

---

# Barcode Buddy Documentation

*Release 1.5.0.0*

Aug 15, 2021



---

## Contents

---

<b>1</b>	<b>Why use Barcode Buddy and how does it work</b>	<b>3</b>
<b>2</b>	<b>Contents</b>	<b>5</b>
2.1	Setup . . . . .	5
2.2	Usage . . . . .	10
2.3	Updating Barcode Buddy . . . . .	16
2.4	Advanced usage and configuration . . . . .	16
2.5	Contributions . . . . .	20
2.6	Changelog . . . . .	20



Barcode Buddy for Grocy is an extension for [Grocy](#), allowing to pass barcodes to Grocy. It supports barcodes for products and chores. If you own a physical barcode scanner, it can be integrated, so that all barcodes scanned are automatically pushed to BarcodeBuddy/Grocy.



---

## Why use Barcode Buddy and how does it work

---

Barcode Buddy makes using a barcode scanner a lot easier - ideally the barcode scanner is connected to a server / computer so it can send the barcodes to the program. Alternatively the barcodes can be manually added with the web UI. In contrast to Grocy, unknown barcodes are looked up with OpenFoodFacts.org first, in order to find out the product name / category. If found, Barcode Buddy checks if the user has stored any tags associated with the name: For example, if the name is “Chocolate bar”, it can automatically be set to the Grocy product “Chocolate”.

A known barcode will be processed, so it reduces / increases the inventory or manipulates the shopping list.

If the product could not be looked up, the user can select it manually.

Chores can also be executed with barcodes.





## 2.1 Setup

There are two different ways to setup Barcode Buddy: Either a bare metal approach or docker

### 2.1.1 Docker

#### Requirements

- A host running docker

#### Installation

To download, run the following command, and replace YOURTAG with one from the list below:

```
docker pull f0rc3/barcodebuddy-docker:YOURTAG
```

Tag	Architecture	Version
latest	x86-64	stable
arm32v7-latest	armhf	stable
arm64v8-latest	arm64	stable
latest-dev	x86-64	unstable
arm32v7-latest-dev	armhf	unstable
arm64v8-latest-dev	arm64	unstable

Most of the time, you will need the *latest* tag. If you are running docker on a Raspberry Pi, you will need the *arm32v7-latest* tag.

*Stable* indicates, that you are using the latest release which should work without any bugs. *Unstable* is the latest developer version, which might include more features, but could also contain bugs.

If you don't want to download the prebuilt image, you can find the Dockerfile on the [Github project page](#).

### Starting the Container

To start the container, run the following command:

```
docker run -d -v bbconfig:/config -p 80:80 -p 443:443 f0rc3/barcodebuddy-  
↪docker:YOURTAG
```

You can now open `http://DOCKER_HOST_IP/` to set up BarcodeBuddy. If you are already serving a webserver on your Docker host, you need to change the ports, eg.:

```
docker run -d -v bbconfig:/config -p 8080:80 -p 9443:443 f0rc3/barcodebuddy-  
↪docker:latest
```

The following arguments can also be passed:

Argument	Value	Effect
TZ	string	Set the container timezone (eg. "Europe/Berlin")
ATTACH_BARCODESCANNER	true/false	Use attached barcode scanner
IGNORE_SSL_CA	true/false	Accept self-signed SSL certificates
IGNORE_SSL_HOST	true/false	Accept SSL certificates where the host does not match
PUID	0-65536	Sets user ID for container
PGID	0-65536	Sets group ID for container

For more information on how to attach a barcode scanner, see [Using a physical barcode scanner](#)

To pass an argument, use the `-e` function, eg:

```
docker run -d -v bbconfig:/config -e ATTACH_BARCODESCANNER=true -p 80:80 -p 443:443  
↪f0rc3/barcodebuddy-docker:latest
```

If you want to use a different internal URL for connecting to Grocy, enter the internal URL in during the setup. You can then set an external URL for links on the website with the argument

```
-e BBUDDY_EXTERNAL_GROCY_URL=your.url
```

## 2.1.2 Bare Metal

### Requirements

- webserver (eg. NGINX, Apache)
- curl
- Access to the command line
- PHP 7.2 or higher (PHP 8.0 **not** supported at the moment)
- The following PHP modules:
  - curl
  - date
  - json

- redis
- sqlite3
- sockets
- When using a barcode scanner:
  - Root access!
  - sudo
  - screen
  - evtest

## Installation

*Stable* indicates, that you are using the latest release which should work without any bugs. *Unstable* is the latest developer version, which might include more features, but could also contain bugs.

It is strongly recommended to change `pm.max_children` to a value of 10 or higher in `/etc/php7/php-fpm.d/www.conf` (path might be different, depending on PHP version and distribution; for Ubuntu 18.04 it is `/etc/php/7.2/fpm/pool.d/www.conf`).

