

code

Vsebina `settings.gradle.kts` :

```
rootProject.name = "work-logger"
```

Vsebina `build.gradle.kts` :

```
plugins {
    id("java")
}

group = "org.example"
version = "1.0-SNAPSHOT"

repositories {
    mavenCentral()
}

dependencies {
    testImplementation(platform("org.junit:junit-bom:5.9.1"))
    testImplementation("org.junit.jupiter:junit-jupiter")
    implementation("net.dv8tion:JDA:5.0.0-beta.24")
    implementation("ch.qos.logback:logback-classic:1.4.14")
    implementation("com.google.code.gson:gson:2.10.1")
    implementation("com.moandjiezana.toml:toml4j:0.7.2")
}

tasks.test {
    useJUnitPlatform()
}

tasks.register<Jar>("fatJar") {
    group = "build"
    archiveBaseName.set("work-logger")
    archiveClassifier.set("all")
    duplicatesStrategy = DuplicatesStrategy.EXCLUDE
    manifest {
        attributes("Main-Class" to "io.github.andrejfox.worklogger.Main")
    }

    from(configurations.runtimeClasspath.get().map {
        if (it.isDirectory) {
            it
        } else {
            zipTree(it)
        }
    })

    with(tasks.jar.get())
}
```

Vsebina `main.java` :

```
package io.github.andrejfox.worklogger;

import io.github.andrejfox.worklogger.commands.AddCommand;
import io.github.andrejfox.worklogger.commands.MailCommand;
import io.github.andrejfox.worklogger.commands.PayCommand;
import io.github.andrejfox.worklogger.commands.RmCommand;
import net.dv8tion.jda.api.JDA;
import net.dv8tion.jda.api.JDABuilder;
import net.dv8tion.jda.api.events.session.ReadyEvent;
import net.dv8tion.jda.api.hooks.ListenerAdapter;
import net.dv8tion.jda.api.requests.GatewayIntent;
import org.jetbrains.annotations.NotNull;

public class Main {
    public static JDA api;
    public static void main(String[] args) {
        Util.createDefaultIfNotExists("mail.txt", "/default-mail.txt");
        Util.createDefaultIfNotExists("config.toml", "/default-config.toml");
        Util.loadConfig();
        Util.loadTagOrder();

        api = JDABuilder.createDefault(Util.CONFIG.botToken())
            .enableIntents(
                GatewayIntent.GUILD_MEMBERS,
                GatewayIntent.GUILD_MESSAGE_REACTIONS,
                GatewayIntent.GUILD_MESSAGES
            )
            .addEventListeners(
                new AddCommand(),
                new MailCommand(),
                new RmCommand(),
                new PayCommand()
            )
            .build();

        api.updateCommands().addCommands(
            AddCommand.register(),
            MailCommand.register(),
            RmCommand.register(),
            PayCommand.register()
        ).queue();

        api.addListener(new ListenerAdapter() {
            @Override
            public void onReady(@NotNull ReadyEvent event) {
                Util.updateNotPayedBoard();
            }
        });

        //Used to add freshly added files to not paid list
        //Util.refreshToNotPayedList();
    }
}
```