SimplePlugins Documentation

Release 1.5.2-2

liketechnik

Apr 23, 2018

Topics:

1 Usage **3**

2 Bug Reports **5**

3 Documentation **7**

4 Development **9**

5 Versioning scheme **11**

6 Acknowledgements **13**

7 Documentation **15**

7.1 Setup SimplePlugins 15

7.2 Commands with SimplePlugins 17

7.3 Javadoc 17

8 Indices and tables **29**

Create simple minecraft *mods* (Plugins) without setting up a complete forge development environment.

CHAPTER 1

Usage

Clone the blueprint repo version matching the version of SimplePlugins you want to use or download the .zip-Archive of it and extract it (You can rename the directory containing the extracted files to anything you want). The link to the correct blueprint version and the download of the .zip-Archive can be found alongside every release of SimplePlugins.

Inside the directory, run `gradlew setupDecompWorkspace` to initialize forge. Then, depending on the IDE you use, run `gradlew eclipse` or `gradlew idea genIntellijRuns`.

You can now begin creating your plugin:

- Adjust the build.gradle (`group` and `archiveBaseName`) to your needs.
- Either delete and start from scratch or rename the default plugin class
- Enter the name into the manifest attributes

Look at the documentation to get an idea of how achieve certain things.

CHAPTER 2

Bug Reports

Please report any and all issues you encounter while using this mod on the issue tracker. It is important to include the SimplePlugins, Minecraft and Forge version you are using, in your reports/ideas/suggestions.

CHAPTER 3

Documentation

Many things are already documented right in the source code and the javadoc. For all other things and further research, the documentation of SimplePlugins is available at https://simpleplugins.readthedocs.io/en/latest.

CHAPTER 4

Development

Help with the development is highly appreciated, e. g. pull requests, writing missing documentation or adding ideas.

SimplePlugins's source code is hosted on GitHub at https://github.com/liketechnik/SimplePlugins.

If you want to work on SimplePlugins, clone the repository from github and initialize the forge modding environment as described above.

CHAPTER 5

Versioning scheme

- The first number stands for the minecraft version: The first version 1.x was released for MC version 1.12, so version 3.x would refer to MC Version 1.14.
- The second number stands for added features: For every added feature/release, the number gets increased, i. e the current version is x.0 and a new feature is added, the new version would be x.1.
- The third number gets increased for every forge update (inside one MC version).
- The number after the bullet point gets increased after bugfixes.
- The numbers can only increase for the same MC version, after a new MC Version, all other numbers are reset to 0

CHAPTER 6

Acknowledgements

Thanks to the Forge project for creating a great framework for modding minecraft. Without forge, this would not be possible.

I had the idea for this mod after I met the Canary project, a framework for creating server side plugins for minecraft. Sadly, the latest version was made for the minecraft version 1.8, which was one of the reasons for me to start this project.

CHAPTER 7

Documentation

7.1 Setup SimplePlugins

In this part, you will learn how to setup your plugin. When you finished the setup, you can start looking at the other parts of the tutorial and play with MC.

7.1.1 Setup your plugin

First download the blueprint .zip-Archive, which is marked with a `b` in front of the version, for your preferred version from the release page at github. Unpack the archive to a location where you want to work at your plugin, this place is going to be referred to as the *project directory*. You can rename the folder containing all the stuff to whatever you want to call your plugin.

Next head over the forge setup tutorial and step over steps 1-3, because everything is ready to start with step 4 in your project directory. After you finished step 6, your ready to start customizing your plugin.

