

# Rapport Library Project - Scala

Juillet 2025

---

YAHIOUNE Selma

The Kuik'n'Rolls

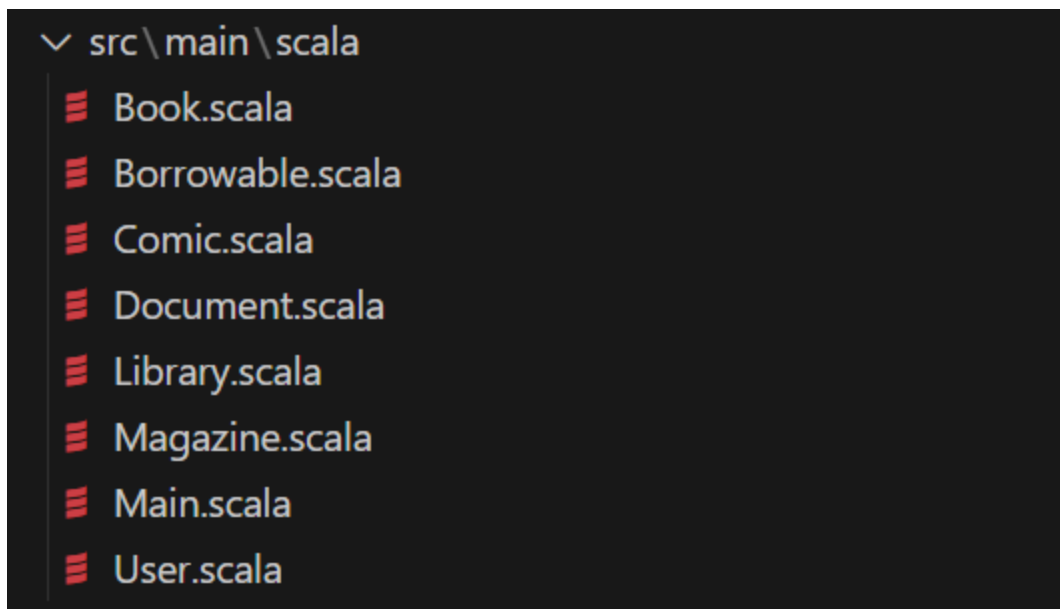
Kodemade - Kuikops

## Objectifs

Réaliser un mini projet en Scala permettant de gérer les documents d'une bibliothèque (livres, magazines, bandes dessinées), avec possibilité d'enregistrer des utilisateurs et de gérer les emprunts/retours.

## Structure

Chaque classe ou trait est déclaré dans un fichier .scala portant le même nom, afin de favoriser la lisibilité, la modularité et la maintenance du code.



## Trait "Borrowable"

```
trait Borrowable {  
  def borrow(): Boolean  
  def returnItem(): Boolean  
}
```

## Classe abstraite “Document”

### Attributs

```
abstract class Document(
  val title : String,
  val author : String,
  val year : Int,
  var isBorrowed : Boolean = false
) extends Borrowable {
```

### Méthode “description()”

Implémentation par défaut, à utiliser dans le cas de création de classe enfant qui ne contient pas de nouvelles variables à afficher.

```
def description(): String = {
  if(isBorrowed){
    s"Type: ${this.getClass.getSimpleName} | Title: $title | Author: $author | Year: $year | Status: Unavailable."
  }
  else {
    s"Type: ${this.getClass.getSimpleName} | Title: $title | Author: $author | Year: $year | Status: Available."
  }
}
```

→ this.getClass.getSimpleName permet de récupérer le nom de la classe (Document, Book, Comic etc.)

### Méthode “borrow()”

```
↑ borrow
override def borrow(): Boolean = {
  if (!isBorrowed) {
    isBorrowed=true
    true
  }
  else {
    false
  }
}
```

## Méthode “returnItem()”

```
↑returnItem
override def returnItem(): Boolean = {
  if(isBorrowed) {
    isBorrowed=false
    true
  }
  else {
    false
  }
}
```

## Classes héritant de “Document”

### Book

```
class Book(
  ↑title
  title: String,
  ↑author
  author: String,
  ↑year
  year: Int,
  val genre: String
) extends Document(title, author, year) {

  ↑description
  override def description(): String = {
    if(isBorrowed){
      s"Type: ${this.getClass.getSimpleName} | Title: $title | Author: $author | Year: $year | Genre: $genre | Status: Unavailable."
    }
    else {
      s"Type: ${this.getClass.getSimpleName} | Title: $title | Author: $author | Year: $year | Genre: $genre | Status: Available."
    }
  }
}
```

