

Welcome and Setup

Erstellen einer eigenen Mod-Bibliothek

Einrichten der Entwicklungsumgebung

Sichern des Mods in einem Git-Repository

Erstellen einer eigenen Mod-Bibliothek



- Template mod generator
<https://fabricmc.net/develop/template/>
 - Name anpassen
 - Mod ID anpassen
 - Paket-Name anpassen
 - Alle Checkboxes bei Advanced Options abwählen
- Download und auspacken
- Starten mit gradle: **./gradlew runClient**

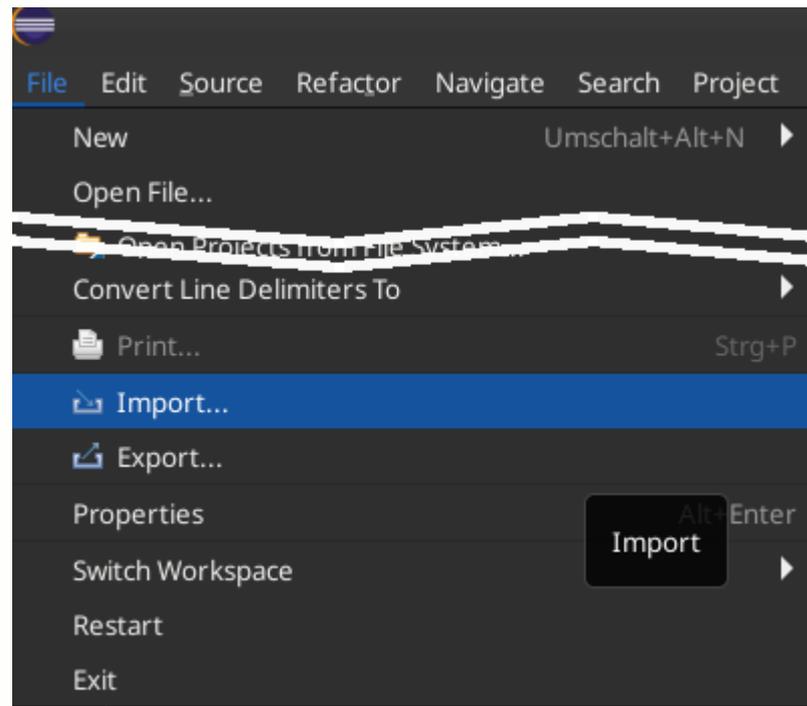
Mod in git-Ordner kopieren

- Git repository remote anlegen
- git clone ...
- Dateien in den neu geklonten Ordner kopieren
- git add .
- git commit -m "my awesome commit message"
- git push
- git pull

