

kumuluzee-examples (0.0.1)

Maksim Kostromin

Version 0.0.1, 2019-05-04 00:34:12 UTC

Table of Contents

1. Introduction	1
2. Implementation	2
2.1. KumuluzEE using Gradle (Kotlin DSL)	2
3. Links	4

Chapter 1. Introduction

Read [project reference documentation](#)

build

```
./mvnw
```

docs

```
./gradlew documentation -Dorg.gradle.jvmargs='-Xmx1g'
```

TODO (Gradle):

- [KumuluzEE with Gradle 1](#)
- [KumuluzEE with Gradle 2](#)
- [KumuluzEE with Gradle 3](#)

Chapter 2. Implementation

2.1. KumuluzEE using Gradle (Kotlin DSL)

```
val kumuluzeeLoader by configurations.creating {
    extendsFrom(configurations["runtime"])
    setTransitive(false)
}
dependencies {
    kumuluzeeLoader("com.kumuluz.ee:kumuluzee-loader:$kumuluzeeVersion")
    // ...
}

tasks {
    register("kumuluzLoader", Copy::class.java) {
        group = "KumuluzEE"
        description = "Repackage KumuluzEE loader"
        from(zipTree(configurations["kumuluzeeLoader"].singleFile).matching {
            include("**/*.class") })
        into("$buildDir/classes/java/main")
        //outputs.upToDateWhen { false }
    }

    register("kumuluzProperties") {
        group = "KumuluzEE"
        description = "Add KumuluzEE boot-loader.properties file"
        val parent = "$buildDir/classes/java/main/META-INF/kumuluzee"
        file(parent).mkdirs()
        file("$parent/boot-loader.properties")
            .writeText("main-class=com.kumuluz.ee.EeApplication\n", UTF_8)
            //.appendText("\n", UTF_8)
    }

    register("kumuluzJar", Jar::class.java) {
        group = "KumuluzEE"
        description = "Build KumuluzEE uber (fat) jar"

        archiveClassifier.set("all")
        duplicatesStrategy = DuplicatesStrategy.EXCLUDE

        manifest {
            attributes("Main-Class" to "com.kumuluz.ee.loader.EeBootLoader")
        }

        // deps
        val libs = configurations.runtimeClasspath
        //libs.get().forEach { println("add dependency: ${it.name}") }
        from(libs.get()) {
            into("lib")
        }
    }
}
```

```
}

// app
val sourcesMain = sourceSets.main
//sourcesMain.get().allSource.forEach { println("add app: ${it.name}") }
from(sourcesMain.get().output)
//from(configurations["kumuluzeeLoader"])

shouldRunAfter("kumuluzLoader", "kumuluzProperties")
dependsOn("kumuluzLoader", "kumuluzProperties")
finalizedBy("build")
}
}

defaultTasks("kumuluzJar")
```

Chapter 3. Links

- [GitHub repo](#)
- [GitHub pages](#)