7.1.2 Make it yours

Now you can start making your plugin truly yours:

1. First open the `build.gradle` file, this file contains information such as the name of your plugin. Change the values of the lines `group` and `archivesBaseName` to your liking (they can both have the same name), in our case this name will be `demoPlugin`. Then choose a name for the main class of the plugin, it is recommended to choose the same name you just used, so our main class will be called `DemoPlugin`. Your `build.gradle` file will then roughly look like this (the lines you should have changed are highlighted):

Listing 7.1: build.gradle

```
buildscript {
repositories {
    jcenter()
    maven { url = "http://files.minecraftforge.net/maven" }
```

```
}
dependencies {
    classpath 'net.minecraftforge.gradle:ForgeGradle:2.3-SNAPSHOT'
}
}
apply plugin: 'net.minecraftforge.gradle.forge'
//Only edit below this line, the above code adds and enables the necessary␣
↪things for Forge to be setup.


version = "1.0"
group = "demoPlugin" // http://maven.apache.org/guides/mini/guide-naming-
↪conventions.html
archivesBaseName = "demoPlugin"

jar {
    manifest {
        attributes(
                'Main-Class': 'DemoPlugin'
        )
    }
}

sourceCompatibility = targetCompatibility = '1.8' // Need this here so␣
↪eclipse task generates correctly.
compileJava {
    sourceCompatibility = targetCompatibility = '1.8'
}

minecraft {
    version = "1.12.2-14.23.2.2618"
    runDir = "run"

    // the mappings can be changed at any time, and must be in the␣
↪following format.
    // snapshot_YYYYMMDD   snapshot are built nightly.
    // stable_#            stables are built at the discretion of the MCP␣
↪team.
    // Use non-default mappings at your own risk. they may not always work.
    // simply re-run your setup task after changing the mappings to update␣
↪your workspace.
    mappings = "snapshot_20171003"
    // makeObfSourceJar = false // an Srg named sources jar is made by␣
↪default. uncomment this to disable.
}
```

2. Then go into the `src/main/java` folder where all your classes will reside. It contains a default class you can rename to the class name you chose in step 1. Then open up the file and change the class name and modify the string passed to the constructor of the superclass too. (You can also edit the author and date information of the javadoc or delete it, if you want). After this, your main class should look like this (changes are highlighted again):

Listing 7.2: src/main/java/DemoPlugin.java

```
import simplePlugins.plugins.api.Plugin;

/**
```

```
* @author tutorial
* @version 1.0
* @date 18 of Februar 2018
*/
public class DemoPlugin extends Plugin {

    public DemoPlugin() {
        super("DemoPlugin");
    }
}
```

Now you're ready to create your first plugin. You have multiple options what you can start with:

- Create your first commands.
- React to events and modify the game.

7.2 Commands with SimplePlugins

With commands you can make minecraft do something, everytime a player uses your command. In this part of the documentation, your are going to learn how to create commands with SimplePlugins.

7.2.1 Your first command

If you have already setup your plugin, you are ready to create your first command now. If you did not setup your plugin yet and need help with it, take a look at *Setup SimplePlugins*.

In this tutorial, we are going to explain the creation of the command steyp by step. In the tutorials that enlarge the topic commands, we will show you the whole command and only explain the new parts.

To begin with the creation of your command, open your plugin's main class. Add a method starting with the name of your command and ending with `Command`, that return nothing and takes a simplePlugins.plugins.api.wrappers.EntityPlayerWrapper:

```
```

7.3 Javadoc

7.3.1 simplePlugins.plugins.api

EventListener

public interface **`EventListener`**

Author Florian Warzecha

Plugin

public abstract class **`Plugin`**

Author Florian Warzecha

Fields

logger

public Logger **`logger`**

Constructors

Plugin

public **`Plugin`** (String *name*)

Methods

getName

public String **`getName`** ()

registerCommandClasses

public void **`registerCommandClasses`** (FMLServerStartingEvent *event*)

This is called during registration of commands. Allows the registration of own command *classes* via the `net.minecraftforge.fml.common.event.FMLServerStartingEvent.registerServerCommand(ICommand)` method.

registerEventClasses

public void **`registerEventClasses`** ()

