





How to make a Minecraft mod Using Minecraft Forge

Andrew Pietila
Technical Analyst @ cPanel

Projects | All tools ¹

1 NEW UPDATE [Update all](#)

 **CLion**
2017.1.2 [Update](#) 

 **IntelliJ IDEA Ultimate**
2017.2.3 

▼ AVAILABLE TOOLS

 **AppCode**
Smart IDE for iOS/OS X development [Install](#) ▼

 **CLion**
A smart cross-platform IDE for C and C++ [Install](#) ▼

 **DataGrip**
Many databases, one tool [Install](#) ▼

 **Gogland**
Up and Coming Go IDE [Install](#) ▼

 **IntelliJ IDEA Community**
Capable and ergonomic Java IDE [Install](#) ▼

Step 1: Open IntelliJ

Projects All tools ¹

1 NEW UPDATE

[Update all](#)

 **CLion**
2017.1.2 [Update](#) 

 **IntelliJ IDEA Ultimate**
2017.2.3 

▼ AVAILABLE TOOLS

 **AppCode**
Smart IDE for iOS/OS X development [Install](#) ▼

 **CLion**
A smart cross-platform IDE for C and C++ [Install](#) ▼

 **DataGrip**
Many databases, one tool [Install](#) ▼

 **Gogland**
Up and Coming Go IDE [Install](#) ▼

 **IntelliJ IDEA Community**
Capable and ergonomic Java IDE [Install](#) ▼

Step 1: Open IntelliJ

- Why IntelliJ?

Projects All tools ¹

1 NEW UPDATE

[Update all](#)

 **CLion**
2017.1.2 [Update](#) 

 **IntelliJ IDEA Ultimate**
2017.2.3 

▼ AVAILABLE TOOLS

 **AppCode**
Smart IDE for iOS/OS X development [Install](#) ▼

 **CLion**
A smart cross-platform IDE for C and C++ [Install](#) ▼

 **DataGrip**
Many databases, one tool [Install](#) ▼

 **Gogland**
Up and Coming Go IDE [Install](#) ▼

 **IntelliJ IDEA Community**
Capable and ergonomic Java IDE [Install](#) ▼

Step 1: Open IntelliJ

- Why IntelliJ?
 - Several mod developers use IntelliJ

Projects All tools ¹

1 NEW UPDATE

[Update all](#)

 **CLion**
2017.1.2 [Update](#) 

 **IntelliJ IDEA Ultimate**
2017.2.3 

▼ AVAILABLE TOOLS

 **AppCode**
Smart IDE for iOS/OS X development [Install](#) ▼

 **CLion**
A smart cross-platform IDE for C and C++ [Install](#) ▼

 **DataGrip**
Many databases, one tool [Install](#) ▼

 **Gogland**
Up and Coming Go IDE [Install](#) ▼

 **IntelliJ IDEA Community**
Capable and ergonomic Java IDE [Install](#) ▼

Step 1: Open IntelliJ

- Why IntelliJ?
 - Several mod developers use IntelliJ
 - Allows easy access to decompiled Minecraft code

