Creating A New Currency

Java

Creating a new currency is very easy; simply extend our Currency class

```
Currency
```

After extending the class, ensure to implement its methods, resulting in a class structured as depicted below

```java
package org.insurgencedev.mobcoins;

import lombok.NonNull;
import org.bukkit.entity.Player;
import org.insurgencedev.insurgencesets.api.ISetsAPI;
import org.insurgencedev.insurgencesets.api.currency.Currency;
import org.insurgencedev.insurgencesets.models.currency.TransactionTypes;

public class ExampleCurrency extends Currency {


    public ExampleCurrency() {
        super("example", "ex");
    }

    @Override
    public boolean canAfford(@NonNull Player player, @NonNull Object amount) {
        return true;
    }

    @NonNull
    @Override
    public TransactionTypes handleDeposit(@NonNull Player player, @NonNull Object amount,
String armorSetName) {
        return TransactionTypes.SUCCESS;
```

```
    }

    @NonNull
    @Override
    public TransactionTypes handleTransaction(@NonNull Player player, @NonNull Object amount,
String armorSetName) {
        return TransactionTypes.SUCCESS;
    }
}
```

MobCoins Example

In the next example, we are going to make a currency for SuperMobCoins

```
package org.insurgencedev.mobcoins;

import lombok.NonNull;
import me.swanis.mobcoins.MobCoinsAPI;
import org.bukkit.entity.Player;
import org.insurgencedev.insurgencesets.api.ISetsAPI;
import org.insurgencedev.insurgencesets.api.currency.Currency;
import org.insurgencedev.insurgencesets.models.currency.TransactionTypes;

public class MobCoinCurrency extends Currency {

    public MobCoinCurrency() {
        super("MobCoins", "SM");
    }

    @Override
    public boolean canAfford(@NonNull Player player, @NonNull Object o) {
        return MobCoinsAPI.getProfileManager().getProfile(player).getMobCoins() >= ((Number)
o).longValue();
    }

    @NonNull
    @Override
    public TransactionTypes handleDeposit(@NonNull Player player, @NonNull Object o, String s)
{
```

```
            if (isInvalidSet(s)) {
                return TransactionTypes.FAIL;
            }


MobCoinsAPI.getProfileManager().getProfile(player).setMobCoins(MobCoinsAPI.getProfileManager()
.getProfile(player).getMobCoins() + ((Number) o).longValue());
            return TransactionTypes.SUCCESS;
        }


        @NonNull
        @Override
        public TransactionTypes handleTransaction(@NonNull Player player, @NonNull Object o,
String s) {
            if (isInvalidSet(s)) {
                return TransactionTypes.FAIL;
            }


            long amount = ((Number) o).longValue();
            if (MobCoinsAPI.getProfileManager().getProfile(player).getMobCoins() < amount) {
                return TransactionTypes.FAIL_INSUFFICIENT_FUNDS;
            }


MobCoinsAPI.getProfileManager().getProfile(player).setMobCoins(MobCoinsAPI.getProfileManager()
.getProfile(player).getMobCoins() - ((Number) o).longValue());
            return TransactionTypes.SUCCESS;
        }


        private boolean isInvalidSet(String armorSet) {
            return armorSet == null || ISetsAPI.getArmorSetManager().findArmorSet(armorSet) ==
null;
        }
}
```