## Webserver setup

This guide is written for a Debian based server, including Ubuntu. If you already have a webserver setup, please make sure to have a look at the [Nginx example file](#), as for the folder `/api/` a rewrite rule has to be added.

## Installing NGINX

- Get root, ideally with `sudo -i`
- Install nginx: `apt-get install nginx`
- If you are running a server with ufw active, run `ufw allow 'Nginx Full'`
- Install all php modules and other requirements `apt-get install php php-fpm php-curl php-date php-json php-sqlite3 php-redis redis redis-server sudo screen evtest`
- If some of the packages are not found, try again including the php version, eg. `apt-get install php7.2-curl ...` (see next point)
- Check what PHP version you are using with `php --version` (eg. “7.2”).
- Copy the [Nginx example file](#) to `/etc/nginx/sites-enabled/`
- Adjust the new file:
  - If you are not using PHP7.2, change the line `fastcgi_pass unix:/var/run/php/php7.2-fpm.sock;` to your PHP version
  - If you are not installing Barcode Buddy to `/var/www/html/barcodebuddy/`, change the line `root /var/www/html/barcodebuddy/;` to your directory
- Follow the steps below to download either the stable or unstable version
- Execute the command `chown www-data:www-data -R /path/to/the/barcodebuddy/folder` for the folder that you just created

- Change `pm.max_children` to a value of 10 in `/etc/php/7.2/fpm/pool.d/www.conf` (adjust path for your PHP version)
- Restart NGINX service `nginx restart`

### Configuring Apache2

We recommend using Nginx. If you are already an Apache2 user, follow these steps to make sure that Barcode Buddy is working correctly:

- Execute `a2enmod rewrite` to make sure that the rewrite module is active
- Make sure that you can use `.htaccess` files for rewriting. For that the option `AllowOverride` for the directory must be set to `All`. You can normally find this configuration in the `apache2.conf` file. For Ubuntu this file is located at `/etc/apache2/apache2.conf`. Search for `AllowOverride` and set it to `All` for the root directory where Barcode Buddy is installed.

Example:

```
[...]
<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>
[...]
```

### Stable version

Download the project and copy all files into your webserver.

### Unstable version

Execute

```
git clone https://github.com/Forceu/barcodebuddy.git .
```

in the folder where you want to install Barcode Buddy to.

### Starting the websocket service

If you have access to your webserver's command line, make sure to start the websocket server. This way you can use the Screen module and if there are any changes, Barcode Buddy will automatically refresh.

Navigate to your installation folder and execute `php wsserver.php` to start the server. To have it run in the background, either use the screen application (recommended)

```
screen -S bbuddyserver -d -m /usr/bin/php /path/to/the/barcodebuddy/folder/wsserver.
↪php
```

or the following command:

```
nohup php wsserver.php &
```

To start the websocket server after a reboot, you can use cron. Make sure to use the crontab for the webserver user (on Debian/Ubuntu this the user `www-data`).

Open the crontab for the user:

```
sudo crontab -e -u www-data
```

And insert the following new line (you might need to adjust the paths):

```
@reboot /usr/bin/screen -S wsserver -d -m /usr/bin/php /var/www/html/barcodebuddy/  
↪wsserver.php
```

### 2.1.3 VirtualBox

We have also released a [VirtualBox](#) image, which automatically downloads the latest docker image and runs it.

#### Installation

Open VirtualBox, and go to File/Host Network Manager. If there is no network listed yet, click on “Create” and make sure that the box for DHCP Server is ticked. [Download the image](#) and open it with VirtualBox, then click on “Import” in the new window.

Start the image - once it is completely running, you will see a login prompt. Above that, you will see two IP addresses. Normally with the second one you can reach the server, so simply connect in your webbrowser to `http://THE_IP/`.

If you need to log in to the image, the default username is `root` and the default password is `barcode`. For security reasons, SSH is disabled, to enable it, execute `rc-update add sshd` (make sure to change your password and to add a non-root user!)

### 2.1.4 Reverse Proxy

If you would like to run Barcode Buddy behind a reverse proxy, you can find an [Nginx configuration](#) in the example folder.

Make sure that you set `fastcgi_pass_header "X-Accel-Buffering"` in the Barcode Buddy host Nginx configuration, or `proxy_buffering off` in the reverse proxy configuration. If buffering is enabled for your reverse proxy, Server-Sent Events (SSE) might not be available and would break the Screen module.

### 2.1.5 Hass.IO

#### Connecting to Grocy

If you are running Grocy in a HASS.io container, further configuration is needed. Open HASS and go to the Grocy plugin section (not Grocy itself). Scroll down and enter 9192 in the Network section and press save. Make sure that you disable SSL in the Grocy config section above, if you are not using a proper certificate. Then restart Grocy. You will now be able to access Grocy under the URL `http://hassio.local:9192`. In Barcode Buddy setup, enter `http://hassio.local:9192/api/` as URL.