Step 2: Create new project



IntelliJ IDEA

Version 2017.2.3

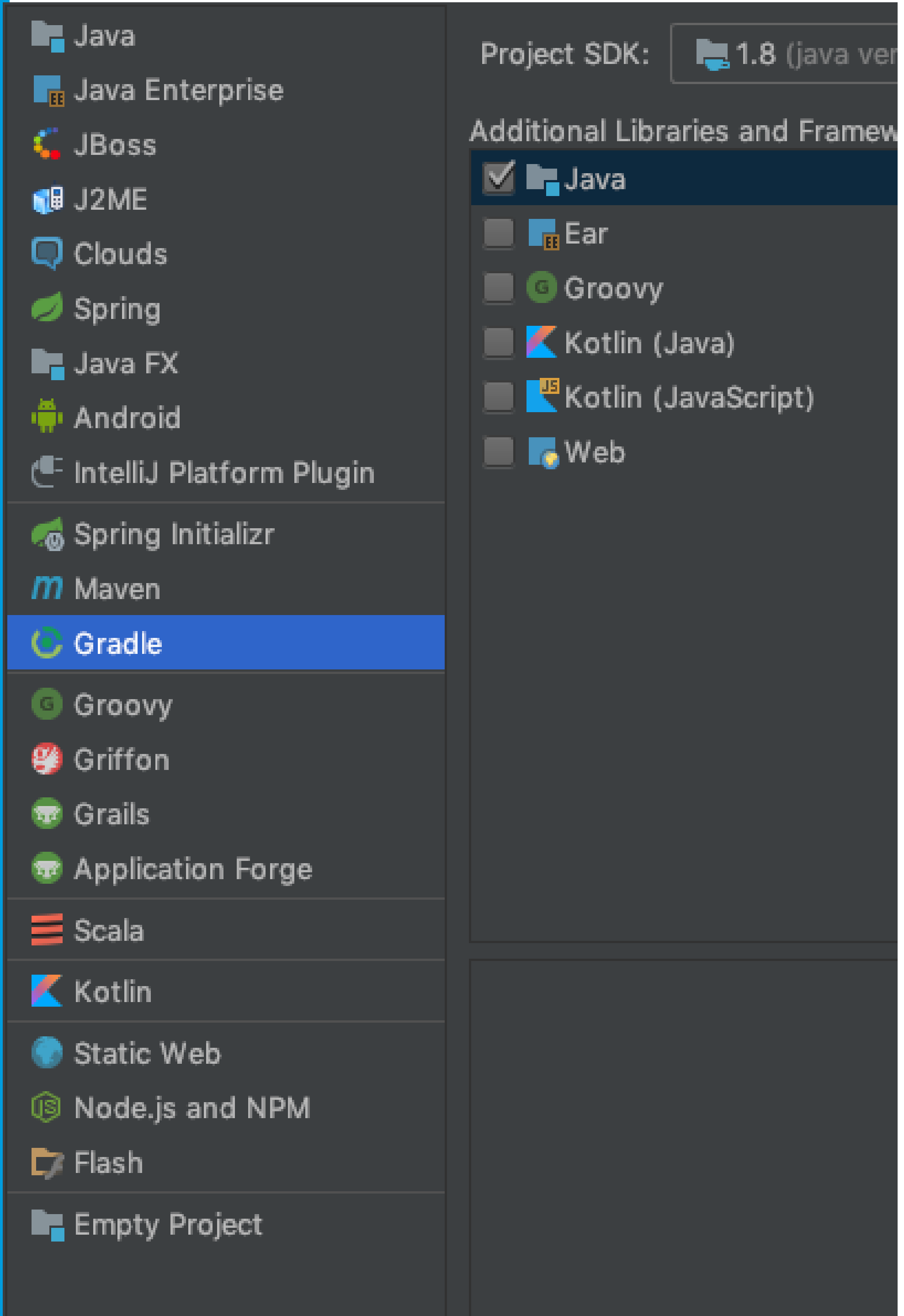
 Create New Project

 Import Project

 Open

 Check out from Version Control ▾

Step 2: Create new project



Step 3: build.gradle



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