## Magazine

```
class Magazine (  
  ↑title  
  title: String,  
  ↑author  
  author: String,  
  ↑year  
  year: Int,  
  val editionNumber: String  
) extends Document (title, author, year) {  
  
  ↑description  
  override def description(): String = {  
    if(isBorrowed){  
      s"Type: ${this.getClass.getSimpleName} | Title: $title | Author: $author | Year: $year | Edition Number: $editionNumber | Status: Unavailable"  
    }  
    else {  
      s"Type: ${this.getClass.getSimpleName} | Title: $title | Author: $author | Year: $year | Edition Number: $editionNumber | Status: Available"  
    }  
  }  
}
```

## Comic

```
class Comic (  
  ↑title  
  title: String,  
  ↑author  
  author: String,  
  ↑year  
  year: Int,  
  val seriesVolume: String  
) extends Document (title, author, year) {  
  
  ↑description  
  override def description(): String = {  
    if(isBorrowed){  
      s"Type: ${this.getClass.getSimpleName} | Title: $title | Author: $author | Year: $year | Series Volume: $seriesVolume | Status: Unavailable"  
    }  
    else {  
      s"Type: ${this.getClass.getSimpleName} | Title: $title | Author: $author | Year: $year | Series Volume: $seriesVolume | Status: Available"  
    }  
  }  
}
```

## Classe User

### Attributs

```
class User(  
    val name : String,  
    var borrowedDocs : List[Document] = List()  
) {
```

### Méthode “borrowDocument()”

```
// Method to borrow a document by adding it to the list of borrowed docs  
def borrowDocument(doc : Document) : Boolean = {  
  
    if (doc.borrow()) {  
        borrowedDocs = doc :: borrowedDocs  
        println(s"\n$name has borrowed ${doc.title}")  
        true  
    }  
    else {  
        println("\nThis document is not currently available")  
        false  
    }  
}
```

### Méthode “returnDocument()”

Cette méthode vérifie d’abord que le document est bien présent dans la liste des documents empruntés par l’utilisateur, et procède le cas échéant à la suppression de ce document de cette liste.

```
// Method to return a document by removing it from the list of borrowed docs
def returnDocument(doc : Document) : Boolean = {
  println(s"\n$name wants to return ${doc.title}.")
  //Verifying that the document is, in fact, borrowed by this user
  if (borrowedDocs.contains(doc)){
    if (doc.returnItem()) {
      borrowedDocs = borrowedDocs.filterNot(_.title == doc.title)
      println(s"\n$name has returned ${doc.title}.")
      true
    }
    else {
      false
    }
  }
  else {
    println(s"\n$name can't return a document they didn't borrow.")
    false
  }
}
```

## Méthode “listBorrowedDocuments()”

Cette méthode liste tous les documents qu’un utilisateur a empruntés en se basant sur la méthode “description” définie précédemment.

```
def listBorrowedDocuments(): Boolean = {
  if (borrowedDocs.isEmpty) {
    println(s"\n$name has not borrowed any documents.")
    false
  } else {
    println(s"\n$name has borrowed the following documents:")
    borrowedDocs.foreach { d => println(s"- ${d.description()}") }
    true
  }
}
```

## Classe Library

### Variables et getteurs

```
class Library () {  
  
    // List of all the documents in the library  
    var documents : List[Document] = List()  
  
    // List of the registred users  
    var users : List[User] = List()  
  
    // Getters  
    def getDocuments: List[Document] = documents.toList  
    def getUsers: List[User] = users.toList  
}
```

### Méthodes “addDocument()” et “addUser()”

Permettent d’ajouter respectivement des documents et des utilisateurs à la bibliothèque.

```
def addDocument(newDoc : Document) : Unit = {  
    // Adding doc lifo (quicker than fifo)  
    documents = newDoc :: documents  
    println(s"The document <<${newDoc.title}>> has been added.")  
}  
  
def addUser(newUser : User) : Unit = {  
    // Add user  
    users = newUser :: users  
    println(s"The user <<${newUser.name}>> has joined us !")  
}
```



## Méthode “listAvailableDocuments()”

```
def listAvailableDocuments(): Unit = {
    // Print available documents
    if (documents.isEmpty) {
        println("\nThere are no documents in the library, come back soon!")
    } else {

        val availableDocs = documents.filterNot(_.isBorrowed)

        if (availableDocs.isEmpty) {
            println("\nAll documents are currently borrowed.")
        } else {
            println("\nAvailable documents :")
            availableDocs.foreach(doc => println(s"- ${doc.description()}"))
        }
    }
}
```

## Classe Main

Classe principale servant à simuler l'application : elle instancie les classes précédentes et appelle leurs méthodes pour tester leur fonctionnement.

