InsurgenceSets

WIP will update description later

Setup

This chapter is for the initial setup of the plugin, mainly to get you started.

Installing The Plugin

Download the plugin from BuiltByBit, then drop it in your server's plugins folder.

Creating A New ArmorSet

Create a new set with the armorset command

> /isets setcreate <name>

```
/isets setcreate example
```

> After executing that command you should now be seeing a new file inside /plugins/ISets/sets/example.yml

Now go modify the file to your liking

Booster Types

Currently we only support EdPrison and Vault We are taking suggestions to add support to more plugins, just let us know which ones you would like us to support.

EdPrison

We support all EdPrison currencies

Example:

```
Boosts:
  - Namespace: CURRENCY
    Type: Tokens
    Percent: false
    Settings:
      Boost_Amount: 10
```

We support boosting of all EdPrison enchants

Example:

```
Boosts:
  - Namespace: EDP_ENCHANT
    Type: TokenMiner
    Percent: true
    Settings:
      Boost_Amount: 10
```

Vault

We support Vaults Money currency

```
Boosts:
  - Namespace: CURRENCY
    Type: Money
    Percent: false
    Settings:
      Boost_Amount: 10
```

Messages

List of things that can be done to messages. All messages supports hex colors

Prefix

Insert the Prefix from settings.yml into any message just add `{prefix}` to the message

Json

Minecraft allows you to create more advanced messages using Json you can use a json text generator like this one here

To use json in a message all you have to do is put [JSON] at the start of the message

Example:

```
"[JSON][“”,{"text":"Example ","color":"dark_red"},{"text":"message
","color":"#E124E1"},{"text":"with ","color":"dark_gray"},{"text":"hover
","color":"gold","hoverEvent":{"action":"show_text","contents":"Hover
text!"}},{"text":"text","color":"red"}]"
```

Player variable

You can also show the players name by inserting `{player}` into the message

Send Message As Action Bar

You can choose to send a chat message as an actionbar instead by adding `<actionbar>` to the start of the message

Send Message As Toast

You can choose to send a chat message as a toast by adding `<toast>` to the start of the message

Send Message As A Title

You can choose to send a chat message as a title by adding `<title>` to the start of the message

Send Message As A Boss Bar

You can choose to send a chat message as a bossbar by adding `<bossbar>` to the start of the message, it will show a boss bar for 10 seconds.

Center A Message In Chat

To center a message in chat add `<center>` to the start of the message.

Placeholders

PlaceholderAPI

Below is a list of placeholders that can be used to retrieve data about the armor sets.

| {setid} needs to be replaced with the set name

| {piece} needs to be replaced with helmet | chestplate | leggings | boots

Get the virtual fragments a player has for a specific set

```
%isets_{setid}_fragments%
```

Get the max level for a set

```
%isets_{setid}_{piece}_max_level%
```

Get the current level of a players set piece

```
%isets_{setid}_{piece}_level%
```

Shows whether a player has the set piece equipped

```
%isets_has_{setid}_{piece}_equipped%
```

Shows whether a set piece is upgraded to the max level

```
%isets_is_{setid}_{piece}_maxed%
```

Get the current fragment generator the player has enabled

```
%isets_fragment_generator%
```

Custom Buttons

A guide on how to make a custom button in our main set menu. Mainly used to open menus that are not part of our system like deluxemenus etc

Setting up a button

1. First open up the main-set-menu.yml in a text editor
2. Now navigate down to the `Custom_Buttons: []` and remove the square brackets
3. Then go to the line under it and go in by 2 spaces
4. Now add the things bellow

```
- Icon: SPRUCE_DOOR
  Display_Name: "&c&lExample"
  Slot: 35
  Lore:
    - ""
    - "&7➥ Example action"
    - ""
  Action: |-
```

You now have a custom button ready to be used however click it does not do anything right now.

Importing Java Classes

To import Java classes in to your project is very simple, in this example we will show you how to execute a command

by importing the Bukkit class from the server

```
import "org.bukkit.Bukkit"
```

Now that Bukkit has been imported you can find a list of methods you can execute Bukkit class now lets execute a command

```
import "org.bukkit.Bukkit"

local consoleSender = Bukkit:getConsoleSender() --Getting the console as a sender

Bukkit:dispatchCommand(consoleSender, "say this is an example message.")
```

Now this may be a little hard to figure out if you have never coded anything, so keeping it simple for now.