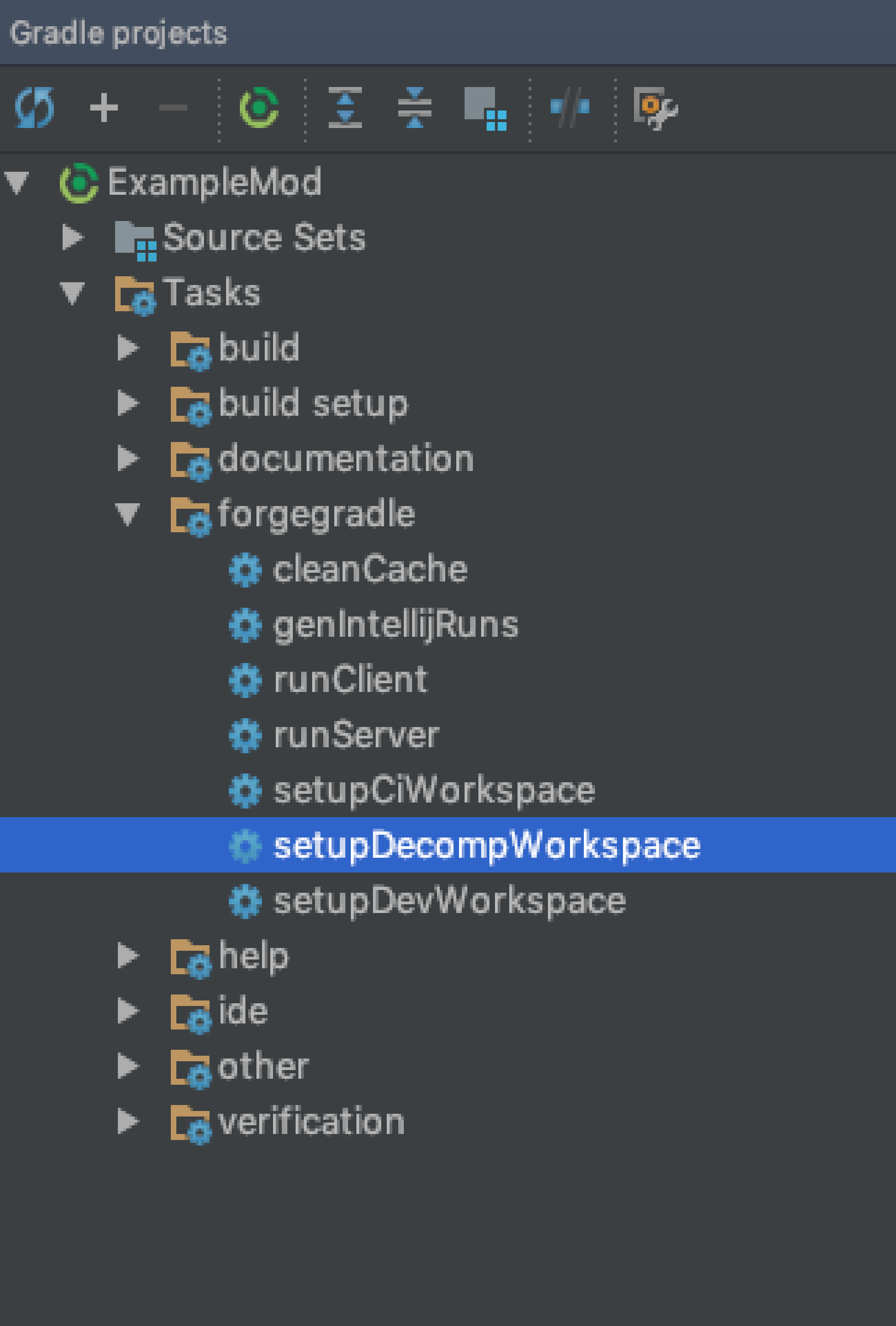
    dependencies {
        classpath group: 'net.minecraftforge.gradle', name: 'ForgeGradle', version:
            '2.3-SNAPSHOT'
    }
}
```


Step 3: build.gradle

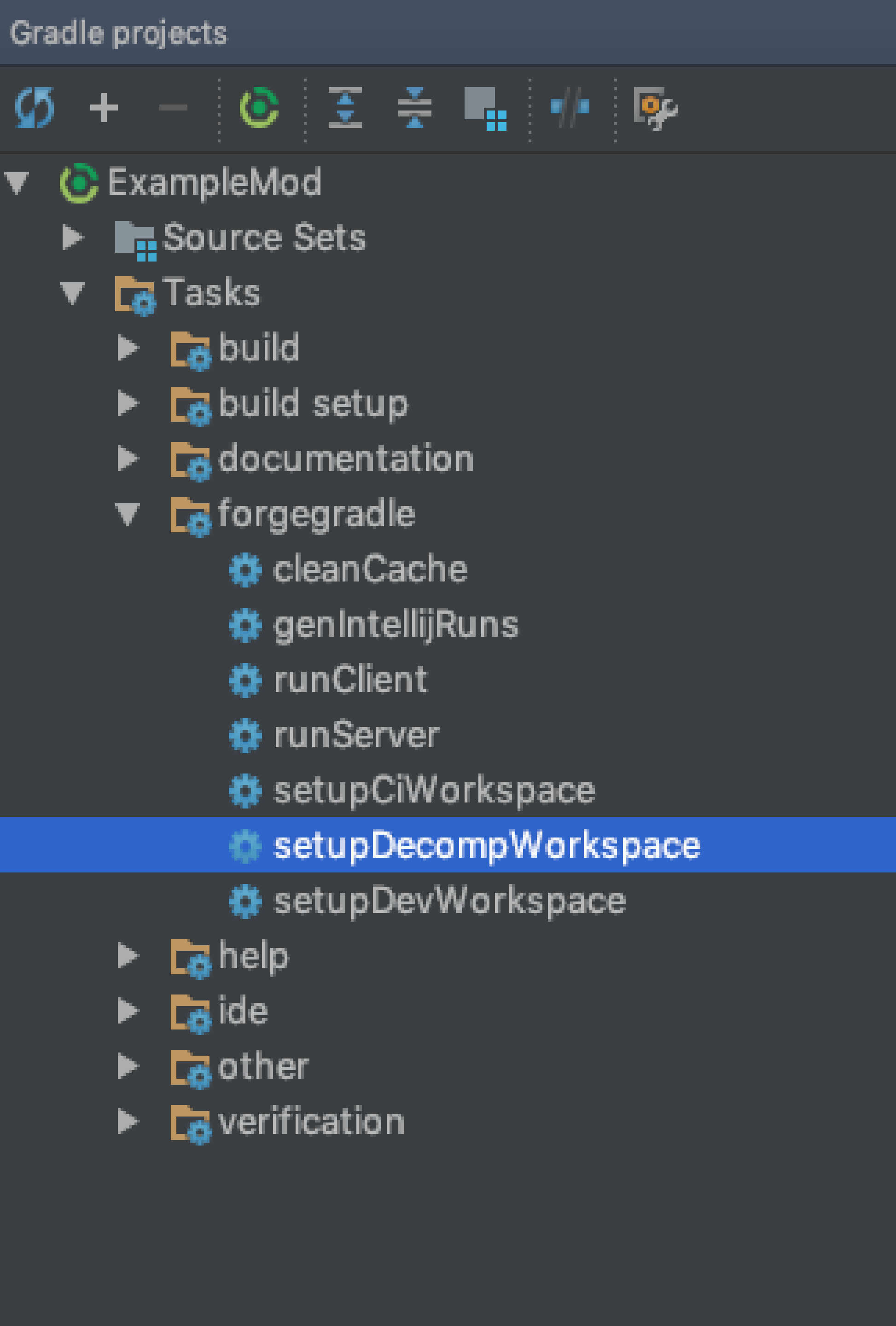