## 2.2 Usage

### 2.2.1 First Start

Open your Grocy website. Click on Settings in the top right corner and then click on *Manage API keys*. Click on *add* and copy the long API key to your clipboard. You can now open Barcode Buddy by visiting your webserver's URL. The setup will first let you create a user and then ask you to enter the API details. Make sure to have the trailing *"/api/"* for your Grocy URL at the end.

### 2.2.2 Using The Web UI

#### Overview

When you open the web ui, you will see three cards:

- New Barcodes: Barcodes that are unknown to Grocy, but the name could be looked up
- Unknown Barcodes: Barcodes that are unknown to Grocy and could not be looked up
- Processed Barcodes: A history of all barcodes that were processed by Barcode Buddy

If you have entered a barcode that is linked to a Grocy product with the tare feature enabled, a fourth section will pop-up named "Actions required", where you can enter the weight for the product.

#### Special Barcodes

There are seven special barcodes - if you scan the barcode, Barcode Buddy goes into a different mode: Eg. in Purchase mode all barcodes that are scanned will be added to Grocy's inventory.

Mode	Default Barcode	Explanation
Consume	BBUDDYC	All items scanned in this mode will be removed from the inventory
Consume (spoiled)	BBUDDYCS	All items scanned in this mode will be removed from the inventory and marked as spoiled
Consume all	BBUDDYCA	All products scanned in this mode will have all stock removed from their inventory
Purchase	BBUDDYP	All items scanned in this mode will be added to the inventory
Open	BBUDDYO	All items scanned in this mode will be marked as opened
Inventory	BBUDDYI	Displays the current amount of items in stock of that product
Add to shoppinglist	BBUDDYAS	All items scanned in this mode will be added to the default shopping list
Quantity	BBUDDYQ-?	Sets a quantity for a barcode. Replace ? with number of items. Eg. if you have a carton with 10 eggs, scan the barcode BBUDDY-Q-10 and then the barcode of the carton. The next time you scan the carton barcode, it will register as 10 units.

You can find printable copies of the barcodes in the [example folder](#) .

## Using the UI

Once a barcode was added and is not recognised by Grocy, it will be added to the list in the web ui. Simply select the Grocy product from the drop-down list. If the name could be looked up, you will also see checkboxes for each word that was used in the name. If you want the selected Grocy product preselected for a different barcode that includes such a word, tick it and then press either “Consume” or “Add”. The barcode will then saved to Grocy and the next time you scan it, the product will automatically be processed.

Barcode Buddy

GrocyScreen

### New Barcodes

Name	Barcode	Quantity	Product	Action	Tags	Create	Remove
Coca Cola	54490000	1	Coke	ADD CONSUME	<input type="checkbox"/> Coca <input checked="" type="checkbox"/> Cola	CREATE PRODUCT	REMOVE
Greek Yoghurt	27030868	1	Yoghurt	ADD CONSUME	<input type="checkbox"/> Greek	CREATE PRODUCT	REMOVE
Sella basmati Rice	890800668362	1	Rice	ADD CONSUME	<input type="checkbox"/> Sella <input type="checkbox"/> basmati <input checked="" type="checkbox"/> Rice	CREATE PRODUCT	REMOVE
Sodastream Apple Mix	7290002793229	1	= None =	ADD CONSUME	<input type="checkbox"/> Sodastream <input type="checkbox"/> Apple <input type="checkbox"/> Mix	CREATE PRODUCT	REMOVE

### Unknown Barcodes

No unknown barcodes yet.

### Processed Barcodes

2019-08-29 15:19:40: Unknown barcode looked up, found name: Sodastream Apple Mix. Barcode: 7290002793229  
 2019-08-29 15:19:01: Unknown barcode looked up, found name: Sella basmati Rice. Barcode: 890800668362  
 2019-08-29 15:19:00: Unknown barcode looked up, found name: Greek Yoghurt. Barcode: 27030868  
 2019-08-29 15:19:00: Unknown barcode looked up, found name: Pizza reich belegt mit Tomaten, Thunfisch und Käse auf knusprig dünnem Boden, tiefgefroren. Barcode: 4001724819301  
 2019-08-29 15:17:24: Unknown barcode looked up, found name: Coca Cola. Barcode: 54490000

ADD BARCODE

Barcode Buddy
 Help
 Source Code
 License
 Version 1.2.2.0
 by Marc Ole Bulling

## 2.2.3 Adding Barcodes

### Adding Barcodes Manually

The easiest option, ideally for testing out Barcode Buddy: Simply open the web ui and click on “Add barcode”. Enter the barcode (use one line per barcode) and click on “Add”.

If you are using a barcode scanner, but don’t want to attach it to Barcode Buddy (yet), you can also plug it into the device that runs the webbrowser and use it to enter the barcodes in the text field. Each line is parsed as a barcode.

