Creating an Effect Booster (Java)

Below is an example of how you can create a booster that applies effects to the player

Main Class

```
package com.insurgencedev.potioneffectsaddon;

import com.insurgencedev.potioneffectsaddon.listeners.BoosterListener;
import org.insurgencedev.insurgenceboosters.api.addon.IBoostersAddon;
import org.insurgencedev.insurgenceboosters.api.addon.InsurgenceBoostersAddon;

@IBoostersAddon(name = "PotionEffects", version = "1.0.0", author = "Insurgence Dev Team",
description = "Apply Potion Effects")
public class PotionEffectsAddon extends InsurgenceBoostersAddon {

    @Override
    public void onAddonStart() {
        new MyConfig();
    }

    @Override
    public void onAddonReloadablesStart() {
        registerEvent(new BoosterListener());
    }

}
```

Listener Class

```
package com.insurgencedev.potioneffectsaddon.listeners;
```

```
import com.insurgencedev.potioneffectsaddon.MyConfig;
import org.bukkit.entity.Player;
import org.bukkit.event.EventHandler;
import org.bukkit.event.EventPriority;
import org.bukkit.event.Listener;
import org.bukkit.event.player.PlayerItemConsumeEvent;
import org.bukkit.event.player.PlayerJoinEvent;
import org.bukkit.event.player.PlayerRespawnEvent;
import org.bukkit.potion.PotionEffectType;
import org.insurgencedev.insurgenceboosters.api.IBoosterAPI;
import org.insurgencedev.insurgenceboosters.events.IBoosterEndEvent;
import org.insurgencedev.insurgenceboosters.events.IBoosterStartEvent;
import org.insurgencedev.insurgenceboosters.libs.fo.Common;
import org.insurgencedev.insurgenceboosters.libs.fo.remain.CompMaterial;
import org.insurgencedev.insurgenceboosters.models.booster.GlobalBoosterManager;
import org.insurgencedev.insurgenceboosters.settings.IBoostersPlayerCache;

import java.util.ArrayList;
import java.util.List;
import java.util.UUID;

public final class BoosterListener implements Listener {

    private final List<UUID> activeList = new ArrayList<>();
    private final String NAMESPACE = "CUSTOM_EFFECT";

    @EventHandler
    private void onStart(IBoosterStartEvent event) {
        if (event.getBoosterData().getType().equals(MyConfig.boosterType) &&
                event.getBoosterData().getNamespace().equals(NAMESPACE)) {

            Player player = event.getPlayer();

            if (MyConfig.disabledWorlds.contains(player.getWorld().getName())) {
                event.getBoosterData().setActive(false);
                return;
            }

            if (alreadyHasEffect(player)) {
                event.getBoosterData().setActive(false);
```

```
                return;
            }

            activeList.add(player.getUniqueId());
            applyEffect(player);
        }
    }

    @EventHandler
    private void onEnd(IBoosterEndEvent event) {
        if (event.getBoosterData().getType().equals(MyConfig.boosterType) &&
                event.getBoosterData().getNamespace().equals(NAMESPACE)) {

            Player player = event.getPlayer();

            activeList.remove(player.getUniqueId());
            removeEffect(player);
        }
    }

    @EventHandler
    private void onConsume(PlayerItemConsumeEvent event) {
        if (CompMaterial.fromItem(event.getItem()).equals(CompMaterial.MILK_BUCKET)) {
            Player player = event.getPlayer();

            IBoostersPlayerCache.BoosterFindResult pResult =
IBoosterAPI.getCache(player).findActiveBooster(MyConfig.boosterType, NAMESPACE);
            if (pResult instanceof IBoostersPlayerCache.BoosterFindResult.Success) {
                Common.runLater(2, () -> applyEffect(player));
                return;
            }

            GlobalBoosterManager.BoosterFindResult gResult =
IBoosterAPI.getGlobalBoosterManager().findBooster(MyConfig.boosterType, NAMESPACE);
            if (gResult instanceof GlobalBoosterManager.BoosterFindResult.Success) {
                Common.runLater(2, () -> applyEffect(player));
            }
        }
    }
```

```
    @EventHandler(priority = EventPriority.MONITOR)
    private void onRespawn(PlayerRespawnEvent event) {
        Player player = event.getPlayer();

        IBoostersPlayerCache.BoosterFindResult pResult =