```
>_ apply plugin: 'net.minecraftforge.gradle.forge'
```

```
minecraft {  
    version = "1.12.1-latest"  
    runDir = "run"  
    mappings = "snapshot_20170907"  
}
```

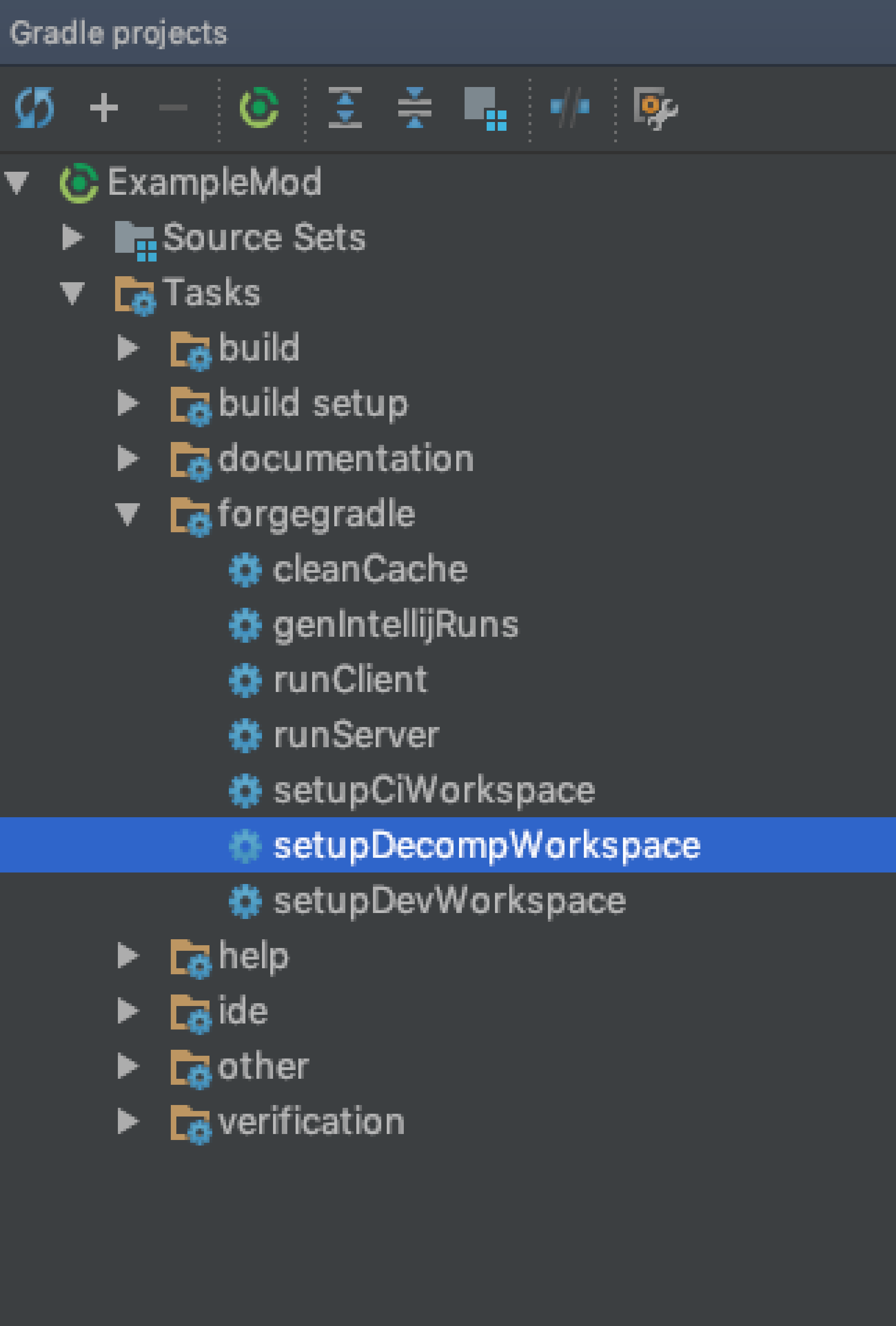


Step 4: setupDecompWorkspace



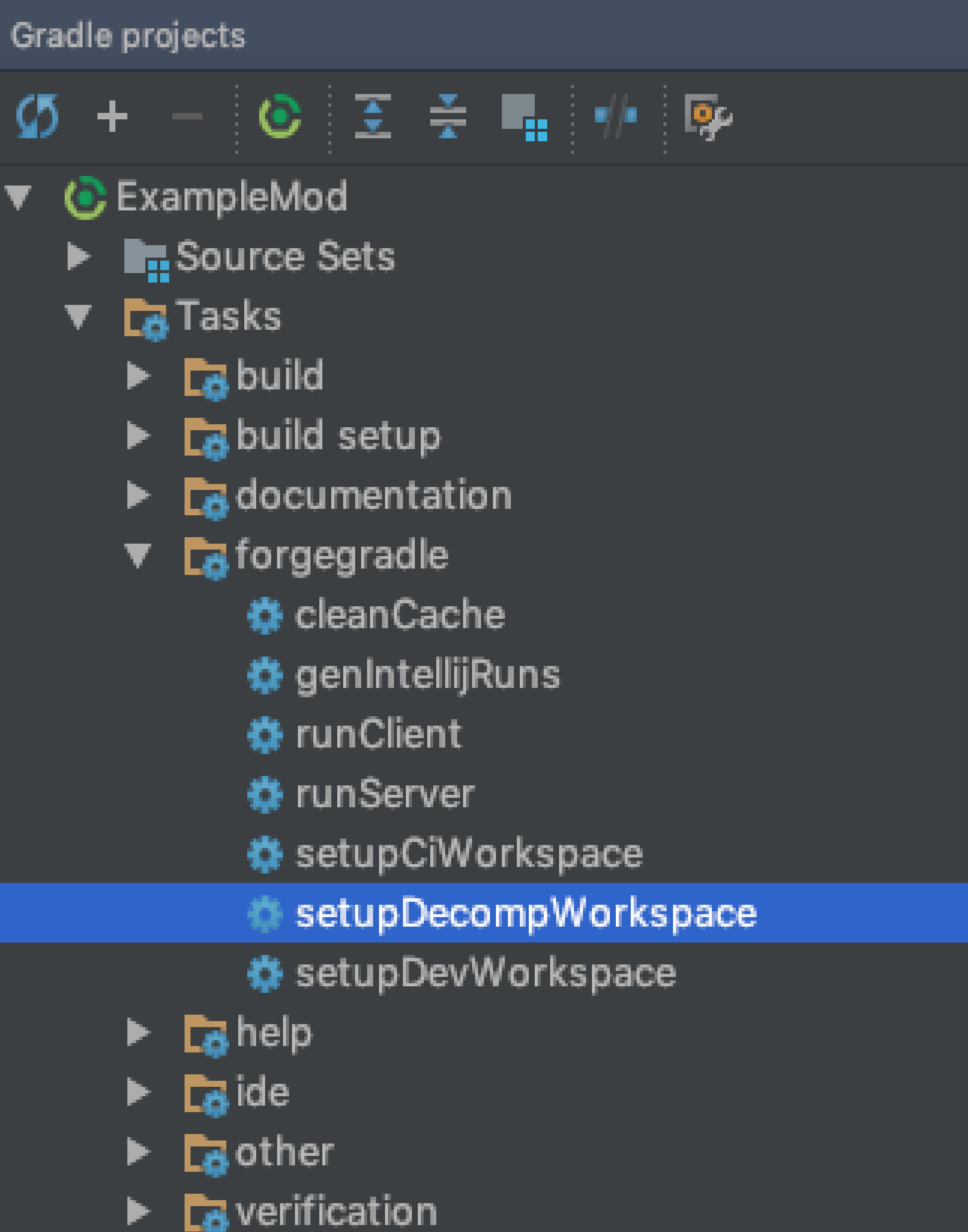
Step 4: setupDecompWorkspace

- Why setupDecompWorkspace?



Step 4: setupDecompWorkspace

- Why setupDecompWorkspace?
- setupDecompWorkspace provides comments



Step 4: setupDecompWorkspace

- Why setupDecompWorkspace?
 - setupDecompWorkspace provides comments
 - setupDevWorkspace – Relies upon IntelliJ's FernFlower decompiler, no comments.

Step 5: @Mod Annotation



