Co musi być w głównej klasie pluginu Bukkit/Spigot, żeby config działał?

# 1. Dziedziczenie po JavaPlugin

Twoja główna klasa pluginu MUSI dziedziczyć po JavaPlugin:

public class MyPlugin extends JavaPlugin {

# 2. Tworzenie i inicjalizacja ConfigManager

W klasie głównej tworzysz obiekt ConfigManager i go inicjalizujesz w onEnable():

private ConfigManager configManager;  
  
@Override  
public void onEnable() {  
 saveDefaultConfig();  
 this.configManager = new ConfigManager(this);  
  
 // Testowy odczyt wartości z configa  
 getLogger().info("Enabled: " + configManager.getBoolean("settings.enabled"));  
}

# 3. saveDefaultConfig()

Użyj saveDefaultConfig() w onEnable(), żeby utworzyć config.yml jeśli jeszcze nie istnieje.

# 4. Struktura plugin.yml

Upewnij się, że masz poprawną ścieżkę do klasy głównej w plugin.yml:

name: MojPlugin  
version: 1.0  
main: me.twojplugin.MyPlugin  
api-version: 1.20

# 5. Użycie ConfigManager w innych klasach (opcjonalne)

Możesz przekazywać ConfigManager do listenerów i innych klas, np.:

public class MojListener implements Listener {  
  
 private final ConfigManager config;  
  
 public MojListener(ConfigManager config) {  
 this.config = config;  
 }  
  
 @EventHandler  
 public void onJoin(PlayerJoinEvent event) {  
 String msg = config.getString("messages.welcome");  
 event.getPlayer().sendMessage(ChatColor.translateAlternateColorCodes('&', msg));  
 }  
}

Rejestracja w klasie głównej:

getServer().getPluginManager().registerEvents(new MojListener(configManager), this);

# 6. Komenda do przeładowania configa (opcjonalnie)

if (cmd.getName().equalsIgnoreCase("reloadconfig")) {  
 configManager.reload();  
 sender.sendMessage("Config przeładowany!");  
 return true;  
}