### Adding Barcodes Automatically

The preferred way. Most barcode scanners register as a USB keyboard. That way, it is possible to grab the input and send it to Barcode Buddy.

### Using a physical barcode scanner

#### Barcode scanner connected to an Android device

If you use a barcode scanner that is connected to your Android device, you can use the [Android App](#) to send all input to Barcode Buddy. Open the app, go to Settings and activate `Enable Bluetooth Barcode Scanner`. Then return to the main menu. Every time you scan a barcode while being in the app’s main menu, all input is sent to Barcode Buddy. For more info, refer to [Using Barcode Buddy app for Android](#).

#### Barcode scanner connected to a Linux device

Plug in your barcode scanner to the Linux computer / server you will be using. Run the command `evtest` as root. You will see a list of devices, select the one that is your barcode scanner and remember the number (e.g.. `event6`). Scan a barcode. You will now see output in the `evtest` program. If not, you have selected the wrong source.

### Docker

Create a docker container with

```
docker run -v bbconfig:/config -e ATTACH_BARCODESCANNER=true -p 80:80 -p 443:443 --  
→device /dev/input/eventX f0rc3/barcodebuddy-docker:YOURTAG
```

where X in `--device /dev/input/eventX` is the number of your event you selected previously. You might need to change the values for the ports. Scan a barcode - it should be sent directly to Barcode Buddy.

### Bare Metal

Navigate to the example folder in the Barcode Buddy directory. In the file `grabInput.sh` edit the following values:

- **If your barcode scanner is attached to the same computer / server:**
  - `SCRIPT_LOCATION`: Replace with the location where your `index.php` file is located
- **If the scanner is attached to a different computer / server:**
  - `SERVER_ADDRESS`: Replace with the URL where your `index.php` file can be accessed from
  - `USE_CURL`: Set to “true”
- **If the web server does not run as user `www-data` (uncommon):**



- WWW\_USER: Set to the name of the user

Then run as root

```
bash grabInput.sh /dev/input/eventX
```

where X is the number of your event you selected previously. Scan a barcode - it should be sent directly to Barcode Buddy.

To run the script in the background, run

```
screen -S barcodegrabber -d -m /bin/bash /path/to/the/barcodebuddy/folder/example/  
↪grabInput.sh /dev/input/eventX
```

## Using Barcode Buddy app for Android

Download the app here: [Google Play Store](#)

Once installed, open Barcode Buddy and navigate to the menu **API**. There click on the three dots in the top right corner and select **Add mobile app**. Open your Barcode Buddy app and then scan the displayed QR code. Once the automatic setup is complete, tap on the barcode symbol to start scanning.

## Using a 3rd party application / script

If you want to write your own script, there are two ways to send the barcodes to Barcode Buddy: either by calling `php index.php yourBarcode` or by calling the URL: `https://your.bbuddy.url/api/action/scan?apikey=myApiKey&add=123456`. Only one barcode can be given with each call. Replace `myApiKey` with an API key generated in the main menu. For more information about the API visit [API](#).

## 2.2.4 Menus

### Settings menu

#### General Settings

In this tab you can set the barcodes for changing Barcode Buddy modes. For example, if you scan the barcode “BBUDDY-P”, Barcode Buddy will change to “Purchase” mode and add all following items to your Grocy inventory. By default it is in “Consume” mode. The edit field below allows you to set the time in minutes, which is required to pass in order to revert back to the default “Consume” mode. E.g. if “Purchase” mode is active and the field is set to 10 minutes, Barcode Buddy will revert back to “Consume” mode 10 minutes later. If set to 0, reverting is disabled.

If you scan the “Inventory” barcode, Barcode Buddy will simply output the current stock, but not change any values. If an unknown barcode is scanned, it is added to the regular list.

The “Add to shopping list” barcode adds all future barcodes to the default shopping list.

With the “Revert after single item scan in “Open” or “Spoiled” mode” checkbox ticked, Barcode Buddy only stays in this mode for one scan and then reverts back to the default “Consume” mode. It does not affect the “Purchase” mode however!

With “Remove purchased items from shopping list” enabled, items that are scanned in purchase mode are removed from all Grocy shopping lists.

With “Consume amount of quantity saved for barcode” enabled, Barcode Buddy will check if you saved a quantity for this barcode with the default barcode BBUDDY-Q-[X]. Normally, it will only use this amount for adding products to

the inventory, with this option however it will also remove the same amount from the inventory when you are scanning the barcode in Consume mode.

The option “Use generic names for lookup” makes it easier to tag products. If found, it will use the generic name for a product instead of a brandname. For example instead of using “GreatCompany extra virgin oil”, Barcode Buddy will name the product “Olive Oil”.

When “more verbose logs” is disabled, only barcode scans are logged in the log part of the main page.

