



Welcome and Setup

Erstellen einer eigenen Mod-Bibliothek

Einrichten der Entwicklungsumgebung

Sichern des Mods in einem Git-Repository

Ein erstes eigenes Item

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**

Einrichten der Entwicklungsumgebung



- Am Beispiel von Eclipse
 - **./gradlew eclipse**
 - **./gradlew genSources**
- Eclipse starten und bestehendes Projekt importieren
- Projekt in Eclipse starten (runClient)
- Projekt anpassen
 - Client anlegen
 - Client registrieren

Client anlegen

- Neue Klasse hinzufügen

```
package de.jottyfan.gta.dgp;

import net.fabricmc.api.ClientModInitializer;

/**
 * @author jotty
 */
public class GTAGDPClient implements ClientModInitializer {
    @Override
    public void onInitializeClient() {
    }
}
```

Client registrieren



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

```
...  
"entrypoints": {  
  ...  
  "client": [  
    "de.jottyfan.gta.gdp.GTAGDPCClient"  
  ]  
}
```

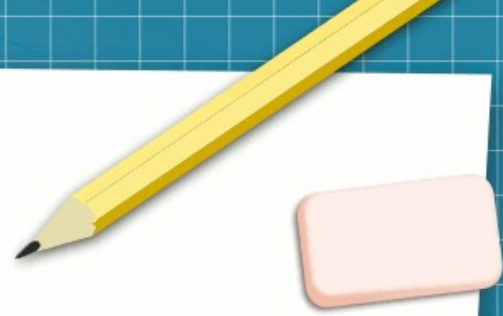

Sichern des Mods in einem Git-Repository



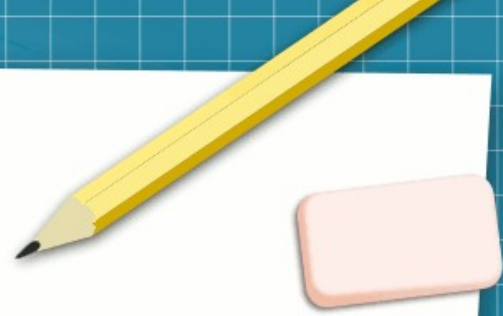
- Git repository remote anlegen
- git clone ...
- Dateien in den neu geklonten Ordner kopieren
- git add .
- git commit -m "my awesome commit message"
- git push
- git pull

Ein erstes eigenes Item

- Neues Paket **item**, darin neue Klasse **ModItems**
 - Item anlegen
 - Item registrieren
 - Registrierung einbinden



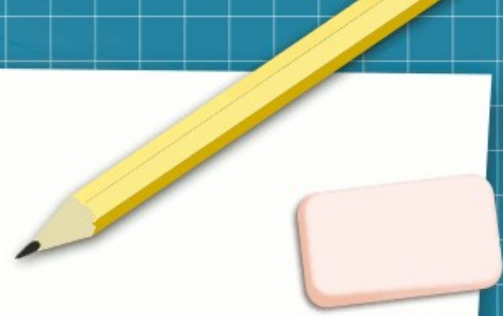
Item anlegen



- In de.jottyfan.gta.gdp.item.ModItems

```
public static final Item STUB = registerItem(  
    Identifier.of(GTAGDP.MOD_ID, "stub"), new Item.Settings());
```

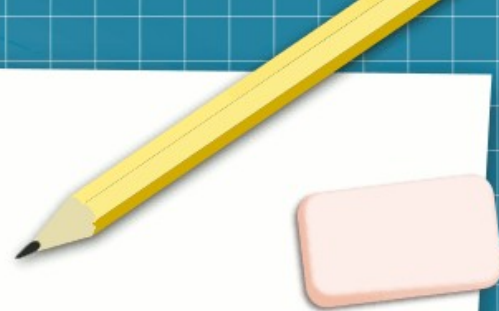

Item registrieren



- In `de.jottifyan.gta.gdp.item.ModItems`

```
private static Item registerItem(Identifier identifier, Settings settings) {  
    return Registry.register(Registries.ITEM, identifier,  
        new Item(settings.useItemPrefixedTranslationKey()  
            .registryKey(RegistryKey.of(RegistryKeys.ITEM, identifier))));  
}
```

