

Easy Gacha

This is a very easy, configurable implementation of a [Gashapon](#)-style device.

HEY YOU! / Directions / Instructions / Getting Started / Start Here

Note: These instructions are for starting with the script. If you're using the "Easy Gacha - Ready to use" object, skip step #2.

1. Drop your Gacha items into a prim.
 2. Drop the "Easy Gacha" script into the same prim.
 3. You'll see the following things happen:
 4. Hover-text will appear telling you it needs to be configured
 5. A configuration URL will be sent directly to only you (the owner)
 6. Click the link, and have fun!
-

Features

User-Friendly

- No need to rename any items
- Incredibly easy to configure
- Most configuration done automatically
- Detects and prevents most mistakes
- Can set rarity per item
- Supports 30+ items configured at a time
- Multiple-play option (can buy multiple objects at a time)
- Can limit number of sales per item
- Can limit total number of sales for entire Gacha
- Can payout to multiple people with each play
- Guaranteed to hand out inventory and refund change correctly
- Can be restricted to group-only play
- Can handle no-copy items
- Can configure payment buttons
- Can send IM reports when people play
- Can send Email reports when people play
- Dashboard can be loaded on a smart-phone (with JavaScript enabled)

Geeky

- Only asks for debit permission once if possible
 - Ultra low lag (zero when not in use)
 - All data stored in-world (except for Gacha registry and single JavaScript file)
 - Configuration can be exported
 - Configuration can be imported by hand or from notecard
 - Automatic self memory management
 - Gives statistics about actual sales versus configured rarity
 - Protects against modifying existing object configurations (safe to use in any object)
 - Has an optional API for writing plugins and addons (doesn't spam object unless requested)
 - Guaranteed to only use one system URL at a time
 - Shortens URLs to make them easy to use
-

Frequently Asked Questions

Can I remove the sign from the object?

Yes! It's modifiable, so just unlink the prims and delete the sign. If you have the skill and feel like doing it, you could even re-texture it.

What happens if someone pays too much?

Easy Gacha will give them change.

What happens if someone pays too little?

Easy Gacha will refund the whole amount given and remind them of the price.

What if there aren't enough items for the amount someone paid?

Easy Gacha will refund the excess.

What if there aren't any items left?

Easy Gacha will turn off until more items are available or you reconfigure it.

What if the script gets low on memory?

Easy Gacha will not allow anyone to play, but will give you the ability to export your configuration before restarting the script. I haven't actually been able to reach this point yet without configuring way too many items, but it should be robust enough to handle it.

What if there are no URLs available?

Easy Gacha will still be able to be played if it was already configured, but will not be visible to the registry and will not be able to have its configuration changed. If the owner touches the object, it will try to get a new URL again.

What if I accidentally give someone else the admin link?

Then they can administer your Easy Gacha, change its settings, change the price, etc. If you didn't mean to share, or want to remove that permission, just take your Easy Gacha back into your inventory and drop it in-world again. Each time it is rezzed it will create a new URL.

Why aren't the permissions more open?

Because otherwise I cannot vouch for its safety. If I made the script modifiable, someone with ill intent could make it steal money, then give it away. It would still list me as the creator, but would be malicious.

The creator of the script should be listed as [Zan Lightfoot](#)

Also, if I made the objects modifiable and transferable, someone could put something malicious inside them and they'd still list me as the creator. Because the kit is free and the source code is open, there should be no problem with each person that wants one ordering a free copy from the marketplace.

If you want your notecards and your gacha-box to be transferable, simply create your own objects and notecards (copy the contents from mine) and set the permissions how you please. The script is transferable because it is no-mod.

Where do I go for help?

You may contact me and I will probably help out, but I cannot guarantee a quick (or any) response. This was a labor of love. There is no official support, and Second Life comes... second. Please read the license.

Safety Warning

If the **script** you have wasn't created by [Zan Lightfoot](#), then I cannot vouch for its safety. Whoever created it may have altered it. You have been warned.

Folders

The script can give people a folder which is the name of the Easy Gacha object followed by the date.

For example, if you name the object "My Stuff", then someone might receive a folder that says:

My Stuff (Easy Gacha: 3 items 2013-10-06)

The Awesome People (Contributors)

- Many thanks to [Rix Spyker](#) for creating the excellent mesh Easy Gacha model for everyone to use!
 - Thanks to [beatrux Muircastle](#) for beta testing and giving great feedback!
 - Thanks to [Ravina Bumstein](#) for beta testing and giving great feedback!
 - Thanks to [Nova Spyker \(luna.talon\)](#) for putting up with the long hours, being the first adopter, and beta testing!
-

Caveats / Assumptions

Configuration

If the script is deleted or reset, any configurations not exported and saved to a notecard or elsewhere will be lost.

URL Resource Limitations

The script requires one available URL on the parcel/sim. See the [LSL HTTP server](#) for more details. Specifically, if it cannot obtain a URL, then it cannot be configured. As long as the ownership and inventory haven't changed, it can be stored and re-rezzed, but the URL is required for initial configuration. This can be resolved by finding the script(s) which are eating up more than their fair share of URLs and giving their creators a good slap in the face.

