spring-boot-kotlin-blog (0.0.1)

Maksim Kostromin

Version 0.0.1, 2018-07-17 04:51:38 UTC

Table of Contents

Travis CI status: [Build Status]

Chapter 1. Introduction

Read project reference documentation

Initially generated by using generator-jvm yeoman generator (kotlin-spring-boot)

Chapter 2. Implementation

webflux REST API

```
@Configuration
class WebFluxRest {

  @Bean
  fun restRoutes() = router {

    ("/").nest {

      contentType(APPLICATION_JSON_UTF8)

      GET("/api/**") {
        val path = it.uri().toURL().path
        val parts = path.split("/")
        val name = if (parts.isEmpty()) "buddy" else parts.last()
        val params = it.queryParams().toSingleValueMap().toMutableMap()
            .filterNot { it.value.isNullOrEmpty() or it.value.isNullOrBlank() }
        val result = if (params.isEmpty()) mapOf("hello" to name) else params
        val publisher = result.toMono()
        ok().body(publisher)
      }
    }
  }
}
```

web MVC

```
@Configuration
@EnableConfigurationProperties(ThymeleafProperties::class)
class ThymeleafConfig(val engine: ISpringWebFluxTemplateEngine) : WebFluxConfigurer {

  @Bean
  fun thymeleafChunkedAndDataDrivenViewResolver(): ThymeleafReactiveViewResolver {
    val viewResolver = ThymeleafReactiveViewResolver()
    viewResolver.responseMaxChunkSizeBytes = 8192
    viewResolver.templateEngine = engine
    viewResolver.order = 1
    return viewResolver
  }

  override fun configureViewResolvers(registry: ViewResolverRegistry) {
    registry.viewResolver(thymeleafChunkedAndDataDrivenViewResolver())
  }
}
```

webflux MVC

```
@Configuration
class WebFluxRest {

  @Bean
  fun restRoutes() = router {

    ("/").nest {

      contentType(APPLICATION_JSON_UTF8)

      GET("/api/**") {
        val path = it.uri().toURL().path
        val parts = path.split("/")
        val name = if (parts.isEmpty()) "buddy" else parts.last()
        val params = it.queryParams().toSingleValueMap().toMutableMap()
            .filterNot { it.value.isNullOrEmpty() or it.value.isNullOrBlank() }
        val result = if (params.isEmpty()) mapOf("hello" to name) else params
        val publisher = result.toMono()
        ok().body(publisher)
      }
    }
  }
}
```

Chapter 3. Build, test and run

gradle

```
./gradlew
java -jar build/libs/*.jar
bash build/libs/*.jar

./gradlew build composeUp
./gradlew composeDown
```

maven

```
./mvnw
java -jar target/*.jar
bash target/*.jar

bash mvnw com.dkanejs.maven.plugins:docker-compose-maven-plugin:1.0.1:up -P docker
bash mvnw com.dkanejs.maven.plugins:docker-compose-maven-plugin:1.0.1:down -P docker
```

Chapter 4. Links

- GitHub repo
- GitHub pages