```
>_ [...]
import net.minecraftforge.fml.common.Mod;
[...]

@Mod(modid="modname", name="Mod Name",
      dependencies="After:somemod;Before:SomeOtherMod")
public class ModMainClassName {
}
```

Step 5: @Mod Annotation



```
>_ [...]
import net.minecraftforge.fml.common.event.FMLPreInitializationEvent;
import org.apache.logging.log4j.Logger;
[...]
public class ModMainClassName {
    static Logger logger;

    @Mod.EventHandler
    public static void preInit(FMLPreInitializationEvent event) {
        logger = event.logger
    }
}
```

Step 5: @Mod Annotation



```
>_ [...]
import net.minecraftforge.fml.common.event.FMLInitializationEvent;
[...]

public class ModMainClassName {
    @Mod.EventHandler
    public static void init(FMLInitializationEvent event) {
        // Register ore dictionary items here, perform what cross mod functions are
        // possible.
    }
}
```

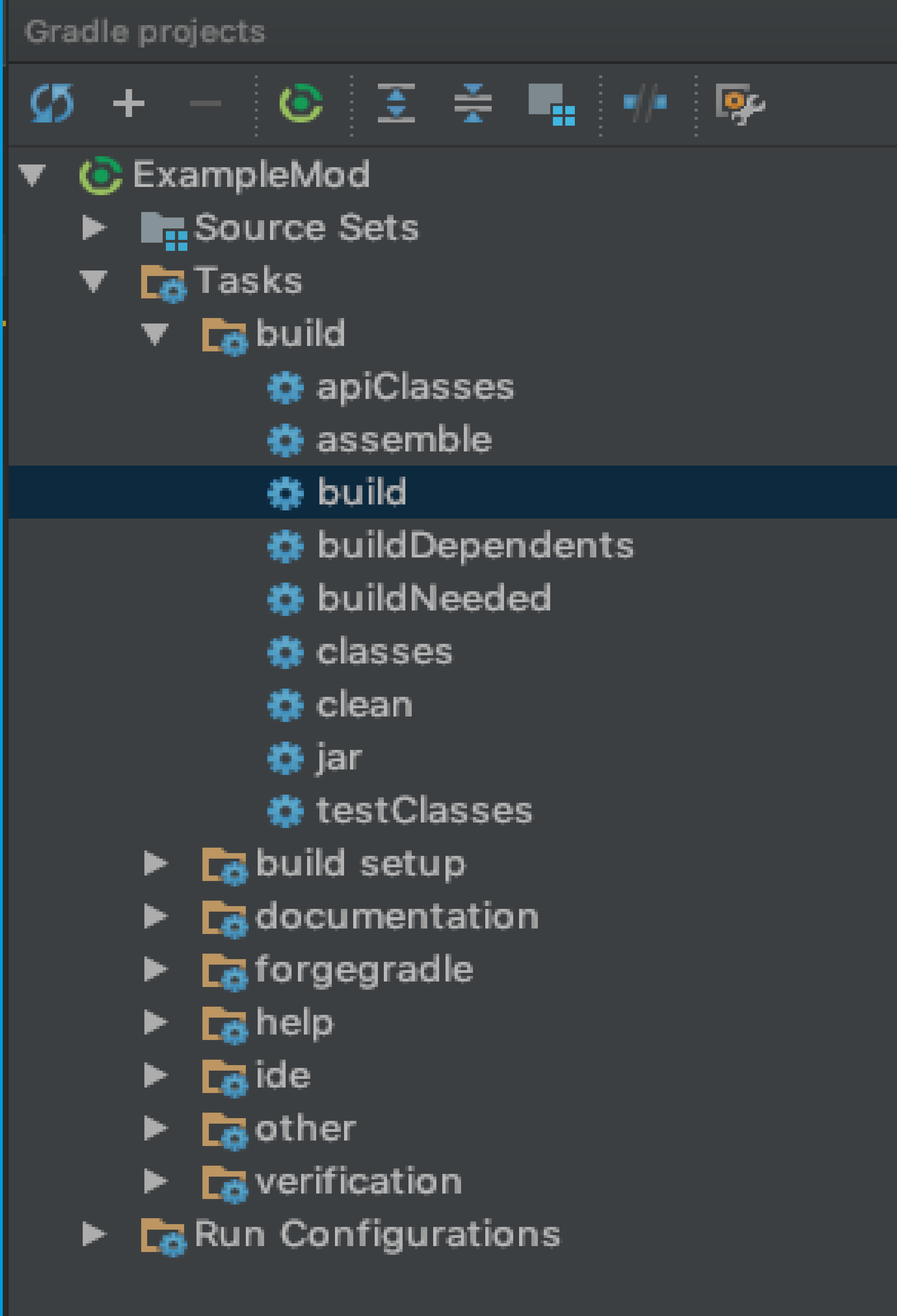

Step 5: @Mod Annotation



```
>_ [...]
import net.minecraftforge.fml.common.event.FMLPostInitializationEvent;
[...]