Action Setups

onButtonClick()

the onButtonClick() is how you tell the button that you want to do something when its being clicked, we will show you how you can execute a command on left clicking example open a deluxemenu menu

Its important you make sure to put player menu clickType inside the ( ) of the onButtonClick(player, menu, clickType)

```
import "org.bukkit.event.inventory.ClickType"
import "org.bukkit.Bukkit"

function onButtonClick(player, menu, clickType)
   if clickType == ClickType.LEFT then
     Bukkit:dispatchCommand(player, "dm open example")
   end
 end
```

Animating the title

Animate the title you can show a temporary title when a button is clicked

However its important you dont animate the title if you are opening another menu with a command, just keep that in mind

```
import "org.bukkit.event.inventory.ClickType"

function onButtonClick(player, menu, clickType)
   if clickType == ClickType.LEFT then
     menu:animateTitle("&cRight Clicked!")
   end
```

```
end
```

ClickType checking

```
import "org.bukkit.event.inventory.ClickType"
import "org.bukkit.Bukkit"

local console = Bukkit:getConsoleSender()

function onButtonClick(player, menu, clickType)
   if clickType == ClickType.Left then
     menu:animateTitle("&cLeft Clicked!")
   elseif clickType == ClickType.RIGHT then
     menu:animateTitle("&cRight Clicked!")
   end
end
```

Debugging

Information on how to enable debugging.

Using The Debug Section inside settings.yml

To utilize the Debug section in the settings.yml file, you need to add Debug sections. As of now, this feature is not yet available, but it will be incorporated in future releases.

List of debug sections that can be enabled.

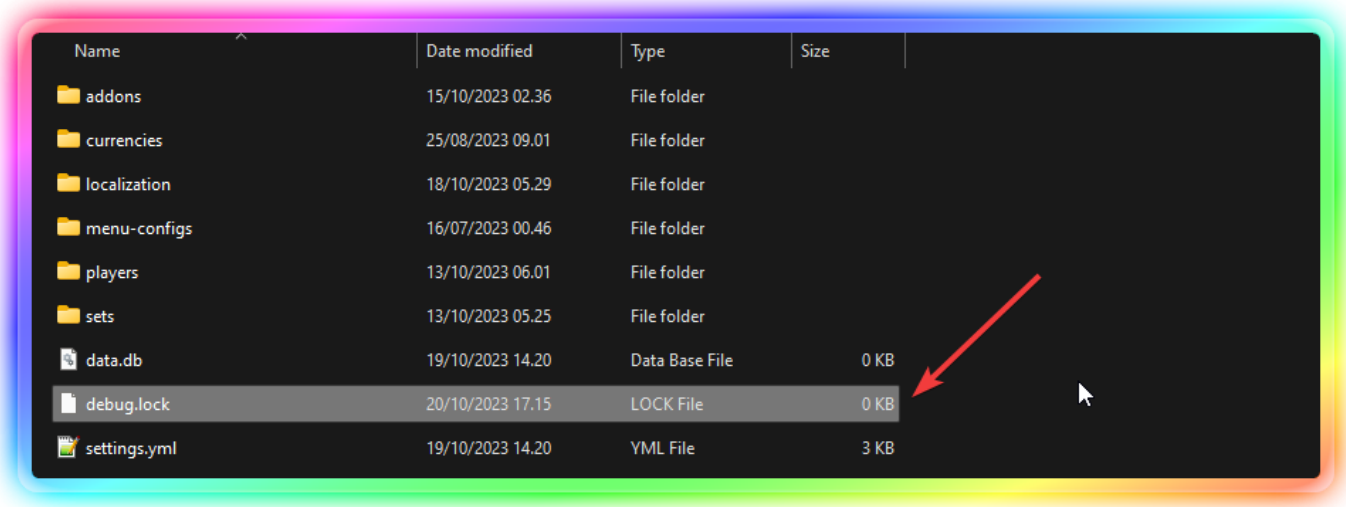
```
Debug:
- 'armorSetManager'
- 'currency'
- 'addonLoader'
- 'register'
- 'auto-register'
- 'armorPieceButton'
- 'fragment_generator'
- 'give_Or_Update_Fragment'
- 'get_Fragment_Item'
```

Enabling Debug Mode

To enable debug mode, follow these steps:

1. Go to the folder where the plugin "ISets" is installed. It is usually located in the "plugins" folder of your server.
2. Inside the "ISets" folder, create a new file and name it "debug.lock". Make sure to include the file extension ".lock".
3. Once the "debug.lock" file is created, the plugin will be in debug mode.

