PriceLabel 10

Integration Guide

Rev-20200923

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.

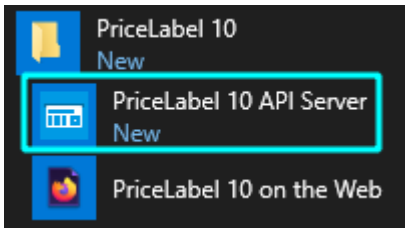
The program supports 3 API options:

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

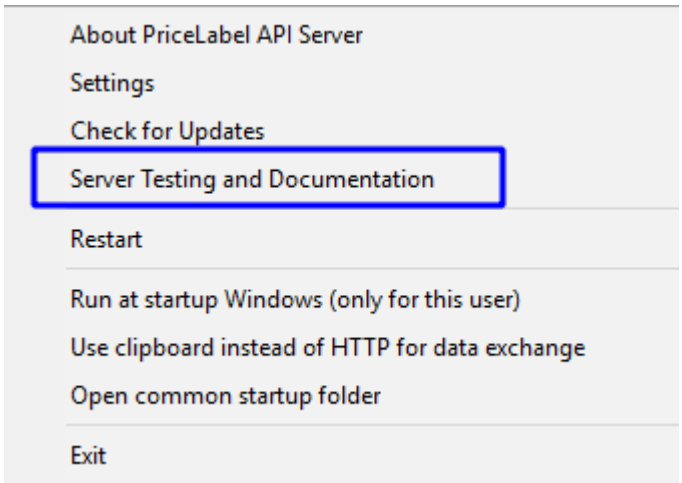
2.1 REST API

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:



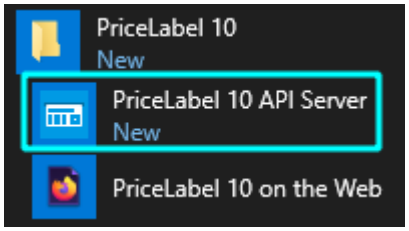
You can try the API in action by going to the local documentation page via the server API context menu in the tray:



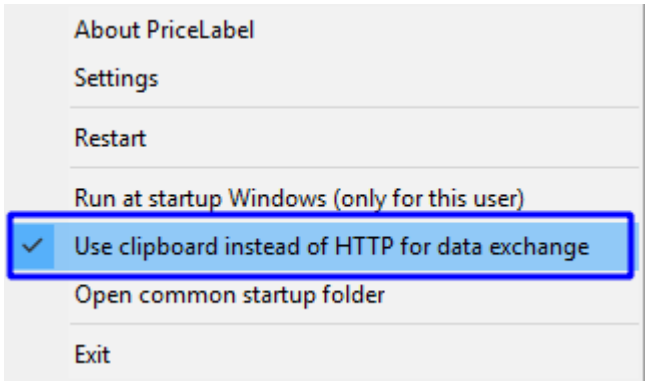
2.2 Clipboard API

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, the following lines should be placed on the clipboard:

1. "PriceLabelClipboardAPIRequest"
2. Path along with parameters as in the REST API documentation. The path and parameters are combined according to the URI composing rule. For example, "/templates?encoding=utf-8&format=text"
3. Request body as in the REST API documentation (optional string)

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

Sample request to get a list of templates:

PriceLabelClipboardAPIRequest
/templates?encoding=utf-8&format=text

2.2.2 Server response

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

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

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 data.
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.

-performtask
task execution.

Parameters:

-file <Path to the data file>
<Path to the data file> - the full path to the file with 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.

-taskid <Task ID>

<Task ID> - task ID from the *Tasks* catalog.

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 *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.

The JSON format is used to represent the data.

2.4.1 Ready-made data

The data is in JSON format as it is described in the "Request body" sections of the REST API documentation.

There are 2 JSON objects to be appended to this data:

"version":"5","request":"<Path>"

, where <Path> is the path as in the REST API documentation. For example, "/templates".

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

2.4.1.1 Data Examples

Opening the Label Designer

(Use template with ID 9)

```
{"version":"5",
"request":"/windows/designer",
"templateid":9,
}
```

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
source	String	Data Source Type: • **SQLite**: for direct connection to the	Yes	

		SQLite database. • **ODBC**: to connect to any data source, for example, Access, Excel, MySQL, etc.		
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	

There are 2 JSON objects to be appended to this data:

"version":"5","request":"<Path>"

, where <Path> is the path as in the REST API documentation. For example, "/labels/print".

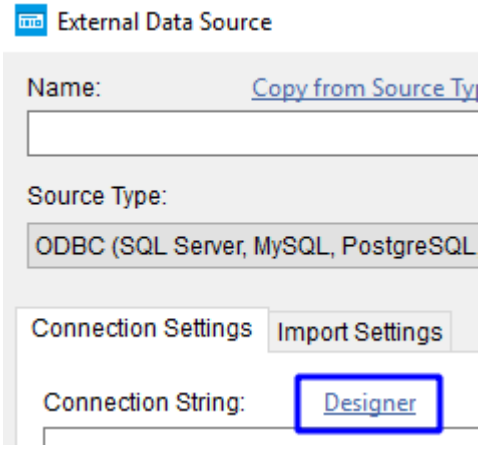
You can familiarize yourself with the REST API at the link https://pricelabel.app/docs/api

How to get the connection string

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 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.

2.4.2.3 Data Examples

Obtaining data from the SQLite database

(Open a preview, Use template with ID 9)

```
{"version":"5",
"request":"/labels/preview",
"templateid":9,
"source":"SQLite",
"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

(Open a preview, Use template with ID 9)

```
{"version":"5",
"request":"/labels/preview",
"templateid":9,
"source":"ODBC",
"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 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)	