

Forum:[Hack Forums](#)

Thread: [Steam Backpack System](#)

Closed User - April 23, 2016, 6:25 am

This is a very simple script I wrote ages ago while I was creating a steam-related betting website.

I will only provide visuals as I can't ensure you'll use the same backend system as mine.

Cost.php fetches the item prices from steam market(lowest price) which isn't reliable at all, however you can easily change the script to use steamanalyst(If you purchase API access)

helper.php

PHP Code

```
<?php
```

```
include 'cost.php';
```

```
class playerItems {
```

```
    public $name;  
    public $marketHashName;  
    public $marketName;  
    public $borderColor;  
    public $backgroundColor;  
    public $type;  
    public $tradeable;  
    public $marketable;  
    public $comodity;  
    public $marketTradeRestriction;
```

```
    public $descriptionType = array();  
    public $descriptionValue = array();  
    public $descriptionColor = array();  
    public $descriptionDefIndex = array();  
    public $descriptionIsItemSetName = array();
```

```
    public $actionName = array();  
    public $actionLink = array();
```

```
    public $marketActionName = array();  
    public $marketActionLink = array();
```

```
    public $tagsInternalGame = array();  
    public $tagsName = array();  
    public $tagsCategory = array();  
    public $tagsCategoryName = array();  
    public $tagsColor = array();
```

```
    public $appID;
```

```

public $classID;
public $instanceID;
public $iconURL;
public $iconURLLarge;
public $iconDragURL;
public $price;

}

libxml_use_internal_errors(true);

$playerInv = array();

$i = 0;
$link = "http://steamcommunity.com/profiles/". $id."/inventory/json/730/2";
$string = file_get_contents($link);
$string = utf8_encode($string);
$obj = json_decode($string);

if($obj->success == true)
{
    set_error_handler(function() { /* ignore errors */ });

    foreach($obj->rgDescriptions as $keyB => $valueX){
        $playerInv[] = new playerItems();
    }

    foreach($obj->rgDescriptions as $keyA => $value){

        $playerInv[$i]->name = ($value->name != null ? $value->name : "NULL");
        $playerInv[$i]->marketHashName = ($value->market_hash_name != null ? $value->
market_hash_name : "NULL");
        $playerInv[$i]->marketName = ($value->market_name != null ? $value->market_name :
"NULL");
        $playerInv[$i]->borderColor = ($value->name_color != null ? $value->name_color : "NULL");
        $playerInv[$i]->background-color = ($value->background_color != null ? $value->
background_color : "NULL");
        $playerInv[$i]->type = ($value->type != null ? $value->type : "NULL");
        $playerInv[$i]->tradeable = ($value->tradable != null ? $value->tradable : "NULL");
        $playerInv[$i]->marketable = ($value->marketable != null ? $value->marketable : "NULL");
        $playerInv[$i]->commodity = ($value->commodity != null ? $value->commodity : "NULL");
        $playerInv[$i]->marketTradeRestriction = ($value->market_tradable_restriction != null ? $value
->market_tradable_restriction : "NULL");

        foreach($value->descriptions as $keyB => $desc){
            ($desc->type != null ? array_push($playerInv[$i]->descriptionType, $desc->type) : "" );
            ($desc->value != null ? array_push($playerInv[$i]->descriptionValue, $desc->value) : "" );
            ($desc->color != null ? array_push($playerInv[$i]->descriptionColor, $desc->color) : "" );
            ($desc->app_data->def_index != null ? array_push($playerInv[$i]->descriptionDefIndex,
$desc->app_data->def_index) : "" );
            ($desc->app_data->is_itemset_name != null ? array_push($playerInv[$i]->
descriptionIsItemSetName, $desc->app_data->is_itemset_name) : "" );
        }
    }
}