You can choose to let the armor sets increase your currency earnings. When your currency is provided to the player, it must trigger an event at which you can listen to and boost accordingly.

```
package org.insurgencedev.mobcoins;
```

```java
import lombok.NonNull;
import me.swanis.mobcoins.MobCoinsAPI;
import me.swanis.mobcoins.events.MobCoinsReceiveEvent;
import org.bukkit.entity.Player;
import org.bukkit.event.EventHandler;
import org.bukkit.event.Listener;
import org.bukkit.inventory.ItemStack;
import org.insurgencedev.insurgencesets.api.ISetsAPI;
import org.insurgencedev.insurgencesets.api.currency.Currency;
import org.insurgencedev.insurgencesets.libs.fo.remain.nbt.NBTItem;
import org.insurgencedev.insurgencesets.models.armorset.ArmorSet;
import org.insurgencedev.insurgencesets.models.currency.TransactionTypes;
import org.insurgencedev.insurgencesets.models.upgrade.Boost;
import org.insurgencedev.insurgencesets.models.upgrade.Upgrade;
import org.insurgencedev.insurgencesets.settings.ArmorSetData;
import org.insurgencedev.insurgencesets.settings.ISetsPlayerCache;

public class MobCoinReceiveListener implements Listener {

    @EventHandler
    public void onEarn(MobCoinsReceiveEvent event) {
        if (!MobCoinsCurrencyAddon.isDependentEnabled()) {
            return;
        }

        Player player = event.getProfile().getPlayer();
        ISetsPlayerCache cache = ISetsPlayerCache.from(player);

        ItemStack[] armorContents = player.getInventory().getArmorContents();
        for (ItemStack item : armorContents) {
            if (item != null) {
                NBTItem nbtItem = new NBTItem(item);
                if (!nbtItem.hasTag("armorSet")) {
                    continue;
                }

                ArmorSet armorSet =
ISetsAPI.getArmorSetManager().findArmorSet(nbtItem.getString("armorSet"));
                if (armorSet == null) {
                    continue;
```

```java
                }

                String armorSetName = armorSet.getName();
                String itemType = item.getType().name().split("_")[1];
                ArmorSetData armorSetData = cache.getArmorSetData(armorSetName);
                if (armorSetData == null) {
                    continue;
                }

                Object levels = getLevelsFromType(itemType, armorSetData);
                if (levels instanceof Integer) {
                    Upgrade upgrade = armorSet.findPieceLevels(itemType, (Integer) levels);
                    if (upgrade == null) {
                        continue;
                    }

                    for (Boost boost : upgrade.getBoosts()) {
                        if ("CURRENCY".equals(boost.getNamespace()) &&
boost.getType().equals("MobCoins")) {
                            double boostAmount =
boost.getBOOST_SETTINGS().getDouble("Boost_Amount");
                            event.setAmount(calcAmountToGive(event.getAmount(), boost,
boostAmount));
                        }
                    }
                }
            }
        }
    }

    private long calcAmountToGive(long amountFromEvent, Boost boost, double boostAmount) {
        if (boost.isPercent()) {
            return (long) (amountFromEvent * (1 + boostAmount / 100));
        } else {
            return (long) (amountFromEvent * (boostAmount < 1 ? 1 + boostAmount :
boostAmount));
        }
    }

    private Object getLevelsFromType(String type, ArmorSetData armorSetData) {
```

```
        return switch (type) {
            case "HEAD", "HELMET" -> armorSetData.getHelmetLevels();
            case "CHESTPLATE" -> armorSetData.getChestplateLevels();
            case "LEGGINGS" -> armorSetData.getLeggingsLevels();
            case "BOOTS" -> armorSetData.getBootsLevels();
            default -> false;
        };
    }
}
```

Main class

```java
package org.insurgencedev.mobcoins;

import org.bukkit.Bukkit;
import org.insurgencedev.insurgencesets.api.ISetsAPI;
import org.insurgencedev.insurgencesets.api.addon.ISetsAddon;
import org.insurgencedev.insurgencesets.api.addon.InsurgenceSetsAddon;
import org.insurgencedev.insurgencesets.libs.fo.Common;


@ISetsAddon(name = "SM-Mobcoins", version = "1.0.0", author = "Insurgence Dev Team",
description = "Use SuperMobCoins's mobcoins as a currency")
public class MobCoinsCurrencyAddon extends InsurgenceSetsAddon {

    @Override
    public void onAddonStart() {
        if (isDependentEnabled()) {
            registerEvent(new MobCoinReceiveListener());
        }
    }


    @Override
    public void onAddonReloadablesStart() {
        if (isDependentEnabled()) {
            ISetsAPI.getCurrencyManager().registerCurrency(new MobCoinCurrency());
        }
    }


    public static boolean isDependentEnabled() {
        return Bukkit.getPluginManager().isPluginEnabled("SuperMobCoins");
    }
```

```
}
```

Revision #8
Created 15 November 2023 22:36:56 by Dragon
Updated 18 March 2024 06:52:26 by Hxtch