### Running Library Management System:

Using Intellij/Eclipse:

1. Let maven resolve dependencies
2. run SpringBootApplication

### Functionalities Exposed:

Book Controller class

The REST APIs exposed are:

* The create book API: POST [http://localhost:8080/api/books](http://localhost:8080//api/books) creates a book entity
* The get books API: GET <http://localhost:8080//api/books> get all the books from the repository
* The get books by id API: GET [http://localhost:8080/api/books/{id}](http://localhost:8080/api/books/%7Bid%7D) get the details of a specific book by ID
* The update book API: PUT [http://localhost:8080/api/{id}](http://localhost:8080/api/%7Bid%7D) update an existing book's information by id.
* The delete book API: Delete [http://localhost:8080/api/{id}](http://localhost:8080/api/%7Bid%7D) a book from the library by id.

Patron Controller class

The REST APIs exposed are:

* The create patrons API: POST <http://localhost:8080/api/patrons> creates a patron entity
* The get patrons API: GET <http://localhost:8080//api/patrons> get all the patrons from the repository
* The get patron by id API: GET [http://localhost:8080/api/patrons/{id}](http://localhost:8080/api/patrons/%7Bid%7D) get the details of a specific patron by ID
* The update patron API: PUT [http://localhost:8080/api/patrons/{id}](http://localhost:8080/api/patrons/%7Bid%7D) update an existing patron’s information by id.
* The delete patron API: Delete [http://localhost:8080/api/patrons/{id}](http://localhost:8080/api/patrons/%7Bid%7D) a book from the system by id.

Patron Controller class

The REST APIs exposed are:

* The create borrowing record API: POST [http://localhost:8080//api/borrow/{bookId}/patron/{patronId}](http://localhost:8080//api/borrow/%7BbookId%7D/patron/%7BpatronId%7D) creates a borrowing record entity and allow patrons to borrow books
* The get patrons API: GET <http://localhost:8080//api/patrons> get all the patrons from the repository