You will also find several checkboxes to enable / disable lookup providers. Some might require an API key, which can be entered at the lower section. The lookup order can also be changed by clicking on a provider and moving it to a different position without releasing the mouse key.

The more providers you have activated, the slower the lookup will be (especially for failed lookups). However using more providers also means having higher chances of a successful lookup.

### Grocy API

Here you can change your Grocy API details. Refer to *First Start*.

### Redis Status

If you have installed a redis server (*Setup*), you can enable it here. Once it is enabled, it will cache the Grocy products and therefore speed up Barcode Buddy a lot. It is recommended to use a redis server, but not required.

### Websocket Status

This section gives the status of the websocket server and if Barcode Buddy is able to connect to it

### Chores

This menu lists all available Grocy chores. Simply enter a barcode for a chore and press “Add”. The next time you scan this barcode, the chore will be executed. To change the barcode, simply edit it and press “Edit”. To remove, delete the barcode and press “Edit”.

### Tags

All saved tags are listed here

### Adding tags

Scan a barcode that was not recognized by Grocy yet, but could be looked up. Before pressing “Add” or “Consume” in the main menu, select a word from the list to the right. The next time a barcode is looked up that contains the word, the product is preselected.

### Managing tags

The list shows an overview of the tags. Click on “Delete” to remove the tag.

## Quantities

This features is for products that come in packs containing more than one item.

In the settings you see the quantity barcode (default “BBUDDY-Q-“). If you scan a barcode that starts with this text and has a number at the end, Barcode Buddy sets the quantity of the units from the previously scanned barcode to the number. For example: You scan Barcode “123”, which is a pack of 6 eggs. Then you scan the barcode “BBUDDY-Q-6”. The next time you scan the barcode “123” in purchase mode, Barcode Buddy will automatically add 6 eggs. All barcodes are synced with Grocy. Deleting a quantity in this menu will also unlink the quantity in Grocy.

## API

In this menu you can create and revoke Barcode Buddy API keys. Refer to [API](#)

## Admin

In this menu you can download a backup of your database file. To restore a backup, simply overwrite your current database file (default: `/data/barcodebuddy.db`).

It is also possible to logout, so that you need to enter your username and password again.

## 2.2.5 Barcode Buddy Federation

### Overview

Barcode Buddy Federation is an external service, which enables the user to search the Barcode Buddy Federation database for unknown barcodes. By default it is turned off.

It works by sending all barcodes that are associated with a Grocy product to an external server, so that other users can look them up. No personal data will be stored in this process and no data will be used for commercial purposes.

### Enabling Federation Lookup

The Federation lookup can be enabled and disabled in the menu “Federation”. There you will find a button to enable / disable this feature.

### How to use Federation Lookup

#### Barcode Lookup

In order to use the lookup feature, Federation must be enabled. If a lookup was unsuccessful with other lookup providers, Barcode Buddy will connect to the Federation database and request the name of the product. If found, the name will be used for the barcode and displayed in the main page under “New Barcodes”. You can also change the lookup order in the settings menu.

#### Multiple names

Sometimes you might notice that a blue button appears next to the name. This is the case if multiple names were returned by the Federation server. Click on the button to change the name to another one displayed on the selection. Note: By selecting a new name, you actually vote for the name. So if more people select a better fitting name, this name will be the top result the next time.

## Reporting Barcode Names

In case you encounter an offensive or malicious name, you can report it by clicking on the flag next to the name in the main menu. This flag is only visible if the name was actually provided by a Federation lookup.

## 2.3 Updating Barcode Buddy

### 2.3.1 Docker

To update, run the following command:

```
docker pull f0rc3/barcodebuddy-docker:YOURTAG
```

Then stop the running container and follow the same steps as in SETUP. All userdata will be preserved, as it is saved to the bbuddy volume (-v command)

### 2.3.2 Bare Metal

#### Stable version

To update, download the latest release and unzip it to the directory that contains the old version. Overwrite any existing files.

#### Unstable version

To update, execute the command `git pull`

## 2.4 Advanced usage and configuration

### 2.4.1 Further configuration

#### config.php

Once you started Barcode Buddy for the first time, you will find the file `config.php` in the folder `data`. This file is for further configuration - the following values can be changed:

