

The dates and times are transmitted in the format described in the Data Types in the formulas.

2.4.1.3 Products parameters

The values of parameters of products can be of the following types:

- String. For example:
  - "ProductName":"Battery"
- Number. For example:
  - "ProductPrice":123.45
- Boolean. For example:
  - "IsFavorite":false
- The object (only the *ProductImage* parameter). For example:
  - "ProductImage":{"datatype":"file", "data":"http://products.com/image.jpg"}
  - "ProductImage":{"datatype":"file", "data":"C:\image.jpg"}

Parameter names can be used both standard and own:

- The standard product parameters are described in the section Parameters and variables in formulas. Some standard parameters of products are used only as part of arrays. Such parameters are described below. The *ProductImage* parameter is described in detail below.
- You can use any custom parameter names in cases:
  - When adding new columns to the product catalog.
    In this case, new parameters of the products appear corresponding to these columns.
  - When printing using external data.
    In this case, the names of the parameters in the template must coincide with the names of the parameters of the products in the transmitted data. Also in this case, the parameter values can only be of a string type, but in the future they can be converted to any desired data type.

2.4.1.4 Parameter *ProductImage* (product image)

The value of the parameter is an object consisting of two pairs key:value:

- **datatype**: indicates the method of obtaining the image. May be:
  - **file**: to get an image from a file or by URL.
  - **base64**: Obtaining an image from the encoded Base64 string.
- **data**: source of the image (file path, URL, coded Base64 string).

Example:

"ProductImage":{"datatype":"file", "data":"http://products.com/image.jpg"}

2.4.1.5 Parameters of products that are used only as part of arrays

Barcode, Barcode Type

Used only as part of the *Barcodes* array (Barcodes).

Example:

"Barcodes":[{"Barcode":"5607017160496", "BarcodeType":0,"BarcodeIsMain":false},
{"Barcode":"4607017160497", "BarcodeType":3,"BarcodeIsMain":true},
{"Barcode":"6607017160495", "BarcodeType":0,"BarcodeIsMain":false}]

In the example, 3 product barcodes with the indication of their types were transferred to the program, the main barcode is also indicated.

SerialNumber

It is used only as part of the *SerialNumbers* array (Serial numbers).

Example:

"SerialNumbers":[{"SerialNumber":"SN123456337","SerialNumber IsMain":true},
{"SerialNumber":"SN223456333","SerialNumber IsMain":false}]

In the example, 2 product serial numbers were transferred to the program, the main serial number is also indicated.

Group (Name of the product folder)

Used only as part of the *Groups* array (path to the product folder in the catalog - the list of folders from the product catalog root to the product folder).

Example:

Groups":[{"Group":"Expendables"},
{"Group":"Batteries"},
{"Group":"USA"}]

The example shows the path to the product folder (the list of folders from the product catalog root to the product folder): "Expendables" \ "Batteries" \ "USA".

2.4.1.6 Features of some parameters

CurrentDate

If the value of the parameter is empty, the program will use the current date and time instead.

2.4.1.7 Data Examples

Data for two products.
Parameters of the program management are specified:

- **Open a preview**
- **Use template with ID 9**

```
{"PriceLabelData":"",
"version":"3",
"source":"",
"runparameters": {"preview":"", "templateid":9},
"items":[
{"ProductLabelCount":3,
"Groups":[{"Group":"Expendables"},{"Group":"Batteries"}],
"ProductName":"Panasonic LR03/4 (AAA) Alkaline 1,5V",
"Barcodes":[{"Barcode":"5607017160496"},{"Barcode":"4607017160497"}],
"ProductPrice":0.5,
"UnitName":"ea",
"ProductCode":"00004"},
{"ProductLabelCount":1,
"Groups":[{"Group":"Expendables"},{"Group":"Batteries"},{"Group":"USA"}],
"ProductName":"Energizer AAA Max Alkaline E92",
"Barcodes":[{"Barcode":"5607017160496"},{"Barcode":"4607017160497"}],
"ProductPrice":0.59,
"ProductPrice2":0.6,
"ProductPriceOld":0.65,
"UnitName":"ea",
"ProductCode":"00027"}
]
}
```

