



Singleton & Prototype

*The force has
Kotlin Language
to help you in
this journey.*



```
object BooksRepository {  
    private val books = mutableListOf<Book>()  
  
    fun saveOrUpdate(book: Book) {  
        val existingBookIndex = books.indexOfFirst { it.id == book.id }  
        books.removeAt(existingBookIndex)  
        books.add(book)  
    }  
  
    fun all(): List<Book> = books  
  
    fun getBookById(id: UUID) = books.first { it.id == id }  
}
```

```
data class Book(  
    val id: UUID,  
    val numberOfPages: Int,  
    val title: String,  
    val author: String,  
)
```

Singleton,
Kotlin has



```
object BooksRepository {  
    private val books = mutableListOf<Book>()  
  
    fun saveOrUpdate(book: Book) {  
        val existingBookIndex = books.indexOfFirst { it.id == book.id }  
        books.removeAt(existingBookIndex)  
        books.add(book)  
    }  
  
    fun all(): List<Book> = books  
  
    fun getBookById(id: UUID) = books.first { it.id == id }  
}
```

*Practice we
need.*



```
fun main() {  
    val romeoAndJulietBookID = UUID.randomUUID()  
    val book = Book(  
        id = romeoAndJulietBookID,  
        numberOfPages = 129,  
        title = "Romeo and Juliet",  
        author = "William"  
    )  
  
    BooksRepository.saveOrUpdate(book)  
  
    // Ops the author name is wrong...  
    val bookFromDataBase = BooksRepository.getBookById(romeoAndJulietBookID)  
  
    val updatedBook = bookFromDataBase.copy(author = "William Shakespeare")  
  
    BooksRepository.saveOrUpdate(updatedBook)  
  
    // Fine  
}
```

Prototype,
Kotlin has



```
fun main() {  
    val romeoAndJulietBookID = UUID.randomUUID()  
    val book = Book(  
        id = romeoAndJulietBookID,  
        numberOfPages = 129,  
        title = "Romeo and Juliet",  
        author = "William"  
    )  
  
    BooksRepository.saveOrUpdate(book)  
  
    // Ops the author name is wrong...  
    val bookFromDataBase = BooksRepository.getBookById(romeoAndJulietBookID)  
  
    val updatedBook = bookFromDataBase.copy(author = "William Shakespeare")  
  
    BooksRepository.saveOrUpdate(updatedBook)  
  
    // Fine  
}
```


Prototype,
Kotlin has



```
fun main() {
    val rom...
    val book...
    id
    num
    tit
    au
}

BooksRe

// Ops the author name is wrong...
val bookFromDataBase = BooksRepository.getBookById(romeoAndJulietBookID)

val updatedBook = bookFromDataBase.copy(author = "William Shakespeare")

BooksRepository.saveOrUpdate(updatedBook)

// Fine
}
```



By Moro