> Note that this file is only used as a marker to indicate that the debug mode is enabled. Deleting the file will turn off the debug mode.



After enabling the debug mode by creating the `debug.lock` file and restarting the server, you can use the `/isets debug` command to generate a `debug.zip` file. This file contains various logs and other debugging information that can be useful in troubleshooting issues with the plugin. You can then send this file to the support team for further analysis and assistance in resolving any issues you may be experiencing.

Developer API

This guide assumes you already know how Java works. If you have any suggestions or questions about the API then reach out to us through Discord or any other contact form

Creating Addon Lua

To accommodate addons that necessitate only a minimal amount of code, we have introduced the option to utilize Lua in place of Java. The process of establishing a Lua addon is straightforward and user-friendly.

Setting up the addon

Addons require these fields or else it will fail to load

```
name = "Example-Addon"
version = "1.0.0"
author = "InsurgenceDev"
description = {
  "This is an example lua addon"
}
```

Printing text to console when addon is started

To print text to the console when the addon is started you need to create a function named "onAddonStart" really its the same as Java addons

```
name = "Example-Addon"
version = "1.0.0"
author = "InsurgenceDev"
description = {
  "This is an example lua addon"
}
```

```
function onAddonStart()
    print("Lua addon started")
end
```

Importing classes from java

to import a class is very simple

```
import 'java.util.ArrayList'
```

Above will import the ArrayList java class example of how to use it

```
import 'java.util.ArrayList'


local arrayList = ArrayList()


function onAddonStart()
  arrayList:add('Example 1')
  arrayList:add('Example 2')
  arrayList:add('Example 3')
  arrayList:add('Example 4')
end
```

Utils

There is a utils package as well you can find it here

Creating A Custom Fragment Generator Lua

It is possible to use Lua to develop a simple or new fragment generator for additional events.

First create a new folder in `/Isets/` named fragment-generators should be like `/Isets/fragment-generators/`. Once that is done, you should create a new file. You can name it whatever you want. For this guide, I will name it `example-generator.Lua`.

Creating the generator file

We will create a generator for the `BlockBreakEvent`. It will be similar to the default generator provided by the plugin.

```
package.path = pluginFolder .. "\\fragment-generators\\listeners\\?.lua"

namespace = "example"
source = "blockBreakEvent"
require "BlockBreakEvent"

function handleGeneration(player, settingsMap)
  if settingsMap:getBoolean("Enabled") and math.random() <= settingsMap:getDouble("Chance")
then
    local amount = (settingsMap:getDouble("Amount_To_Give") <= 0) and 1 or
settingsMap:getDouble("Amount_To_Give")
    local fragment = utils.findFragment(player)

if fragment ~= nil then
  if settingsMap:getBoolean("Physical") then
    givePhys(player, amount, fragment)
  else
    giveVirtual(player, amount, fragment)
```

```
  end
end

   end
end

function givePhys(player, amount, fragment)
  fragment:giveOrUpdateFragment(player, math.floor(amount), false)
  utils.tell(player, "&2You been given " .. amount .. " &6fragments &2for &6" ..
fragment:getArmorSetName())
end

function giveVirtual(player, amount, fragment)
  local cache = utils.getCache(player)
  local currentBalance = cache:getFragmentAmount(fragment:getArmorSetName())
  cache:updateFragmentAmount(fragment.armorSetName, currentVirtualBalance +
math.floor(amount))
  utils.tell(player, "&2You been given " .. amount .. " &6fragments &2for &6" ..
fragment:getArmorSetName())
end
```