This is called during registration of events. Allows the registration of own event *classes* via the `net.minecraftforge.fml.common.eventhandler.EventBus.register(Object)` method. Call this method on the following instance: `net.minecraftforge.common.MinecraftForge.EVENT_BUS`

setLogger

public void **`setLogger`** (Logger *logger*)

7.3.2 simplePlugins.plugins.api.annotations

ParamCommand

public @interface **`ParamCommand`**

Author Florian Warzecha

SimpleCommand

public @interface **SimpleCommand**

Author Florian Warzecha

7.3.3 simplePlugins.plugins.api.commands

ParamCommand

public class **ParamCommand** extends *SimpleCommand*

Author Florian Warzecha

Constructors

ParamCommand

public **ParamCommand** (String *name*, String *usage*, String[] *paramsUsage*, Method *command*, *Plugin* *plugin*)

Methods

execute

public void **execute** (MinecraftServer *server*, ICommandSender *sender*, String[] *args*)

getUsage

public String **getUsage** (ICommandSender *sender*)

SimpleCommand

public class **SimpleCommand** extends CommandBase

Author Florian Warzecha

Fields

command

protected Method **command**

name

protected String **name**

plugin

protected *Plugin* **`plugin`**

usage

protected String **`usage`**

Constructors

SimpleCommand

public **`SimpleCommand`** (String *name*, String *usage*, Method *command*, *Plugin* *plugin*)

Methods

execute

public void **`execute`** (MinecraftServer *server*, ICommandSender *sender*, String[] *args*)

getName

public String **`getName`** ()

getPlugin

public *Plugin* **`getPlugin`** ()

getUsage

public String **`getUsage`** (ICommandSender *sender*)

7.3.4 simplePlugins.plugins.api.events

EventWrapper

public class **`EventWrapper`**

Author Florian Warzecha

Constructors

EventWrapper

public **`EventWrapper`** (Event *event*)

Methods

cancel

public void **`cancel`** ()

getUnwrapped

public Event **`getUnwrapped`** ()

7.3.5 simplePlugins.plugins.api.events.entity

EntityEventWrapper

public class **`EntityEventWrapper`** extends *EventWrapper*

Author Florian Warzecha

Constructors

EntityEventWrapper

public **`EntityEventWrapper`** (EntityEvent *event*)

Methods

getEntity

public *EntityWrapper* **`getEntity`** ()

getUnwrapped

public EntityEvent **`getUnwrapped`** ()

7.3.6 simplePlugins.plugins.api.events.entity.living

LivingEventWrapper

public class **`LivingEventWrapper`** extends *EntityEventWrapper*

Author Florian Warzecha

Constructors

LivingEventWrapper

public **`LivingEventWrapper`** (LivingEvent *event*)

Methods

getEntityLiving

public *EntityLivingBaseWrapper* **`getEntityLiving`** ()

getUnwrapped

public LivingEvent **`getUnwrapped`** ()

7.3.7 simplePlugins.plugins.api.events.entity.player

PlayerEventWrapper

public class **`PlayerEventWrapper`** extends *LivingEventWrapper*

Author Florian Warzecha

Constructors

PlayerEventWrapper

public **`PlayerEventWrapper`** (PlayerEvent *event*)

Methods

getEntityPlayer

public *EntityPlayerWrapper* **`getEntityPlayer`** ()

getUnwrapped

public PlayerEvent **getUnwrapped**()

PlayerInteractEventWrapper

public class **PlayerInteractEventWrapper** extends *PlayerEventWrapper*

Author Florian Warzecha

Fields

event

PlayerInteractEvent **event**

Constructors

PlayerInteractEventWrapper

public **PlayerInteractEventWrapper**(PlayerInteractEvent *event*)

Methods

getUnwrapped

public PlayerInteractEvent **getUnwrapped**()

7.3.8 simplePlugins.plugins.api.events.entity.player.itemUseEvent

ItemUseEvent

public class **ItemUseEvent** extends *PlayerInteractEventWrapper*

Author Florian Warzecha

Constructors

ItemUseEvent

public **ItemUseEvent**(PlayerInteractEvent.RightClickItem *event*)