Argument	Value	Effect	Default
PORT_WEBSOCKET_SERVER	65535	The port that the websocket server listens to. Change if you running multiple instances or the default port is already used by another application.	47631
DATABASE_PATH	A writable path	The path were the database file is written to. Make sure that the webserver does not allow the download of the file. Path includes filename.	./data/barcodebuddy.db
CURL_TIMEOUT	5-60	How long to wait for a request	20
CURL_ALLOW_INSECURE_SSL_CERT	true/false	Allow self-signed SSL certificates	false
CURL_ALLOW_INSECURE_SSL_HOST	true/false	Allow self-signed SSL certificates where the host does not match	false
REQUIRE_API_KEY	true/false	Require API key for authentication when using API	true
DISABLE_AUTHENTICATION	true/false	Disables user management, if enabled, Barcode Buddy will not ask for local username/password	false
IS_DEBUG	true/false	Enable verbose error messages. If you think something is wrong, you can enable this and send a bugreport	false
HIDE_LINK_GROCY	true/false	Set to true to remove the link to Grocy in the top header	false
HIDE_LINK_SCREEN	true/false	Set to true to remove the link to the Screen module in the top header	false
EXTERNAL_GROCY_URL	null or URL	If you are using an internal URL for API communication to Grocy, but require a different URL for external access, you can enter the URL here. (Replaces URL in top header)	null
AUTHENTICATION_BYPASS_NETS	array	List of IPs and subnets that can bypass authentication. If using with a reverse proxy, ensure TRUSTED_PROXIES is set correctly	empty
TRUSTED_PROXIES	array	List of IPs and subnets that will be trusted for X-Forwarded-For header information	empty
SEARCH_ENGINE	String	URL for searchengine to use when manually looking up barcode	<a href="https://google.com/search?q=">https://google.com/search?q=</a>
BASEURL	String	Option to add a baseurl if you are running Barcode Buddy behind a proxy	/
OVERRIDE_USER_CONFIG	mapped array	To override settings that can be set in the UI, uncomment the specific line. Overridden values cannot be changed in the UI afterwards	empty

*Note: config.php is a copy of config-dist.php in the root directory, which is tracked by version control, but not included by php. Any changes in config-dist.php will be ignored by Barcode Buddy*

### configProcessing.inc.php

If you need to change the paths of the config.php or the user database location, you can do this in the configProcessing.inc.php file, found in the incl folder. As editing the file might break updating Barcode Buddy, it is rather recommended to use *Environment variables* instead of editing this file. The following values can be changed:

Argument	Value	Effect	Default
CONFIG_PATH	A writable path	The path were the config.php file is written to. Path includes filename.	./data/config.php
AUTHDB_PATH	A writable path	The path were the user database file is written to. Make sure that the webserver does not allow the download of the file. Path includes filename.	./data/users.db

## 2.4.2 Environment variables

Environment variables can be passed to Barcode Buddy - that way you can configure it without editing any files.

All `const` declarations found in *config.php* can be passed as an environment variable, but must have the prefix `BBUDDY_`.

Example: To disable authentication, you need to set `DISABLE_AUTHENTICATION` to `true`. Therefore you need to pass the variable `BBUDDY_DISABLE_AUTHENTICATION` with the value `true` (see *Passing environment variables to Barcode Buddy*)

**Note:** `OVERRIDDEN_USER_CONFIG` is declared as an array in `config.php`. This is the only environment variable you need to pass as an array with the delimiter `;`.

Example: To set the Grocy API details you need to declare `GROCY_API_URL` and `GROCY_API_KEY`. As you can see in `config.php`, they are part of the `OVERRIDDEN_USER_CONFIG` declaration (basically all configurations that can be changed through the web ui are part of that). You therefore need to pass the environment variable `OVERRIDDEN_USER_CONFIG` with the value `GROCY_API_URL=https://myurl/api/;GROCY_API_KEY=1234`

### Passing environment variables to Barcode Buddy

#### Docker

Pass the variable with the `-e` argument. Example for disabling authentication and setting curl timeout to 30:

```
docker run -d -v bbconfig:/config -e BBUDDY_DISABLE_AUTHENTICATION=true -e BBUDDY_
↪CURL_TIMEOUT_S=30 -p 80:80 f0rc3/barcodebuddy-docker:latest
```

Example for passing API details:

```
docker run -d -v bbconfig:/config -e BBUDDY_OVERRIDDEN_USER_CONFIG="GROCY_API_
↪URL=https://myurl/api/;GROCY_API_KEY=1234" -p 80:80 f0rc3/barcodebuddy-docker:latest
```

#### Bare Metal

You need to add the variable to your Nginx configuration that you created in *Webserver setup*. For each environment variable, add the following line in the `location ~ \.php$` block:

```
fastcgi_param BBUDDY_XXXXX 'value';
```

Example: Disabling authentication and setting curl timeout to 30:

```
[...]
location ~ \.php$ {
    fastcgi_param BBUDDY_DISABLE_AUTHENTICATION 'true';
    fastcgi_param BBUDDY_CURL_TIMEOUT_S '30';
    fastcgi_read_timeout 80;
    include fastcgi_params;
    include snippets/fastcgi-php.conf;
    fastcgi_pass unix:/var/run/php/php7.2-fpm.sock;
}
[...]
```

Example: Passing API details:

```
[...]
location ~ \.php$ {
    fastcgi_param BBUDDY_OVERRIDDEN_USER_CONFIG 'GROCY_API_URL=https://myurl/api/;
    ↪GROCY_API_KEY=1234';
    fastcgi_read_timeout 80;
    include fastcgi_params;
    include snippets/fastcgi-php.conf;
    fastcgi_pass unix:/var/run/php/php7.2-fpm.sock;
}
[...]
```

### 2.4.3 API

Barcode Buddy offers an API that can be reached at `http(s)://your.bbinstall.url/api`. By visiting the URL, you will get current the documentation website with an overview of all API functions.