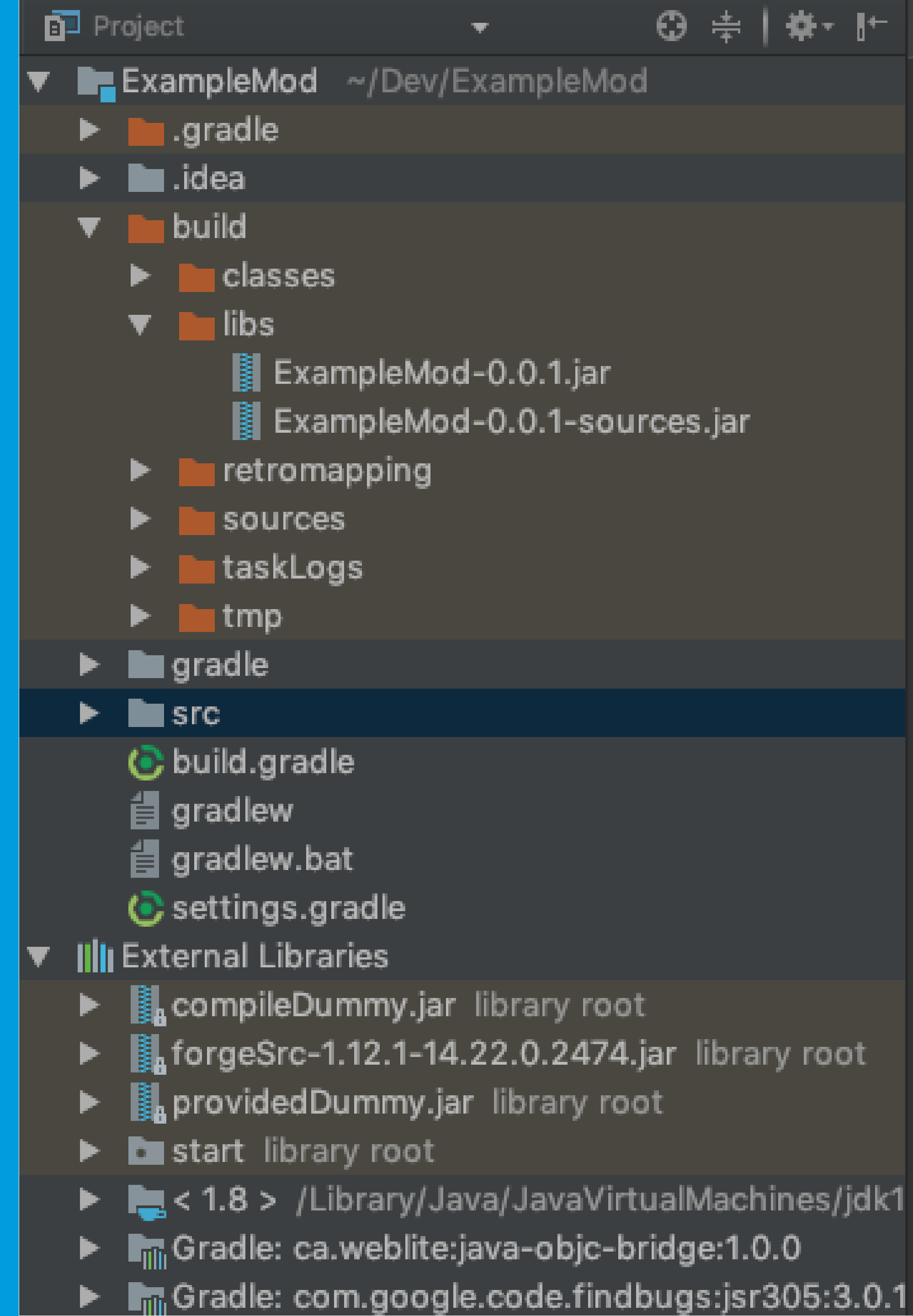
public class ModMainClassName {
    @Mod.EventHandler
    public static void postInit(FMLPostInitializationEvent event) {
        // Perform cross-mod interaction actions here.
        // If possible, preferable to do earlier. Not possible with all mods though.
        // Events are sorted by @Mod(dependencies="").
    }
}
```

Step 6: Build



Step 6: Build

- Output: \$projectDir/build/libs/



What's Next?

- Simple: Items, Blocks, Tile Entities
- Advanced: Mobs, Custom Models, X-Mod, etc.

Items



```
>_ [...]
import net.minecraft.item.Item;
[...]
public class ExampleItem extends Item {
}
```

Items



```
>_ [...]
import net.minecraft.creativetab.CreativeTabs;
[...]
public class ExampleItem extends Item {
    public ExampleItem() {
        setCreativeTab(CreativeTabs.MISC);
    }
}
```

Items



```
>_ [...]
public class ExampleItem extends Item {
    public ExampleItem() {
        setMaxStackSize(16);
    }
}
```

Items



```
>_ [...]
public class ExampleItem extends Item {
    public ExampleItem() {
        setHasSubtypes(true);
    }
}
```


Items



```
>_ [...]
public class ExampleItem extends Item {
    public ExampleItem() {
        setMaxDamage(1280);
    }
}
```

Items



```
>_ [...]
public class ExampleItem extends Item {
    public ExampleItem() {
        setUnlocalizedName("example_item");
    }
}
```

Items



```
>_ [...]
public class ExampleItem extends Item {
    public ExampleItem() {
        setRegistryName("modname_exampleitem");
    }
}
```

Items



```
>_ [...]
public class ExampleItem extends Item {
    @Override
    public ActionResult<ItemStack> onItemRightClick(World world, EntityPlayer player,
        EnumHand hand) {
        return super.onItemRightClick(world, player, hand);
    }
}
```

Items



```
>_ [...]
public class ExampleItem extends Item {
    @Override
    public boolean hasEffect(ItemStack stack) {
        return true;
    }
}
```



Items