Agent Identification

User/agent identification MUST be provided as UUID to be certain, because SL has yet to implement a way of looking up a UUID from a user-name or legacy-name reliably. W-Hat's agent lookup service has been implemented and seems to work reliably, but should always be verified because it's not official. To that end, W-Hat is used to find the agent key, then the agent key is used to get the username and display-name.

API

```
llMessageLinked( LINK_SET , 3000168 , (string)totalItems , buyerId );
```

If this feature is turned on, this script tells the entire object each time a purchase is made, indicating the total number of items, but not listing the individual items received. Excellent for most-recent board or custom, direct thank-yous.

```
llMessageLinked( LINK_SET , 3000169 , itemName , buyerId );
```

If this feature is turned on, this script tells the entire object which items were received. COULD OVERFLOW THE QUEUE ON MULTI-PLAY. Useful for playing a sound or particle effect when a specific item is bought, or custom reporting. Also useful if you want to keep a subscription list based on received items. Be mindful of your event queue.

```
llMessageLinked( LINK_SET , 3000170 , baseUrl , adminKey );
```

Only occurs if the owner creates and puts an inventory item with the name "EasyGachaAPI SignalOnNewURL" into the same prim as the script. This script will then tell the entire object the new URL that was allocated and the adminKey in use. Useful if you want to automatically signal your own external service to watch the script.

Protocol

Overview

Use the [LSL HTTP server](#) to simplify configuration and control by putting advanced controls on the hosted pages of GitHub, then making browser-to-object calls. This prevents needing any parsing engine within the script, and greatly simplifies the "server-client" type interactions within the script. It means the only third party server needed is the host for the admin/stats interface. It also greatly reduces the memory impact of the script, freeing up space for further configuration.

CORS

Cross-Origin Resource Sharing has a number of security implications within browsers, and not all browsers handle it the same way. Thus the following rules have been adopted:

- Always request/use a non-SSL URL (nothing here is truly private/important)
- Only use the GET and POST verbs
- No custom headers (sadly that includes CORS, see LSL HTTP Server docs)
- Expect no cookies

See [this](#) for some more detail on why.

See [this](#) for a list of supported features in LSL, which sadly doesn't include setting the CORS headers on responses.

To get around all this, we officially load the initial page via the script, and it loads all its libraries, CSS, and images via a single script tag. That way all AJAX calls are technically to the same domain of the original page.

Structure

```
GET secondLifeURL/[?dev=1]#<page>
  Where the browser is sent
GET lslguru.github.io/easy-gacha/<version>/easy-gacha.min.js
  Initial library load which takes care of bootstrapping and loading
  everything else
POST secondLifeURL/[adminKey/]<verb>/<subject>
  Body should always be JSON-encoded array, though it may be an empty
  list. This overhead is acceptable for gaining simplicity in parsing.
```

Reference Links

SecondLife

- [LSL Functions](#)
- [LSL HTTP server](#)

UI Libraries/Frameworks

- [Bootstrap](#)
 - [Marionette](#)
 - [Backbone](#)
 - [jQuery](#)
 - [Handlebars](#)
 - [Moment](#)
 - [Font Awesome](#)
-

Features NOT implemented

Default rarity

Setting a default rarity means we'd have to add to the items list every time a non-configured item was chosen. That would definitely run out of memory eventually, and would also add an unreasonable level of additional complexity. Instead, I'm going to make it as easy as possible to configure a batch of items at the same time. That way you could drop in 10,000 items (would take a while to load...) and configure them ~100 at a time. Memory testing will indicate where the limits to number of configured items is.

Help Icons

There would be so many of them scattered all over the screens that it would be ridiculous. Just move the mouse around... the hover-tips should provide sufficient help.

Infinite Inventory

Yes, this could be done. I even came up with a reasonably fast way to scan the inventory. The configuration becomes a nightmare, and is definitely NOT easy for people...

More than 50 items handed out at a time

Go ahead, try handing out 100 **unsaved** notecards (placeholders) in a folder. See what happens... I sure was surprised! Regular objects don't run into the problem until a higher number, but that's still unacceptable.

gh-pages volo

```
$ grunt volo:add:-amd:underscore:exports=_:v5/vendor/underscore
$ grunt volo:add:-amd:jquery:exports='$':v5/vendor/jquery
$ grunt volo:add:-amd:backbone:exports=Backbone:depends=v5/jquery,v5/underscore:v5/vendor/backbone
$ grunt volo:add:-amd:-noprompt:bootstrap:v5/vendor/bootstrap
$ grunt volo:add:-amd:backbone.wreqr:depends=v5/vendor/backbone:v5/vendor/backbone.wreqr
$ grunt volo:add:-amd:backbone.babysitter:depends=v5/vendor/backbone:v5/vendor/backbone.babysitter
$ grunt volo:add:-amd:marionette:exports=Marionette:depends=v5/vendor/underscore,v5/vendor/backbone,v5/vendor/backbone.wreqr,v5/vendor/bac
```