#### Interacting with the API

Unless disabled, all API calls will need an API key as authentication. A key can be generated in the web UI in the menu “API”. The API key needs to be passed in the body or can be added as a GET variable. Example getting info with curl:

```
curl -X GET -H "BBUDDY-API-KEY: myApiKey" "https://your.bbuddy.url/api/system/info"
```

Manually getting the info by adding the GET variable:

```
https://your.bbuddy.url/api/system/info?apikey=myApiKey
```

All functions that require parameters (*except* `/action/scan`), expect them as a form/post parameter. Example: Setting the current mode to `STATE_PURCHASE (2)`:

```
curl -X POST -H "BBUDDY-API-KEY: [[apiKey]]" -F 'state=2' "https://your.bbuddy.url/
    ↪api/state/setmode"
```

#### Non-standard API: `/action/scan`

As mentioned above, the `/action/scan` also looks for GET parameters, in addition to the regular form/post parameters. This is to make it easier for scripts / apps to pass barcodes to Barcode Buddy.

Instead of the POST parameter `barcode` you can also pass the GET parameter `add` or `text` instead. Example Passing the barcode 123456 by just requesting the URL:

```
https://your.bbuddy.url/api/action/scan?apikey=myApiKey&add=123456
```

### 2.4.4 Plugins

Barcode Buddy offers plugin support. All PHP scripts in the folder `plugins` are automatically loaded. See also the [example script](#).

## 2.5 Contributions

All contributions are very welcome! If you have an issue or would like to add a pull request, please visit our Github pages:

<https://github.com/Forceu/barcodebuddy>

<https://github.com/Forceu/barcodebuddy-docker>

## 2.6 Changelog

### 2.6.1 Overview of all Changes

#### v1.8.0.0: 25 Jan 2021

- Added Barcode Buddy Federation
- Replaced internal quantity management completely with Grocys barcode quantity management. On first startup all your existing quantities will be added to the corresponding Grocy barcode.
- Added back button and mode selection in screen module
- Added opengtindb.org lookup provider
- Tags for deleted Grocy products are shown now as “inactive tags” and can be deleted
- Cache is now synced in background, increasing performance
- All fonts are now hosted locally
- Fixed crash when using Barcode “Consume all”
- Minor fixes and improvements

#### v1.6.5.1: 05 Jan 2021

- Grocy  $\geq$  3.0.1 required
- When creating a new Grocy product, creation time is checked with Grocy’s server time

#### v1.6.5.0: 04 Jan 2021

- PHP 7.2 and php-redis now required
- Added option for Redis caching, which significantly increases performance
- Added drag & drop for lookup providers to change the lookup order
- Product locations are displayed now in Inventory Mode
- Added AlbertHeijn Lookup
- With Grocy 3.0, prefilling a barcode is not possible any more when creating a new product. Barcode Buddy now uses the last created product.
- Fixed adding items to shopping list
- Minor changes and bugfixes



**v1.6.0.0: 22 Dec 2020**

- Grocy >= 3.0.0 required
- Update API for Grocy 3.0.0 (#99)
- Added UPCDatabase.org as a provider (#97) @Teagan42
- Added jumbo.com as provider #96
- Add baseURL config (#93)
- Check if word is used multiple times when creating possible tags, ignore case
- Add to “action required” if unknown barcode associated with item is tare #100 (#101)
- Refactor Provider Lookups for Scalability (#97) @Teagan42
- Added overflow for item name #88, removed X-Frame-Options to work in iframe
- Added min-width for wrapping
- Change Google URL to HTTPS

**v1.5.0.4: 13 May 2020**

- Grocy version 2.7.1+ now required
- Sped up Screen module to display output (up to 3 times faster now)
- Added barcode BBUDDY-CA to consume all items in stock
- Added authentication by IP/IP range - Thanks @cyberjacob
- No API key required if already authorised in web UI
- Added option to consume the same quantity as stored with BBUDDY-Q barcode
- Added example and documentation for Nginx reverse proxy
- API: price and best before date can be passed
- Added option to set external Grocy URL
- Added option to respect grocys quantity conversion
- API menu is now always shown
- Fixed “Add” button not showing on small screens
- Fixed JS problems in Screen module
- Fixed SSE reconnection issue - Thanks @glaaze
- Fixed that barcodes were associated with a product multiple times
- Fixed that tags were not found if the name was using hyphens or other characters
- UI enhancements