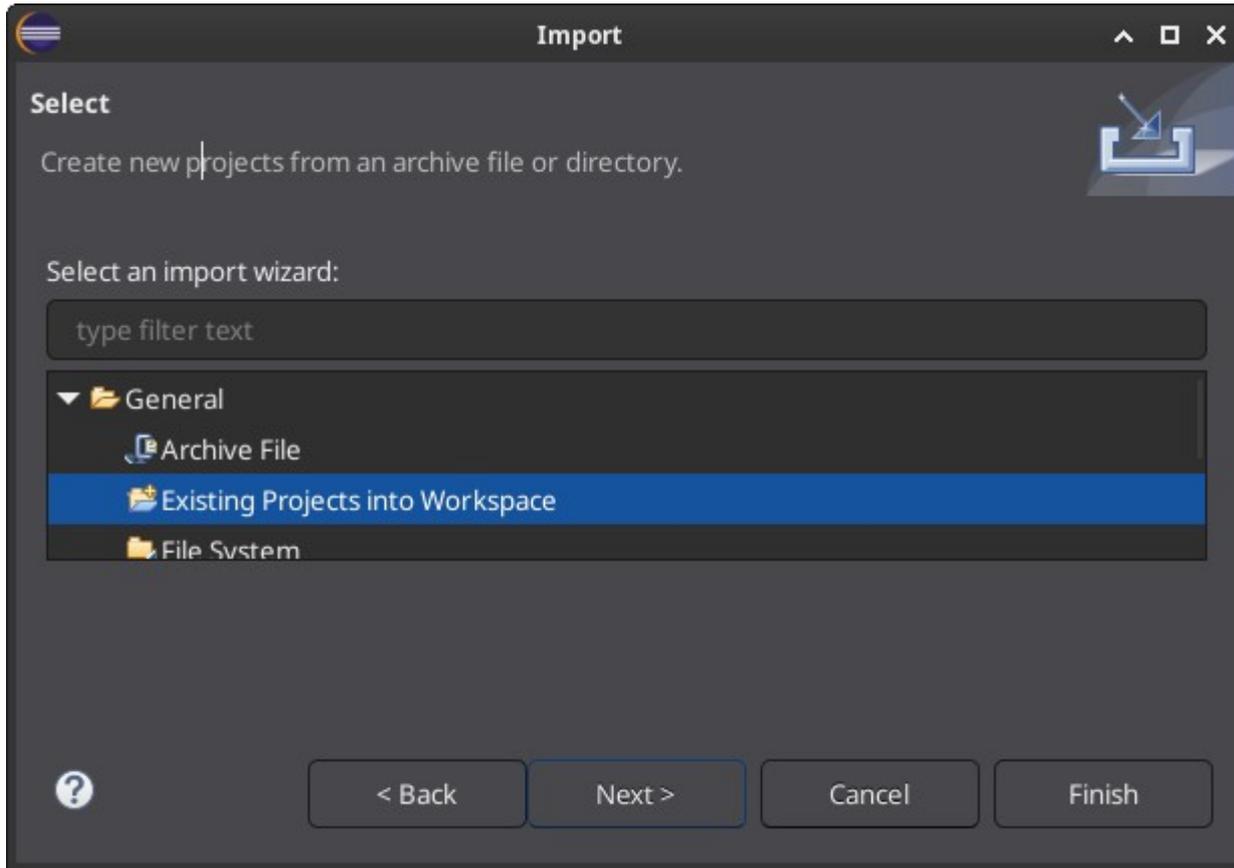


Einrichten der Entwicklungsumgebung Eclipse

- Im Terminal / Konsole eintippen:
 - **./gradlew eclipse**
 - **./gradlew genSources**
- Eclipse starten und bestehendes Projekt importieren