```

```

foreach($value->actions as $keyC => $action){
    ($action->name != null ? array_push($playerInv[$i]->actionName, $action->name) : "" );
    ($action->link != null ? array_push($playerInv[$i]->actionLink, $action->link) : "" );
}

foreach($value->market_actions as $keyD => $marketAction){
    ($marketAction->name != null ? array_push($playerInv[$i]->marketActionName,
$marketAction->name) : "" );
    ($marketAction->link != null ? array_push($playerInv[$i]->marketActionLink, $marketAction->
link) : "" );
}

foreach($value->tags as $keyE => $tags){
    ($tags->internal_name != null ? array_push($playerInv[$i]->tagsInternalGame, $tags->
internal_name) : "" );
    ($tags->name != null ? array_push($playerInv[$i]->tagsName, $tags->name) : "" );
    ($tags->category != null ? array_push($playerInv[$i]->tagsCategory, $tags->category) : "" );
    ($tags->category_name != null ? array_push($playerInv[$i]->tagsCategoryName, $tags->
category_name) : "" );
    ($tags->color != null ? array_push($playerInv[$i]->tagsColor, $tags->color) : "" );
}

$playerInv[$i]->appId = ($value->appid != null ? $value->appid : "NULL");
$playerInv[$i]->classID = ($value->classid != null ? $value->classid : "NULL");
$playerInv[$i]->instanceID = ($value->instanceid != null ? $value->instanceid : "NULL");
$playerInv[$i]->iconURL = ($value->icon_url != null ? $value->icon_url : "NULL");
$playerInv[$i]->iconURLLarge = ($value->icon_url_large != null ? $value->icon_url_large :
"NULL");
$playerInv[$i]->iconDragURL = ($value->icon_drag_url != null ? $value->icon_drag_url :
"NULL");

$playerInv[$i]->marketHashName = ($value->market_hash_name != null ? $value->
market_hash_name : "NULL");

$playerInv[$i]->price = returnCost($playerInv[$i]->marketHashName);

$i++;
}

restore_error_handler();
}
else{echo "Steam is down, please try again in 2 minutes! <3";}

//print_r($playerInv);

?>
cost.php
PHP Code
<?php
function returnCost($item)
{
$item = str_replace("\'", "'", $item);

```

```

$item = str_replace("\\", "", $item);
$item = str_replace(" ", "%20", $item);
$item = str_replace("\\", "", $item);
$sporn = str_replace("%20", " ", $item);

```

```

$link =
"http://steamcommunity.com/market/priceoverview/?currency=1&appid=730&market_hash_name=" .
$item;
$string = file_get_contents($link);
$obj = json_decode($string);
if($obj->{'success'} == "0") die("notfound");
$lowest_price = $obj->{'lowest_price'};
return $lowest_price;
}
?>

```

A simple example of how you execute it on a webpage (I'm using bootstrap, I won't bother to write up a new example)

NOTE: \$id has to be changed to the logged in users steamid to work properly, SteamAuth works perfectly

```

<?php
$id = 'STEAMID';
include 'helper.php';
if (isset($_GET['id'])) {
    $id = ($_GET['id']);
}

$b = 0;
foreach($playerInv as $key => $item){
    echo"
        <div class=\"col-xs-3\" id=\"\" . ++$b . \"\" name=\"\" . $item->marketHashName .
    \"> <section class=\"panel\" style=\"background:transparent;\"> <div class=\"panel-body text center\"
    \" style=\"border:ridge; border-radius:15px; border-color:white;\">".

    "<center><a class=\"quick-button span3\"style=\"width: 32%;margin-top: 5px; margin-left:5px;\"> <im
    g src=\"http://steamcommunity-a.akamaihd.net/economy/image/\".
        $item->iconURL.\"\" data-toggle=\"tooltip\" data-placement=\"bottom\" title=\"\" .
        \"
        \"
        $item->marketHashName.\"
        \"
        \"
    \"\".
        \"style=\"width: 52px; height: 42px;margin-top: 0px; margin-left: 0px;\".

        "<span class=\"notification blue\">".
        ($item->price != null ? $item->price : "$0.00").
    "</span></a></div></section></div>";

    }
?>

```

Review my code please :)

Sakuto - *April 25, 2016, 9:29 am*

Your code is redundant as hell, it's basically the same thing everywhere, consider using array and loop to do that.

You're also mixing php and HTML, which is bad.