

Ktor "quotemanager"

Kotlin Web-Framework

Erklärungen

- Client und Serverseitig
- "Quick and Dirty"
- Datenbankbindung über JPA selber machen
- Ktor Backends (Netty, Jetty, Tomcat)

Todos

Basic Project

Hello World

```
fun Application.module(testing: Boolean = false) {  
    routing {  
        get("/") {  
            call.respondText("Hello Worlds")  
        }  
    }  
}
```

Model

src/model/Model.kt

```
data class Quote(  
    var id: Int?  
) {  
    lateinit var text: String  
    lateinit var person: Person  
}  
  
data class Person(  
    var id: Int?  
) {  
    lateinit var name: String  
}
```

src/Application.kt

```
val persons = mutableListOf(
    Person(1).apply { name="Erik Mayrhofer" },
    Person(2).apply { name="Test" }
)

val quotes = mutableListOf(
    Quote(1).apply{text="SomeQuoteTest"; person=persons[0]},
    Quote(2).apply{text="SomeQuoteTest"; person=persons[0]}
)
```

Quotes Query

build.gradle

```
compile "org.jetbrains.kotlin:kotlin-stdlib-jdk8:$kotlin_version"
```

src/Application.kt

```
install(ContentNegotiation){
    jackson {}
}
routing {
    get("/quotes") {
        call.respond(quotes)
    }
}
```

Test Queries

```
http localhost:8080/quotes/
```

Post Request

```

fun insertQuote(quote: Quote): Quote{
    quote.person = persons.single { it.id == quote.person.id }
    quote.id = (quotes.mapNotNull { it.id }.max() ?: 0) + 1
    quotes += quote
    return quote
}

...
post("/quotes"){
    val quote = call.receive<Quote>()
    insertQuote(quote)
    call.respond(quote)
}

```

Http Templating

```
implementation "io.ktor:ktor-html-builder:$ktor_version"
```

```

get("/quoteui") {
    call.respondHtml {
        body {
            h1 {
                +"Quotes"
            }
            quotes.forEach {
                ul {
                    p {
                        +"Quote: ${it.text}"
                    }
                }
            }
        }
    }
}

```