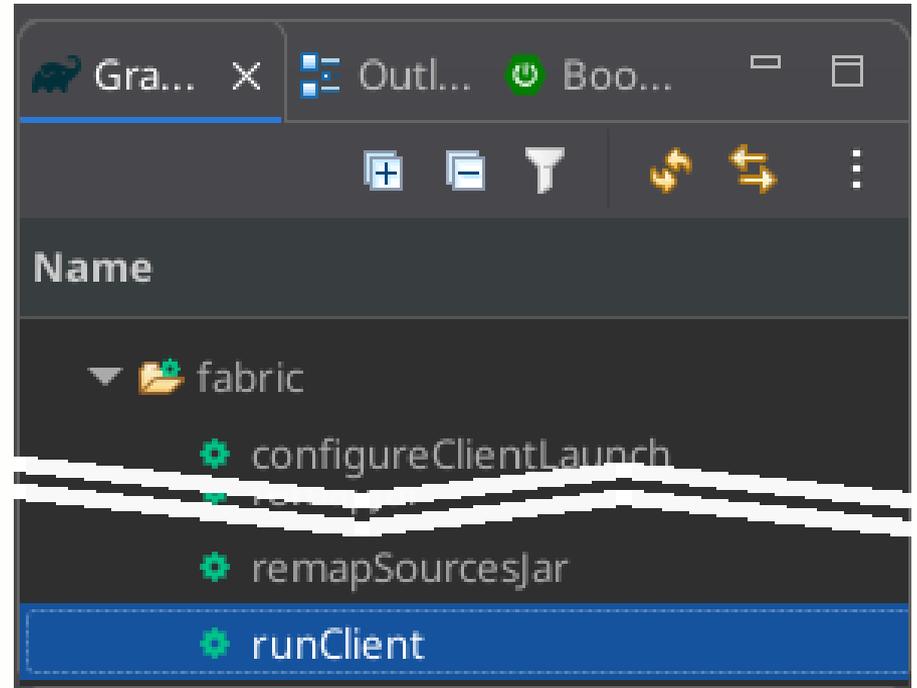


Projekt importieren



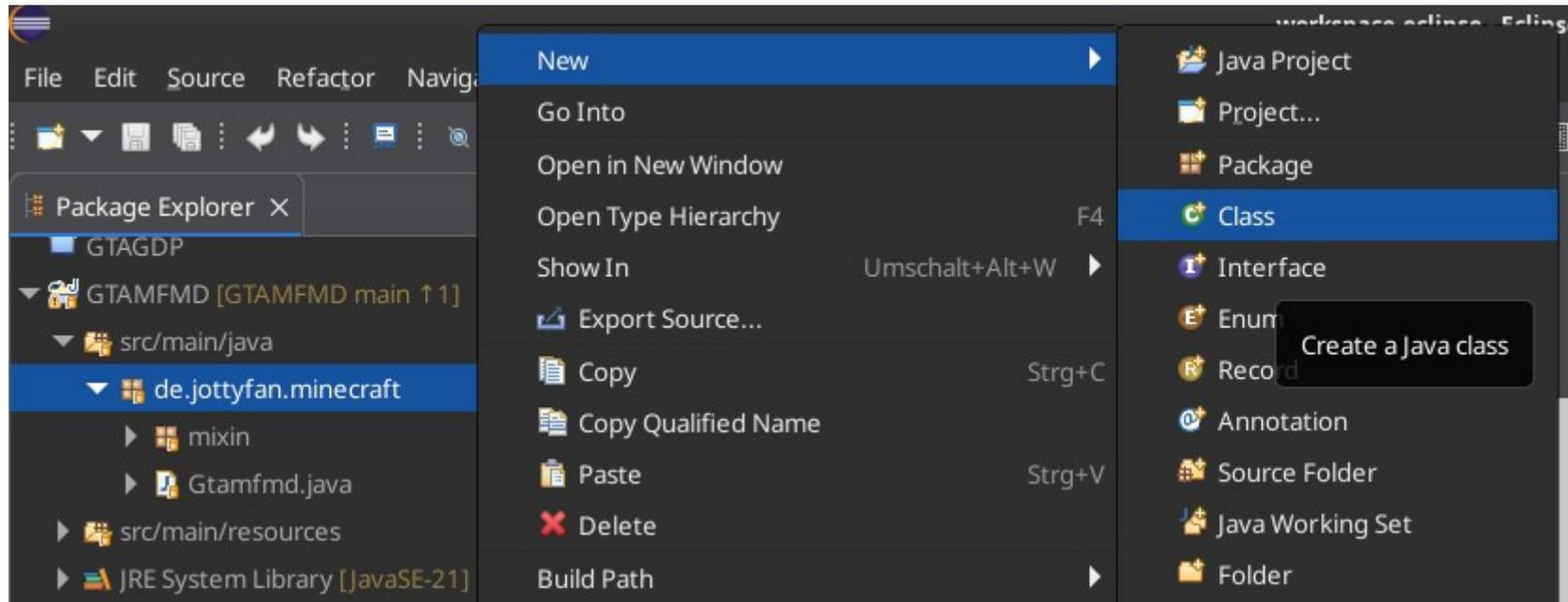
Minecraft aus Eclipse heraus starten

- Ggf. gradle nature hinzufügen
- Ggf. gradle refresh
- Dann: Starten über runClient



Mod anpassen – Grundlagen

- Client **Gtamfmd**Client anlegen



Client anlegen

- Package- und Klassenname bei Schülern anders



```
package de.jottyfan.minecraft;  
  
import net.fabricmc.api.ClientModInitializer;  
  
public class GtamfmdClient implements ClientModInitializer {  
    @Override  
    public void onInitializeClient() {  
    }  
}
```

Client registrieren

- In `src/main/resources/fabric.mod.json` ergänzen:

```
"entrypoints": {  
  "main": [  
    "de.jottyfan.minecraft.Gtamfmd"  
  ],  
  "client": [  
    "de.jottyfan.minecraft.GtamfmdClient"  
  ]  
},
```

Änderungen mit git absichern



- `git add .`
- `git commit -m "my awesome commit message"`
- `git push`
- `git pull`

Oder in Eclipse:

- Rechtsklick auf Projekt, dann Team, dann Commit... und dann:
 - Commit und Push



This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License. It makes use of the works of Mateus Machado Luna.