## Création d'exemples pour initialiser la simulation :

```
object Main extends App {
    println("#####")
    println("##### WELCOME TO OUR LIBRARY ! #####")
    println("#####\n")

    val library = new Library()

    // Creating examples of documents to work with
    val book1 = new Book("Crown of Midnight", "Sarah J. Maas", 2013, "YA Fantasy")
    val book2 = new Book("The Queen of Nothing", "Holly Black", 2019, "YA Fantasy")
    val book3 = new Book("The Assassin's Blade", "Sarah J. Maas", 2014, "YA Fantasy")
    val mag1 = new Magazine("Game Informer", "Game Studios", 2024, "April Edition")
    val comic1 = new Comic("One Piece", "Eiichiro Oda", 2024, "Volume 102")

    val user1 = new User("Selma Yahioune")
    val user2 = new User("Celaena Sardothian")
    val user3 = new User("Jude Duarte")
    val user4 = new User("Mikasa Ackerman")
}
```

Ajout des documents et utilisateurs à la bibliothèque :

```
// Adding the documents to the library
library.addDocument(book1)
library.addDocument(book2)
library.addDocument(book3)
library.addDocument(mag1)
library.addDocument(comic1)

// Adding the users
library.addUser(user1)
library.addUser(user2)
library.addUser(user3)
library.addUser(user4)
```

Appel des différentes méthodes :

```
// A little simulation
library.listAvailableDocuments()
library.users(0).borrowDocument(book1)
library.users(0).borrowDocument(book2)
library.users(0).listBorrowedDocuments()
library.users(2).returnDocument(book3)
library.listAvailableDocuments()
library.users(2).borrowDocument(book3)
library.users(1).borrowDocument(book3)
library.users(0).returnDocument(book1)
library.users(0).listBorrowedDocuments()
library.listAvailableDocuments()

println("\n#####")
println("##### HOPE WE'LL SEE YOU SOON ! #####")
println("#####")
```

## Exécution

```
#####
##### WELCOME TO OUR LIBRARY ! #####
#####

The document <<Crown of Midnight>> has been added.
The document <<The Queen of Nothing>> has been added.
The document <<The Assassin's Blade>> has been added.
The document <<Game Informer>> has been added.
The document <<One Piece>> has been added.
The user <<Selma Yahioune>> has joined us !
The user <<Celaena Sardothian>> has joined us !
The user <<Jude Duarte>> has joined us !
The user <<Mikasa Ackerman>> has joined us !
```

```
Available documents :
- Type: Comic | Title: One Piece | Author: Eiichiro Oda | Year: 2024 | Series Volume: Volume 102 | Status: Available.
- Type: Magazine | Title: Game Informer | Author: Game Studios | Year: 2024 | Edition Number: April Edition | Status: Available.
- Type: Book | Title: The Assassin's Blade | Author: Sarah J. Maas | Year: 2014 | Genre: YA Fantasy | Status: Available.
- Type: Book | Title: The Queen of Nothing | Author: Holly Black | Year: 2019 | Genre: YA Fantasy | Status: Available.
- Type: Book | Title: Crown of Midnight | Author: Sarah J. Maas | Year: 2013 | Genre: YA Fantasy | Status: Available.

Mikasa Ackerman has borrowed Crown of Midnight

Mikasa Ackerman has borrowed The Queen of Nothing

Mikasa Ackerman has borrowed the following documents:
- Type: Book | Title: The Queen of Nothing | Author: Holly Black | Year: 2019 | Genre: YA Fantasy | Status: Unavailable.
- Type: Book | Title: Crown of Midnight | Author: Sarah J. Maas | Year: 2013 | Genre: YA Fantasy | Status: Unavailable.

Celaena Sardothian wants to return The Assassin's Blade.

Celaena Sardothian can't return a document they didn't borrow.
```

```
Available documents :
- Type: Comic | Title: One Piece | Author: Eiichiro Oda | Year: 2024 | Series Volume: Volume 102 | Status: Available.
- Type: Magazine | Title: Game Informer | Author: Game Studios | Year: 2024 | Edition Number: April Edition | Status: Available.
- Type: Book | Title: The Assassin's Blade | Author: Sarah J. Maas | Year: 2014 | Genre: YA Fantasy | Status: Available.

Celaena Sardothian has borrowed The Assassin's Blade

The document The Assassin's Blade is not currently available.

Mikasa Ackerman wants to return Crown of Midnight.

Mikasa Ackerman has returned Crown of Midnight.

Mikasa Ackerman has borrowed the following documents:
- Type: Book | Title: The Queen of Nothing | Author: Holly Black | Year: 2019 | Genre: YA Fantasy | Status: Unavailable.

Available documents :
- Type: Comic | Title: One Piece | Author: Eiichiro Oda | Year: 2024 | Series Volume: Volume 102 | Status: Available.
- Type: Magazine | Title: Game Informer | Author: Game Studios | Year: 2024 | Edition Number: April Edition | Status: Available.
- Type: Book | Title: Crown of Midnight | Author: Sarah J. Maas | Year: 2013 | Genre: YA Fantasy | Status: Available.

#####
##### HOPE WE'LL SEE YOU SOON ! #####
#####
```