```
>_ [...]
public class ExampleItem extends Item {
    @Override
    public void addInformation(ItemStack stack, World world, List<String> tooltip,
        ITooltipFlag flag) {
        tooltip.add("Spooky scary skeletons");
    }
}
```



Items



```
>_ [...]
import net.minecraftforge.fml.common.eventhandler.SubscribeEvent;
[...]
@Mod.EventBusSubscriber(modid="modname")
public class ExampleModRegistration {
    Item exampleItem = new ExampleItem();
    @SubscribeEvent
    public static void registerItems(RegistryEvent.Register<Item> event) {
        IForgeRegistry<Item> registry = event.getRegistry();
        registry.register(exampleItem);
    }
}
```

Item Textures



```
>_ {  
  "parent": "item/generated",  
  "textures": {  
    "layer0": "modname:items/exampleItem"  
  }  
}
```


Item Textures



>_ Json goes in `$projectDir/src/main/resources/assets/$modname/models/item/`
Texture goes in `$projectDir/src/main/resources/assets/$modname/assets/item/`

Block



```
>_ [...]
import net.minecraft.block.Block;
[...]

public class ExampleBlock extends Block {
    // setCreativeTab, setRegistryName, setUnlocalized name usage like Item
}
```

Block



```
>_ [...]
import net.minecraft.block.material.Material;
[...]

public class ExampleBlock extends Block {
    public ExampleBlock() {
        super(Material.ROCK);
    }
}
```

Block



```
>_ [...]
public class ExampleBlock extends Block {
    @Override
    public boolean isFullBlock(IBlockState state) {
        return true;
    }
}
```

Block



```
>_ [...]
public class ExampleBlock extends Block {
    @Override
    public boolean isOpaqueCube(IBlockState state) {
        return true;
    }
}
```

Block



```
>_ [...]
@Mod.EventBusSubscriber(modid="modname")
public class ExampleModRegistration {
    Block exampleBlock = new ExampleBlock();
    Item itemBlockExampleBlock = new ItemBlock(exampleBlock);
    @SubscribeEvent
    public static void registerItems(RegistryEvent.Register<Item> event) {
        IForgeRegistry<Item> registry = event.getRegistry();
        registry.register(itemBlockExampleBlock);
    }

    public static void registerItems(RegistryEvent.Register<Block> event) {
        IForgeRegistry<Block> registry = event.getRegistry();
        registry.register(exampleBlock);
    }
}
```

Block Textures



```
>_ {
  "forge_marker": 1,
  "defaults": {
    "textures": {
      "all": "modname:blocks/exampleBlock"
    }
  },
  "variants": {
    "normal": {
      "model": "cube_all"
    },
    "inventory": {
      "model": "cube_all"
    }
  }
}
```

Block Textures



>_ Model goes in `$projectDir/src/main/resources/assets/$modname/blockstates/`
Texture goes in `$projectDir/src/main/resources/assets/$modname/blocks/`

Tile Entity



```
>_ [...]
import net.minecraft.tileentity.TileEntity;
[...]
public class ExampleTile extends TileEntity {
}
```

Tile Entity



```
>_ [...]
public class ExampleTile extends TileEntity {
    public NBTTagCompound writeToNBT(NBTTagCompound compound) {
        // Write data that needs persistent storage, synchronization.
        return super(compound);
    }

    public void readFromNBT(NBTTagCompound compound) {
        // Read data stored or synchronized by writeToNBT.
        super(compound);
    }
}
```

Tile Entity



```
>_ [...]
import net.minecraft.util.ITickable;
[...]
public class ExampleTile extends TileEntity implements ITickable {
    public void update() {
        // Perform your once-per-tick tasks here. Minecraft ticks 20 times per second
        // (assuming said ticks can be completed in time)
    }
}
```

Tile Entity



```
>_ [...]
public class ExampleBlock extends Block implements ITileEntityProvider {
    public TileEntity createNewTileEntity(World world, int meta) {
        return new ExampleTile();
    }
}
```

Tile Entity



```
>_ [...]
public class ExampleMod {
    @EventHandler
    public static void preInit(FMLPreInitializationEvent event) {
        GameRegistry.register(ExampleTile.class, "modname_exampletile");
    }
}
```

Recipes



```
>_ {
  "conditions": [{
    "type": "item_exists",
    "item": "examplemod:exampleitem"
  }],
  "type": "minecraft:crafting_shaped",
  "pattern": [
    "###",
    "#D#",
    "###"
  ],
  "key": {
    "D": { "item": "minecraft:dirt" }
  },
  "result": { "item": "examplemod:exampleitem" }
}
```

Recipes



>_ Recipes are stored in `$projectDir/src/main/resources/assets/$modName/recipes/`

Language Files



```
>_ # Language files stored in $projectDir/src/main/resources/assets/$modname/lang/  
  
item.modname.exampleitem=Example Item  
block.modname.exampleblock=Example Block
```


Additional Resources

<https://mcforge.readthedocs.io/en/latest/>

Any Questions?