Registering an event listener

To actually create a listener for the `BlockBreakEvent`, you should first create a folder inside `/Isets/`. In this example, we are name the folder 'listeners'; `/Isets/fragment-generators/listeners/`. Create a new file and in this case, we name it `BlockBreak.lua`.

```
utils.subscribeToEvent("org.bukkit.event.block.BlockBreakEvent", function(event)
  local player = event:getPlayer()
  local armorSet =
utils.findArmorSet(utils.getCache(player):getFragmentDataManager():getArmorSetFragmentGen())

  if armorSet ~= nil then
    local type = armorSet:getFragmentGeneration():getString("Type")
    local source = armorSet:getFragmentGeneration():getString("Source")
    local fragmentGenerator = utils.findFragmentGenerator(type, source)

    if fragmentGenerator == nil then
      return
```

```
end
if type == namespace then
  local disabledWorlds = armorSet:getFragmentGeneration():getStringList("Disabled_Worlds")
  local worldName = player:getWorld():getName()

  for i = 0, disabledWorlds:size() -1 do
    if disabledWorlds:get(i) == worldName then
      return
    end
  end

      fragmentGenerator:handleGeneration(player, armorSet:getFragmentGeneration())
    end
  end
end)
```

Utils package

The utils package has a few utility functions.

> Util to register an event listener.

```
subscribeToEvent("path to event class", function(event)
--Code here
end)
```

> Util to retrieve an armor set. If none was located, the function will return nil.

```
findArmorSet("name of the armor set")
```

> Util to retrieve a fragment generator. If no fragment generator was discovered, the function will return nil.

```
findFragmentGenerator("type/namespace", "source")
```

> Util to retrieve a player's cache. It will return nil if the cache cannot be located; if it can, it will return the cache.

```
getCache(player)
```

Use this to determine which fragment generator a player currently has enabled. If "none" is returned, then they did not enable one. The name of the armor set for which they activated the generator for will be returned.

```
findFragment(player)
```

Util to send a message to a player. It will auto translate colors.

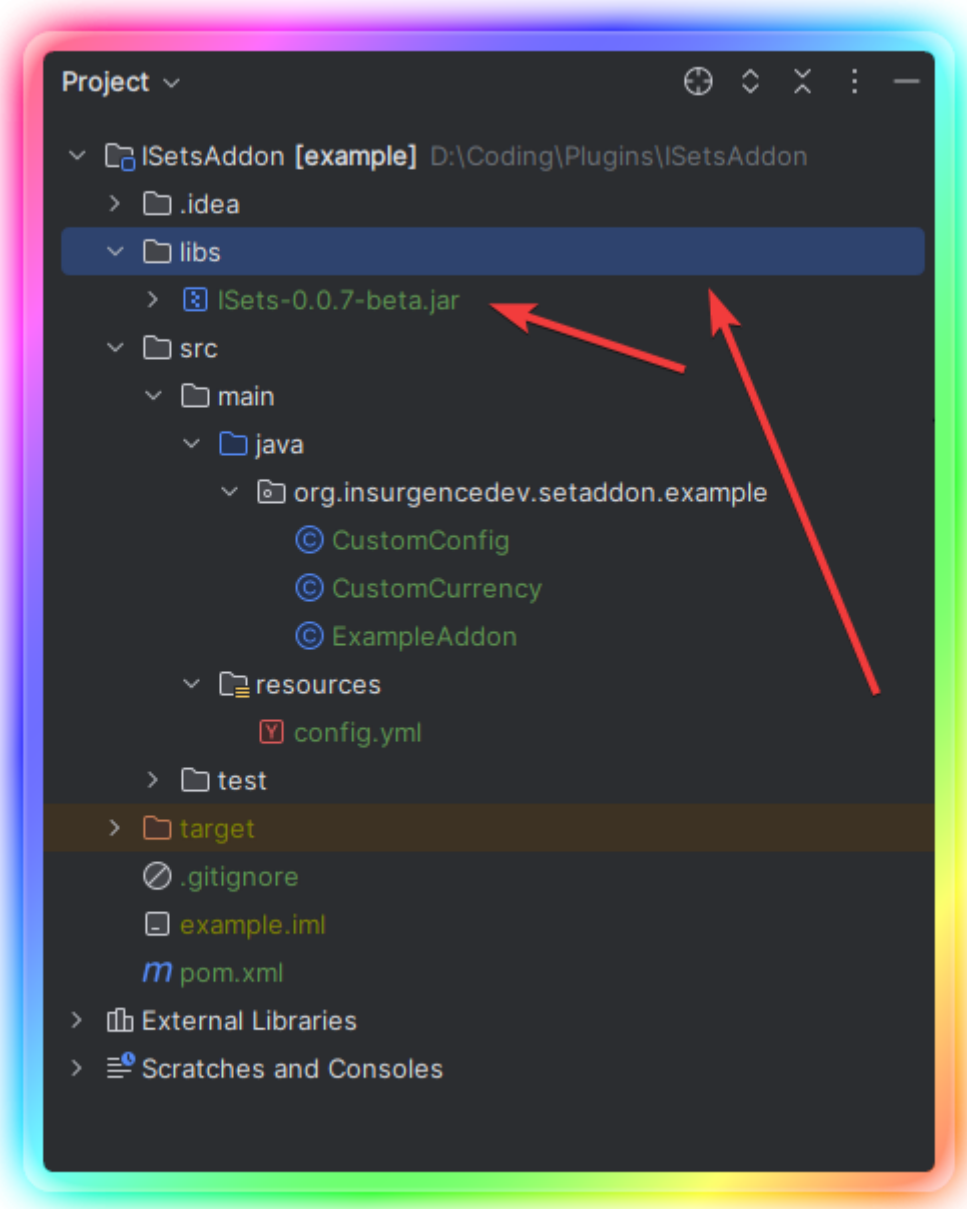
```
tell(player, "message goes here color codes and hex is automaticly translated")
```