**v1.5.0.2: 15 Apr 2020**

- Fixed API call state/setmode
- Environment variable for CONFIG\_PATH was not parsed

- Added variables to hide Grocy and Screen module links
- Other minor changes

### v1.5.0.1: 14 Apr 2020

- Added authentication, can be disabled through config.php
- Added support for Grocys tare feature
- Config.php now found in the data directory
- Added API (if you are adding barcodes with GET, you will need to use the API instead)
- Support for official [Barcode Buddy App](#)
- Fixed race conditions
- Show more detailed errors if they occur
- Barcode Buddy settings can be set through environment variables
- Barcode grabber bash variables can be set through environment variables Thanks. @Biont
- Add logs when adding/consuming new products through UI
- New theme for Screen module. Thanks @hanerd
- Python barcode grabber script for devices that disconnect frequently. Thanks @vondit
- Main ui refreshes through AJAX now without reloading
- Better support for iOS13
- Created robots.txt. Thanks @bbreton09
- Added favicon
- Added admin menu for downloading database backup
- Refactoring / Minor fixes

### v1.4.1.1: 26 Mar 2020

- Fixed problems with SSE, in some cases it was stuck in a feedback loop, causing PHP-FPM to freeze
- Fixed bug that fullscreen setting was not acknowledged
- Added “text” GET variable (alias of “add”) so it can be used with the Android App QR & Barcode Scanner
- Added option to ignore invalid SSL certificates (config.php)
- Added bash script for grabbing barcode scanner input
- Moved documentation to ReadTheDocs
- Detect if API key is rejected by Grocy
- Items that were added by clicking on “Add” in the UI were not removed from the shoppinglist
- Ignore case for sorting products
- Display errors in log view as well
- Set HTTP Agent to BarcodeBuddy
- Check if php-sockets is installed

- Added more docker options
- Minor UI changes

#### **v1.4.0.0: 20 Mar 2020**

- External websockets have been replaced with Server Sent Events. This means that you finally don't need any complicated configuration anymore and that all the websocket features should work without setting anything up. You still need to start the websocket server however, as it is used for internal communication. (SEE is basically a proxy for the websockets)
- Docker image available at <https://github.com/Forceu/barcodebuddy-docker>

#### **v1.3.2.0: 16 Mar 2020**

- Moved database location to “/data/” folder. Existing database will be moved automatically, however you need to make sure that your webserver does not allow access to this directory!
- Added functionality to make running in docker easier
- Tags ignore whitespaces and special characters now
- Added a second variant that grabs barcode scanner input (see example/grabInput\_variant2.py, thanks @ChadOhman )
- Refactoring of code

#### **v1.3.1.1: 24 Oct 2019**

- Fixed bug in which the state reverted to consume immediately
- If an unknown barcode was scanned, the barcode showed up as state in the screen module
- Refactored code.

#### **v1.3.1.0: 14 Oct 2019**

- Fixed issue #14 that disabled buttons when creating a new Grocy product
- The Screen module now shows the current state
- Added settings menu to test websocket connection
- Fixed some websocket bugs

#### **v1.3.0.3: 28 Sep 2019**

- Grocy 2.5.1+ now required
- Screen module now features button to enable sound and wakelock for mobile devices
- Screen module can now be set to open in fullscreen
- Add barcode to add items to the default shopping list

### **v1.3.0.1: 4 Sep 2019**

- fixed several issues with Quantity management

### **v1.3.0.0: 29 Aug 2019**

- Added feature to create new Grocy product
- Added feature to handle multipacks (eg. set quantity per barcode)

### **v1.2.2.0: 13 Aug 2019**

- Added Inventory mode
- Modes can be changed with GET variables
- Setup checks if all required extensions are installed
- Bug fixes

### **v1.2.1.0: 7 Aug 2019**

- Added option to remove purchased item from shopping list
- Many minor fixes, full support for PHP5 now
- Fixed crash from library when websockets were enabled, but server not started

### **v1.2.0.0: 1 Aug 2019**

- Settings are now no longer saved in the config.php file. After upgrading you will be asked to re-enter your Grocy \* API details. If previously active, you need to enable websockets again as well in Menu / Settings.
- Added Chore support - add a barcode for your chore in Menu / Chores.
- Default barcodes were changed, as underscores cannot reliably be output with all barcode scanners

### **v1.1.2.1: 29 Jul 2019**

- Fixed problems that default barcodes were processed and then added to the “unknown barcodes” list
- Added Tag viewer
- Fixed problem where products were not selectable in v1.2.0

### **v1.0.1.1: 28 Jul 2019**

- Added PHP5 support for websocket server
- Hotfix for a communication problem with the database, which stopped Barcode Buddy from working

### **v1.0.0.0: 25 Jul 2019**

- First stable release of the program