2.4.2 Data to connect to the data source

2.4.2.1 Data structure

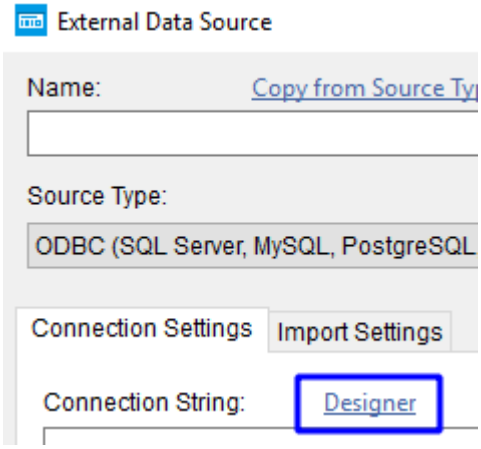
The name of the key in the JSON steam structure (key: value)	Value type	Value	Compulsory	Note
PriceLabelData	String	Empty String	Yes	A label indicating that the data is for PriceLabel.
version	String	3	Yes	The version of the format.
runparameters	Object	The list of pairs key:value, each of which is a program control parameter and its value.	No	Required only when the program is started with a parameter in the command line **allinonefile** or when requesting the HTTP protocol with the command

				allinonefile
source	String	Data Source Type: • **SQLite**: for direct connection to the SQLite database. • **ODBC**: to connect to any data source, for example, Access, Excel, MySQL, etc.	Yes	
connection_string	String	Connection string to the data source: • If the data source is SQLite, then this is the path to the database. For example, "C:\ products.sqlite". • If the ODBC data source is, then this is the connection string to the source. For example, "Provider=Microsoft.Jet .OLEDB.4.0;Data Source=C:\\ products .mdb;Persist Security Info= False".	Yes	
sql_string	String	SQL query to the data source.	Yes	

The connection string to the ODBC data source can be built directly into the program:

- Open the catalog *External Data Sources* (*Menu* > *Catalogs* > *External Data Sources*)
- Press the button **Add**
- Select Source Type *ODBC*

- Click **Designer** in the connection string:



In the query when listing fields, you need to specify their relationship with the parameters of the products. Communication is established by a structure of the form: **Request field** as [**Item parameter**]

For example, "Name as [ProductName]", where the Name field is associated with the ProductName parameter in the program, and, accordingly, the program receives information that the name of the product should be taken from this field.

Example:

"SELECT Name as [ProductName], SKU as [ProductSKU], Image as [ProductImage] FROM products", the names, SKUs and images of products that are taken from the "products" table are transferred here.

2.4.2.2 Products parameters

The values of parameters of products can be of the following types:

- various types of String
- various types of Integer
- various types of Float
- Boolean
- various types of BLOB (ready images)
- various types of Date

Parameter names can be used both standard and own:

- The standard product parameters are described in the section Parameters and variables in formulas. The *ProductImage* parameter is described in detail below.
- You can use any custom parameter names if the data is used to print labels, and not to import data into the program.

2.4.2.3 Parameter *ProductImage* (product image)

If the type of the field value in the query is one of the variants of BLOB, then the image is taken from the query directly.

If the type of the field value in the query is one of the variants of String, the image is obtained from the file or via the http request, if the field value is the file path or URL, respectively.

2.4.2.4 Create a connection to an external data source and export as a file

Add a new data source to the *External Data Sources* catalog and, without writing it, click the **Export to file** button.

2.4.2.5 Data Examples

Obtaining data from the SQLite database.

Parameters of the program management are specified:

- **Open a preview**
- **Use template with ID 9**

```
{"PriceLabelData":"",
"version":"3",
"source":"SQLite",
"runparameters": {"preview":"", "templateid":9},
"connection_string":"c:\\products.sqlite",
"sql_string":"SELECT Name as [ProductName], SKU as [ProductSKU], Image as [ProductImage] FROM products"
}
```

Obtaining data from the Access database via ODBC.
Parameters of the program management are specified:

- **Open a preview**
- **Use template with ID 9**

```
{"PriceLabelData":"",
"version":"3",
"source":"ODBC",
"runparameters": {"preview":"", "templateid":9},
"connection_string":"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\\products.mdb;Persist Security Info= False",
"sql_string":"SELECT Name as [ProductName], SKU as [ProductSKU], Image as [ProductImage] FROM products"
}
```

2.4.3 Data for program management

Works only when the program is started with a parameter in the command line of **allinonefile** or when it is requested by the HTTP protocol with the command **allinonefile**

2.4.3.1 Data structure

The name of the key in the JSON steam structure (key: value)	Value type	Value	Comp ulsory	Note

PriceLabelData	String	Empty String	Yes	A label indicating that the data is for PriceLabel.
version	String	3	Yes	The version of the format.
runparameters	Object	The list of pairs key:value, each of which is the program control parameter and its value.	Yes	