Creating An Addon

Setup a new project then right click the project and create a directory called libs.

Once that is done drag in the latest version of ISets

Download example project here



Setting up the pom.xml

You will have to add a dependency pointing to ISets in the libs folder, and you will also need to add a dependency for the spigot api

```
        <dependency>
            <groupId>org.insurgencedev</groupId>
            <artifactId>insurgencesets</artifactId>
            <version>LATEST</version>
            <scope>system</scope>
            <systemPath>${project.basedir}/libs/ISets-0.0.7-beta.jar</systemPath>
        </dependency>
        <dependency>
            <groupId>org.spigotmc</groupId>
            <artifactId>spigot-api</artifactId>
            <version>1.20.2-R0.1-SNAPSHOT</version>
        </dependency>
```

Setting up the main class

```
package org.insurgencedev.setaddon.example;

import org.insurgencedev.insurgencesets.api.addon.ISetsAddon;
import org.insurgencedev.insurgencesets.api.addon.InsurgenceSetsAddon;
import org.insurgencedev.insurgencesets.libs.fo.Common;
import org.insurgencedev.insurgencesets.models.currency.CurrencyManager;

@ISetsAddon(name = "ExampleAddon", version = "1.0.0", author = "Insurgence Dev Team",
description = {"This is a test", "addon it serves no purpose", "other than for testing"})
public class ExampleAddon extends InsurgenceSetsAddon {

    @Override
    public void onAddonStart() {
    }

    @Override
    public void onAddonReloadablesStart() {
    }
```

```
    @Override
    public void onAddonReload() {
    }

    @Override
    public void onAddonStop() {
    }

}
```

Creating a config for your addon

Create a new class and name it whatever you want, then extend AddonConfig.

In the constructor, use loadAddonConfig(). First argument is the location to the default config located in resources.

```
package org.insurgencedev.setaddon.example;

import org.insurgencedev.insurgencesets.api.addon.AddonConfig;

public class CustomConfig extends AddonConfig {

    public static String TEST_STRING = null;



    public CustomConfig() {
        loadAddonConfig("config.yml", "config.yml");
    }

    @Override
    protected void onLoad() {
        TEST_STRING = getString("Test");
    }
}
```

Creating A New Fragment Generator Java

We made it easy to create new custom fragment generators

Create a new class in your project and have it extend

```
FragmentGenerator
```

Once that is done, you should implement its methods

```
public final class ExampleGenerator extends FragmentGenerator {

    public ExampleGenerator() {
        super("CustomGen", "example");
    }

    @Override
    public void handleGeneration(@NotNull Player player, @NotNull SerializedMap map) {

    }
}
```

After the class is configured to your satisfaction, you must add it to our generator list

```
ISetsAPI.getFragmentGeneratorManager().registerFragmentGenerator(new ExampleGenerator());
```

