EasyInventory Tutorial Return to Documentation

# EasyInventory

## Adding & Removing Items

## Index

## A) Adding items

### 1. Adding an item

Before we being, let me show you how your code should look like.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

using EasyInventory.ServicesInterface;
using EasyInventory.Repository;

public class EasyInventoryTest : MonoBehaviour
{
    // Use this for initialization
    void Start()
    {
        Bank myBank = new Bank(96);

        Item myItem = new Item(12, 5, true);

    }

    // Update is called once per frame
    void Update()
    {

    }
}
```

### 2. The AddItem method

You can your newly created item to your bank by using the following statement.

```
myBank.AddItem(myItem);
```

This method returns true if the item has been added successfully. If your bank is full, the method will throw a FullItemSlotsException.

### 3. The AddItems method

You can also add multiple items at once. This method will also throw an exception if the inventory is full.

```
Item myItem1 = new Item(12, 5, true);
Item myItem2 = new Item(0, 2, true);
Item myItem3 = new Item(3, 99, true);
Item myItem4 = new Item(77, 1, true);
Item myItem5 = new Item(24, 1, true);

Item[] items = new Item[]
    {
        myItem1,
        myItem2,
        myItem3,
        myItem4,
        myItem5
    };

myBank.AddItems(items);
```

There's nothing too special here.

Continue

## B) Removing items

### 1. The RemoveItem method

Just like the methods above, you can remove items given an item as well.

```
        Item myItem = new Item(12, 5, true);
        myBank.RemoveItem(myItem);
```

If your inventory does not have the inventory item or amount to remove, the method will return False, if it was successful it will return True.

### 2. The RemoveItems method

You can also remove multiple items. Just like the *AddItems* method, we can remove many items at once.

```
        Item myItem1 = new Item(12, 5, true);
        Item myItem2 = new Item(0, 2, true);
        Item myItem3 = new Item(3, 99, true);
        Item myItem4 = new Item(77, 1, true);
        Item myItem5 = new Item(24, 1, true);

        Item[] items = new Item[]
            {
                myItem1,
                myItem2,
                myItem3,
                myItem4,
                myItem5
            };

        myBank.RemoveItems(items);
```

Al we have done here was change the last statement to be *RemoveItems*. If the method was successful, it will return True.

Continue

## C) Checking items

### 1. The HasItem method

Alternatively, before adding or removing items, you can check if your bank contains the current item.

```
Item myItem = new Item(12, 5, true);
myBank.HasItem(myItem);
```

This method will return True if the bank service has the item stored.

## 2. The HasItems method

You can also check multiple items at once by using the following statements.

```
Item myItem1 = new Item(12, 5, true);
Item myItem2 = new Item(0, 2, true);
Item myItem3 = new Item(3, 99, true);
Item myItem4 = new Item(77, 1, true);
Item myItem5 = new Item(24, 1, true);

Item[] items = new Item[]
    {
        myItem1,
        myItem2,
        myItem3,
        myItem4,
        myItem5
    };

myBank.HasItems(items);
```

There's no surprise here, the method will return False if the items are not contained within the bank service.

Continue

Up Next Creating an Inventory instance->

**Developed by Corey St-Jacques**
Questions please contact Corey_stjacques@hotmail.com