IBoosterAPI.getCache(player).findActiveBooster(MyConfig.boosterType, NAMESPACE);
        if (pResult instanceof IBoostersPlayerCache.BoosterFindResult.Success) {
            Common.runLater(2, () -> applyEffect(player));
            return;
        }

        GlobalBoosterManager.BoosterFindResult gResult =
IBoosterAPI.getGlobalBoosterManager().findBooster(MyConfig.boosterType, NAMESPACE);
        if (gResult instanceof GlobalBoosterManager.BoosterFindResult.Success) {
            Common.runLater(2, () -> applyEffect(player));
        }
    }

    @EventHandler(priority = EventPriority.MONITOR)
    private void onJoin(PlayerJoinEvent event) {
        GlobalBoosterManager.BoosterFindResult gResult =
IBoosterAPI.getGlobalBoosterManager().findBooster(MyConfig.boosterType, NAMESPACE);
        if (gResult instanceof GlobalBoosterManager.BoosterFindResult.Success) {
            Player player = event.getPlayer();

            if (!activeList.contains(player.getUniqueId())) {
                activeList.add(player.getUniqueId());
                Common.runLater(2, () -> applyEffect(player));
            }
        }
    }

  @EventHandler(priority = EventPriority.MONITOR)
    private void onLeave(PlayerQuitEvent event) {
        Player player = event.getPlayer();

        IBoostersPlayerCache.BoosterFindResult pResult =
IBoosterAPI.getCache(player).findActiveBooster(MyConfig.boosterType, NAMESPACE);
        if (pResult instanceof IBoostersPlayerCache.BoosterFindResult.Success) {
            removeEffect(player);
```

```
            return;
        }

        GlobalBoosterManager.BoosterFindResult gResult =
IBoosterAPI.getGlobalBoosterManager().findBooster(MyConfig.boosterType, NAMESPACE);
        if (gResult instanceof GlobalBoosterManager.BoosterFindResult.Success) {
            removeEffect(player);
        }
    }

    @EventHandler(priority = EventPriority.MONITOR)
    private void onKick(PlayerKickEvent event) {
        Player player = event.getPlayer();

        IBoostersPlayerCache.BoosterFindResult pResult =
IBoosterAPI.getCache(player).findActiveBooster(MyConfig.boosterType, NAMESPACE);
        if (pResult instanceof IBoostersPlayerCache.BoosterFindResult.Success) {
            removeEffect(player);
            return;
        }

        GlobalBoosterManager.BoosterFindResult gResult =
IBoosterAPI.getGlobalBoosterManager().findBooster(MyConfig.boosterType, NAMESPACE);
        if (gResult instanceof GlobalBoosterManager.BoosterFindResult.Success) {
            removeEffect(player);
        }
    }

    private boolean alreadyHasEffect(Player player) {
        PotionEffectType type = PotionEffectType.getByName(MyConfig.effectNamespace);
        return type != null && player.hasPotionEffect(type);
    }

    private void applyEffect(Player player) {
        PotionEffectType type = PotionEffectType.getByName(MyConfig.effectNamespace);
        if (type != null) {
            player.addPotionEffect(type.createEffect(Integer.MAX_VALUE,
MyConfig.effectAmplifier - 1));
        }
    }
```

```
    private void removeEffect(Player player) {
        PotionEffectType type = PotionEffectType.getByName(MyConfig.effectNamespace);
        if (type != null) {
            player.removePotionEffect(type);
        }
    }
}
```

Custom Config Class

```
package com.insurgencedev.potioneffectsaddon;

import lombok.Getter;
import org.insurgencedev.insurgenceboosters.api.addon.AddonConfig;

import java.util.List;

@Getter
public class MyConfig extends AddonConfig {

    public static String boosterType;
    public static String effectNamespace;
    public static int effectAmplifier;
    public static List<String> disabledWorlds;

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

    @Override
    protected void onLoad() {
        boosterType = getString("Type");
        effectNamespace = getString("Effect");
        effectAmplifier = getInteger("Amplifier");
        disabledWorlds = getStringList("Disabled_Worlds");
    }
}
```

Internal Config file

```
Type: Strength
Effect: INCREASE_DAMAGE
Amplifier: 2
Disabled_Worlds:
  - example
```

Revision #11
Created 21 December 2023 10:20:45 by Hxtch
Updated 24 December 2023 15:37:03 by Hxtch