Registrierung einbinden

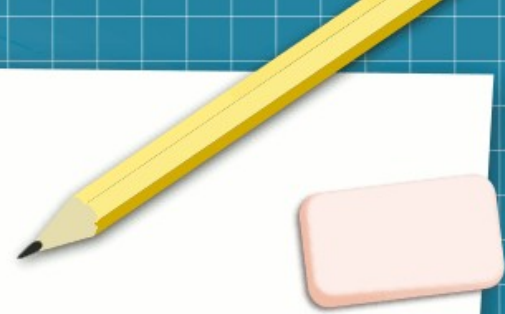


- In de.jottifyan.gta.gdp.item.ModItems

```
public static void registerModItems() {
    GTAGDP.LOGGER.info("registering mod items for " + GTAGDP.MOD_ID);

    ItemGroupEvents.modifyEntriesEvent(ItemGroups.INGREDIENTS)
        .register(entries -> {
            entries.add(STUB);
        });
}
```

In GTAGDP.onInitialize()



```
@Override  
public void onInitialize() {  
    ModItems.registerModItems();  
}
```

In
src/main/resources/assets/gtagdp/items/stub.json

```
{  
  "model": {  
    "type": "minecraft:model",  
    "model": "gtagdp:item/stub"  
  }  
}
```

In

src/main/resources/assets/gtagdp/lang/de_de.json

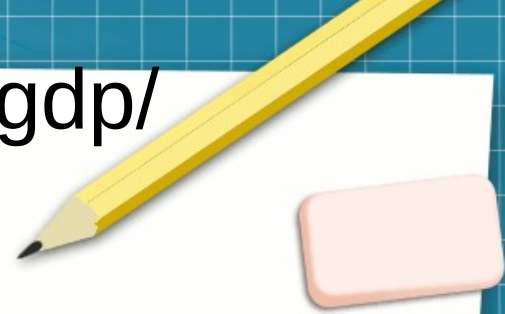
```
{  
  "item.gtagdp.stub": "Stumme1"  
}
```

In
src/main/resources/assets/gtagdp/lang/en_us.json



```
{  
    "item.gtagdp.stub": "Stub"  
}
```

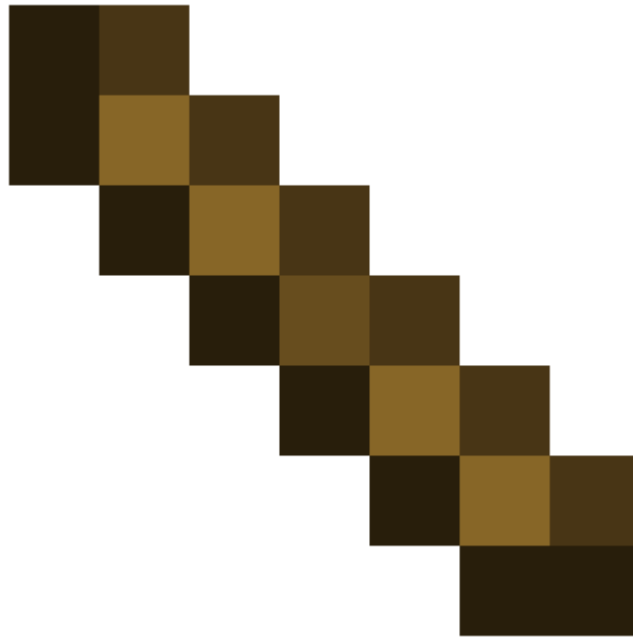

In src/main/resources/assets/gtagdp/
models/item/stub.json



```
{  
  "parent": "minecraft:item/generated",  
  "textures": {  
    "layer0": "gtagdp:item/stub"  
  }  
}
```

Textur anlegen

- In `src/main/resources/assets/gtagdp/textures/item/stub.png`



Erstelle ein 2. Item und binde es ein

- Git commit und push – erst dann mit neuen Features starten
- Namen überlegen
- Item-Bild malen
- In ModlItems einbetten
- Testen
- Git 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.