Methods

getConverter

public static Class<?> **getConverter**()

getUnwrapped

public PlayerInteractEvent.RightClickItem **getUnwrapped**()

getWorld

public *WorldWrapper* **getWorld**()

ItemUseEventConverter

public class **ItemUseEventConverter**

Author Florian Warzecha

Constructors

ItemUseEventConverter

public **ItemUseEventConverter**(Method *eventReceiver*, *Plugin* *plugin*)

Methods

onItemUse

public void **onItemUse**(PlayerInteractEvent.RightClickItem *event*)

7.3.9 simplePlugins.plugins.api.wrappers

EntityPlayerWrapper

public class **EntityPlayerWrapper** extends *EntityLivingBaseWrapper*

Author Florian Warzecha

Constructors

EntityPlayerWrapper

public **EntityPlayerWrapper**(EntityPlayer *player*)

Methods

getUnwrapped

public EntityPlayer **getUnwrapped**()

WorldWrapper

public class **WorldWrapper**

Author Florian Warzecha

Constructors

WorldWrapper

public **WorldWrapper**(World *world*)

Methods

attemptSetBlock

public boolean **attemptSetBlock**(Block *block*, BlockPos *pos*)

attemptSetBlock

public boolean **attemptSetBlock**(Block *block*, double *x*, double *y*, double *z*)

forceSetBlock

public boolean **forceSetBlock**(Block *block*, BlockPos *pos*)

forceSetBlock

public boolean **forceSetBlock**(Block *block*, double *x*, double *y*, double *z*)

getBlockAt

public Block **getBlockAt**(BlockPos *pos*)

getBlockAt

public Block **getBlockAt**(double *x*, double *y*, double *z*)

getUnwrapped

public World **getUnwrapped**()

isBlockAt

public boolean **isBlockAt**(BlockPos *pos*)

isBlockAt

public boolean **isBlockAt**(double *x*, double *y*, double *z*)

removeBlockAt

public boolean **removeBlockAt**(BlockPos *pos*)

removeBlockAt

public boolean **removeBlockAt**(double *x*, double *y*, double *z*)

spawnEntity

public boolean **spawnEntity**(Entity *entity*, double *x*, double *y*, double *z*)

spawnEntity

public boolean **spawnEntity**(Entity *entity*, BlockPos *pos*)

7.3.10 simplePlugins.plugins.api.wrappers.entity

EntityLivingBaseWrapper

public class **EntityLivingBaseWrapper** extends *EntityWrapper*

Author Florian Warzecha

Constructors

EntityLivingBaseWrapper

public **EntityLivingBaseWrapper**(EntityLivingBase *entity*)

Methods

addHealth

public void **addHealth** (float *amount*)

getUnwrapped

public EntityLivingBase **getUnwrapped** ()

kill

public void **kill** ()

EntityWrapper

public class **EntityWrapper**

Author Florian Warzecha

Constructors

EntityWrapper

public **EntityWrapper** (Entity *entity*)

Methods

forceTeleport

public void **forceTeleport** (double *x*, double *y*, double *z*)

forceTeleport

public void **forceTeleport** (BlockPos *pos*)

getUnwrapped

public Entity **getUnwrapped** ()

getWorld

public *WorldWrapper* **getWorld** ()

sendMessageTo

public void **sendMessageTo** (String *message*, TextFormatting *color*)

sendMessageTo

public void **sendMessageTo** (String *message*)

CHAPTER 8

Indices and tables

- genindex
- modindex
- search

Index

A

addHealth(float) (Java method), 27
attemptSetBlock(Block, BlockPos) (Java method), 25
attemptSetBlock(Block, double, double, double) (Java method), 25