Example Fishing Generator

We're not done yet; here's an example of how to make a listener that listens to the desired event for which you want the generator to be created.

```
package com.insurgencedev.fishinggenerator;

import lombok.Getter;
import org.bukkit.entity.Player;
import org.insurgencedev.insurgencesets.api.FragmentGenerator;
import org.insurgencedev.insurgencesets.api.ISetsAPI;
import org.insurgencedev.insurgencesets.libs.fo.Common;
import org.insurgencedev.insurgencesets.libs.fo.collection.SerializedMap;
import org.insurgencedev.insurgencesets.models.armorset.ArmorSet;
import org.insurgencedev.insurgencesets.models.fragment.Fragment;
import org.insurgencedev.insurgencesets.settings.ISetsPlayerCache;
import org.jetbrains.annotations.NotNull;

@Getter
public final class FishingGenerator extends FragmentGenerator {

    public static final String namespace = "IGen";

    public FishingGenerator() {
        super(namespace, "Fishing");
    }

    @Override
    public void handleGeneration(@NotNull Player player, @NotNull SerializedMap map) {
        if (map.getBoolean("Enabled") && Math.random() <= map.getDouble("Chance") / 100) {
            ISetsPlayerCache cache = ISetsAPI.getCache(player);
            ArmorSet armorSet =
ISetsAPI.getArmorSetManager().findArmorSet(cache.getArmorSetFragmentGen());

            if (armorSet != null) {
                Fragment fragment = armorSet.getFragment();
                int amount = getAmount(map);

                if (map.getBoolean("Physical")) {
                    fragment.giveOrUpdateFragment(player, amount, false);
                } else {
                    cache.updateFragmentAmount(cache.getArmorSetFragmentGen(), amount);
                }
```

```
                Common.tellNoPrefix(player, map.getString("Give_Message").replace("{amount}",
"" + amount));
            }
        }
    }

    private int getAmount(SerializedMap map) {
        return map.getBoolean("Dynamic_Amount", false) &&
GeneratorAddon.getConfig().getFragmentAmount(PlayerFishingListener.getCaughtFish()) != null
                ?
GeneratorAddon.getConfig().getFragmentAmount(PlayerFishingListener.getCaughtFish())
                : map.getInteger("Amount_To_Give");
    }
}
```

The listener

```
package com.insurgencedev.fishinggenerator;

import lombok.Getter;
import org.bukkit.Material;
import org.bukkit.entity.Item;
import org.bukkit.entity.Player;
import org.bukkit.event.EventHandler;
import org.bukkit.event.Listener;
import org.bukkit.event.player.PlayerFishEvent;
import org.bukkit.inventory.ItemStack;
import org.insurgencedev.insurgencesets.api.ISetsAPI;
import org.insurgencedev.insurgencesets.models.armorset.ArmorSet;

import java.util.Arrays;
import java.util.List;

public final class PlayerFishingListener implements Listener {

    @Getter
    private static String caughtFish;

    @EventHandler
```

```
    public void onFish(PlayerFishEvent event) {
        if (event.getState().equals(PlayerFishEvent.State.CAUGHT_FISH) && event.getCaught()
instanceof Item) {
            ItemStack item = ((Item) event.getCaught()).getItemStack();

            if (isFish(item.getType())) {
                Player player = event.getPlayer();
                ArmorSet armorSet =
ISetsAPI.getArmorSetManager().findArmorSet(ISetsAPI.getCache(player).getArmorSetFragmentGen())
;

                if (armorSet != null &&
FishingGenerator.namespace.equals(armorSet.getFragmentGeneration().getString("Type"))) {
                    List<String> disabledWorlds =
armorSet.getFragmentGeneration().getStringList("Disabled_Worlds");
                    if (!disabledWorlds.contains(player.getWorld().getName())) {
                        caughtFish = item.getType().name().toLowerCase();

ISetsAPI.getFragmentGeneratorManager().findFragmentGenerator(FishingGenerator.namespace,
armorSet.getFragmentGeneration().getString("Source"))
                                .handleGeneration(player, armorSet.getFragmentGeneration());
                    }
                }
            }
        }
    }

    private boolean isFish(Material material) {
        return Arrays.asList(Material.COD, Material.SALMON, Material.PUFFERFISH,
Material.TROPICAL_FISH).contains(material);
    }
}
```

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

```
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;

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;
                }

                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

```
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");
    }


}
```