2.4.3.2 Data Examples

Parameters of the program management are specified:

- **Open a preview**
- **Use template with ID 9**
- **The data is retrieved from an external source whose parameters are described in the *External Data Sources* catalog entry with ID 1**

```
{"PriceLabelData":"",
"version":"3",
"runparameters": {"preview":"", "templateid":9, "extdataid":1}
}
```

Parameters of the program management are specified:

- **Open template designer**
- **Use template with ID 9**

```
{"PriceLabelData":"",
"version":"3",
"runparameters": {"templateseditor":"", "templateid":9}
}
```

2.4.4 Control parameters of the program

Parameter name	Value type	Value	Description
print	String	Empty String	Print labels.
preview	String	Empty String	Open the preview.
printsettings	String	Empty String	Open print settings.
openinprog	String	Empty String	Run the program and fill in the printable data.
templateseditor	String	Empty String	Open template designer.
extdataid	Number	ID of the data source from the *External Data Sources* catalog	Use data from an external data source described in the *External Data Sources*

			catalog
template	String	Template name in *template designer*	Use the template with the specified name as the primary. This template will be used for all products, except those that have a different template.
templateid	Number	Template ID in *template designer*	Use the template with the specified ID as the primary. This template will be used for all products, except those that have a different template.
printername	String	• The name of the printer in the system • **printtopdf** for saving to PDF file • **printtoexcel** for saving to Excel file • **printtoimages** to save labels as PNG images.	Use the printer with the specified name. If the printer is not specified or the printer is specified, but there is no such printer in the system, then will be used the printer is bound to the template and the current user, and if there is no such binding, the default printer will be used. The printer is bound to the template and the current user in the print settings
printtofile	String	Full path to the file when saving to PDF, Excel or folder path when saving as PNG images.	If the path to the file or folder path is not specified, the save to file / select folder dialog will be displayed. Optional parameter.
showprintprogress	Boolean	Show print progress	By default, print progress is always shown, except for the HTTP server mode.
username	String	User name in the *Users* catalog	Optional parameter. If the password is set to enter the program, if the user name is not specified or specified, but this is not in the program, the login window will be displayed.
userpass	String	User password in the *Users* catalog	Optional parameter. If the password is set to enter the program, if the user's password is not specified or specified, but there is no such password in the program, the login window will be displayed.

gettemplatesnames	String	Empty String	Return a list of label templates. The format of the template list is described below.
encoding	String	• **utf-8** • **utf-16** • **code-page-identifier number** (https://msdn.microsoft.com/en-us/library/dd317756%28v=VS.85%29.aspx), for example 1252 (ANSI Latin 1; Western European (Windows)).	Encoding of data received from the program
favorites	String	Empty String	A helper parameter for **gettemplatesnames**. Return only templates marked as *Favorite*. Optional parameter.
detailed	String	Empty String	A helper parameter for **gettemplatesnames**. Return a detailed list of templates, instead of a simple one. Optional parameter.
getlastprintdatetime	String	Empty String	Return the date of the last print
help	String	Empty String	Open User Guide
operations_with_catalog Command parameters:	Object		Operations with the product catalog (import, delete, read)
type	String	• **import** - products import	Operation type
use_settings_from_extdata_id	Number	ID of the data source from the *External Data Sources* catalog	External data source in the *External data sources* catalog. When importing, the only settings from the *Settings for Import to Product Catalog* tab of this source will be used. Products must be transferred as *Ready-made data*

2.4.5 The format of the data received from the program

Command **operations_with_catalog**

Answer as a JSON object:

{"Products":{"addedCount":N1,"updatedCount":N2,"skippedCount":N3}}

, where "Products" is an indication of the product catalog, N1-N3 is the number of added, updated and skipped products, respectively.

Command **gettemplatesnames**

If the **detailed** parameter is not specified, then the format of the received data:

Label template name 1
Label template name 2
Label template name 3
...
Label template name N

If **detailed** is specified, then the format of the received data:

Data row 1
Data row 2
Data row 3
...
Data row N

Each data row is a sequence of values separated by a tab character (code 9):

Value number in order	Description of value	Note
1	Name	The name of the template in *Template designer*
2	ID	Template ID in *Template designer*
3	Description	Description of the template in *Template designer*
4	ID of the parent folder	Folder ID in *Template designer*
5	Folder flag (0 - template, 1 - folder)	