C

cancel() (Java method), 21
command (Java field), 19

E

EntityEventWrapper (Java class), 21
EntityEventWrapper(EntityEvent) (Java constructor), 21
EntityLivingBaseWrapper (Java class), 26
EntityLivingBaseWrapper(EntityLivingBase) (Java constructor), 26
EntityPlayerWrapper (Java class), 24
EntityPlayerWrapper(EntityPlayer) (Java constructor), 24
EntityWrapper (Java class), 27
EntityWrapper(Entity) (Java constructor), 27
event (Java field), 23
EventListener (Java interface), 17
EventWrapper (Java class), 20
EventWrapper(Event) (Java constructor), 21
execute(MinecraftServer, ICommandSender, String[]) (Java method), 19, 20

F

forceSetBlock(Block, BlockPos) (Java method), 25
forceSetBlock(Block, double, double, double) (Java method), 25
forceTeleport(BlockPos) (Java method), 27
forceTeleport(double, double, double) (Java method), 27

G

getBlockAt(BlockPos) (Java method), 25
getBlockAt(double, double, double) (Java method), 25
getConverter() (Java method), 24
getEntity() (Java method), 21
getEntityLiving() (Java method), 22
getEntityPlayer() (Java method), 22
getName() (Java method), 18, 20
getPlugin() (Java method), 20
getUnwrapped() (Java method), 21–27
getUsage(ICommandSender) (Java method), 19, 20
getWorld() (Java method), 24, 27

I

isBlockAt(BlockPos) (Java method), 26
isBlockAt(double, double, double) (Java method), 26
ItemUseEvent (Java class), 23
ItemUseEvent(PlayerInteractEvent.RightClickItem) (Java constructor), 23
ItemUseEventConverter (Java class), 24
ItemUseEventConverter(Method, Plugin) (Java constructor), 24

K

kill() (Java method), 27

L

LivingEventWrapper (Java class), 22
LivingEventWrapper(LivingEvent) (Java constructor), 22
logger (Java field), 18

N

name (Java field), 19

O

onItemUse(PlayerInteractEvent.RightClickItem) (Java method), 24

P

ParamCommand (Java annotation), 18
ParamCommand (Java class), 19
ParamCommand(String, String, String[], Method, Plugin) (Java constructor), 19
PlayerEventWrapper (Java class), 22
PlayerEventWrapper(PlayerEvent) (Java constructor), 22
PlayerInteractEventWrapper (Java class), 23
PlayerInteractEventWrapper(PlayerInteractEvent) (Java constructor), 23
Plugin (Java class), 17
plugin (Java field), 20
Plugin(String) (Java constructor), 18

R

registerCommandClasses(FMLServerStartingEvent) (Java method), 18
registerEventClasses() (Java method), 18
removeBlockAt(BlockPos) (Java method), 26
removeBlockAt(double, double, double) (Java method), 26

S

sendMessageTo(String) (Java method), 28
sendMessageTo(String, TextFormatting) (Java method), 28
setLogger(Logger) (Java method), 18
SimpleCommand (Java annotation), 19
SimpleCommand (Java class), 19
SimpleCommand(String, String, Method, Plugin) (Java constructor), 20
simplePlugins.plugins.api (package), 17
simplePlugins.plugins.api.annotations (package), 18
simplePlugins.plugins.api.commands (package), 19
simplePlugins.plugins.api.events (package), 20
simplePlugins.plugins.api.events.entity (package), 21
simplePlugins.plugins.api.events.entity.living (package), 22
simplePlugins.plugins.api.events.entity.player (package), 22
simplePlugins.plugins.api.events.entity.player.itemUseEvent (package), 23
simplePlugins.plugins.api.wrappers (package), 24
simplePlugins.plugins.api.wrappers.entity (package), 26
spawnEntity(Entity, BlockPos) (Java method), 26
spawnEntity(Entity, double, double, double) (Java method), 26

U

usage (Java field), 20

W

WorldWrapper (Java class), 25
WorldWrapper(World) (Java constructor), 25