測試環境：

URL：<http://localhost:8080/api/2.0>

**API user**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **區塊** | **編號** | **名稱** | **描述** | **負責測試人員** | **測試結果** |
| user | 1 | addUser | 將資料寫入至User table中 | Adam | OK |
| 2 | deleteByUserId | 依userId值刪除對應User table中所屬全部資料 | Adam | OK |
| 3 | getUserHistory | 依userId值查詢借書/還書紀錄全部資料 | Adam | OK |
| book | 4 | addBook | 將資料寫入至Book table中 | Adam | OK |
| 5 | deleteByBookId | 依bookId值刪除對應Book table中所屬全部資料 | Adam | OK |
| 6 | getBookWhere | 依bookId值查詢目前是否被借走？ | Adam | OK |
| record | 7 | addRecord | 依userId與bookId值將借書資料寫入至Record table中 | Adam | OK |
| 8 | updateRecord | 依userId與bookId值Record table資料修改成還書狀態 | Adam | OK |

\*\*測試檢查重點：

* addUser確認資料是否寫入至User table且正確？
* deleteByUserId確認User table是否delete正確資料？
* getUserHistory確認顯示查詢借書/還書紀錄資料結果正確？
* addBook確認資料是否寫入至Book table且正確？
* deleteByBookId確認 Book table是否delete正確資料？
* getBookWhere 確認顯示資料結果正確？
* addRecord確認借書資料是否寫入至Record table且正確？
* updateRecord確認Record table資料修改成還書狀態？
* 1. addUser

|  |  |
| --- | --- |
| URI | API\_URL/user/addUser |
| Method | POST |
| Description | 將資料寫入至User table中 |
| Input(\*必填) | \*userName=[String] |
| Request Body | {"userName":"Ford"} |
| Response Body | {  "code": 20000,  "errorMsg": "",  "user": {  "id": 24,  "userId": "U6",  "userName": "Ford"  },  "msg": " User insert successfully!! "  } |
| 測試截圖 |  |
| 測試結果 | 通過—20201130 —OK |

* 2. deleteByUserId

|  |  |
| --- | --- |
| URI | API\_URL/user/deleteByUserId |
| Method | DELETE |
| Description | 依userId值刪除對應User table中所屬全部資料 |
| Input(\*必填) | \*userId=[String] |
| Request Body | {"userId":"U6"} |
| Response Body | {  "code": 20000,  "errorMsg": "",  "msg": " User delete successfully!!"  } |
| 測試截圖 |  |
| 測試結果 | 通過—20201130—OK |

* 3.getUserHistory

|  |  |
| --- | --- |
| URI | API\_URL/user/getUserHistory |
| Method | GET |
| Description | 依userId值查詢借書/還書紀錄全部資料 |
| Input(\*必填) | \*userId=[String] |
| Request Body | {"userId":"U1"} |
| Response Body | 正確訊息：  {  "code": 20000,  "errorMsg": "",  "userName": "Adam",  "record": [  {  "id": 11,  "userId": "U1",  "bookId": "B1",  "borrowDate": "2020-11-23T08:19:19.000+00:00",  "returnDate": "2020-11-30T08:19:19.000+00:00",  "actualReturnDate": "2020-11-23T08:30:10.000+00:00",  "bookStatus": 2  },  {  "id": 12,  "userId": "U1",  "bookId": "B2",  "borrowDate": "2020-11-23T08:19:25.000+00:00",  "returnDate": "2020-11-29T08:19:25.000+00:00",  "actualReturnDate": "2020-11-23T08:30:20.000+00:00",  "bookStatus": 2  },  {  "id": 13,  "userId": "U1",  "bookId": "B3",  "borrowDate": "2020-11-23T08:19:35.000+00:00",  "returnDate": "2020-12-01T08:19:35.000+00:00",  "actualReturnDate": **null**,  "bookStatus": 1  }  ],  "msg": " User History!! "  }  錯誤訊息：  {  "code": 99999,  "errorMsg": " Other error !! ",  "userName": **null**,  "record": **null**,  "msg": ""  } |
| 測試截圖 | 正確訊息：    錯誤訊息： |
| 測試結果 | 通過—20201130—OK |

* 4. addBook

|  |  |
| --- | --- |
| URI | API\_URL/book/addBook |
| Method | POST |
| Description | 將資料寫入至Book table中 |
| Input(\*必填) | \*bookName=[String]  \*bookDate=[Integer] |
| Request Body | {"bookName":"飄","bookDate":5} |
| Response Body | {  "code": 20000,  "errorMsg": "",  "book": {  "id": 22,  "bookId": "B6",  "bookName": "飄",  "bookDate": 5,  "bookAmount": 1  },  "msg": " Book insert successfully!! "  } |
| 測試截圖 |  |
| 測試結果 | 通過—20201130 —OK |

* 5. deleteByBookId

|  |  |
| --- | --- |
| URI | API\_URL/book/deleteByBookId |
| Method | DELETE |
| Description | 依bookId值刪除對應Book table中所屬全部資料 |
| Input(\*必填) | \*bookId=[String] |
| Request Body | {"bookId":"B6"} |
| Response Body | {  "code": 20000,  "errorMsg": "",  "msg": " Book delete successfully!!"  } |
| 測試截圖 |  |
| 測試結果 | 通過—20201130—OK |

* 6.getBookWhere

|  |  |
| --- | --- |
| URI | API\_URL/book/getBookWhere |
| Method | GET |
| Description | 依bookId值查詢目前是否被借走 |
| Input(\*必填) | \*bookId=[String] |
| Request Body | {"bookId":"B1"} |
| Response Body | {  "code": 20000,  "errorMsg": "",  "bookName": "戰爭與和平",  "userName": "Caesar",  "returnDate": "2020-11-30T08:30:34.000+00:00",  "msg": " 此書已被借!! "  } |
| 測試截圖 |  |
| 測試結果 | 通過—20201130—OK |

* 7. addRecord

|  |  |
| --- | --- |
| URI | API\_URL/record/addRecor |
| Method | POST |
| Description | 依userId與bookId值將借書資料寫入至Record table中 |
| Input(\*必填) | \*userId=[String]  \*bookId=[String] |
| Request Body | {"userId":"U5","bookId":"B3"} |
| Response Body | {  "code": 20000,  "errorMsg": "",  "record": {  "id": 23,  "userId": "U5",  "bookId": "B3",  "borrowDate": "2020-11-30T09:11:30.331+00:00",  "returnDate": "2020-12-08T09:11:30.331+00:00",  "actualReturnDate": **null**,  "bookStatus": 1  },  "msg": " 已完成借書手續!! "  } |
| 測試截圖 |  |
| 測試結果 | 通過—20201130 —OK |

* 8. updateRecord

|  |  |
| --- | --- |
| URI | API\_URL/record/updateRecord |
| Method | PUT |
| Description | 依userId與bookId 將Record table資料修改成還書狀態 |
| Input(\*必填) | \*userId=[String]  \*bookId=[String] |
| Request Body | {"userId":"U4","bookId":"B2"} |
| Response Body | {  "code": 20000,  "errorMsg": "",  "msg": " 已完成歸還手續!! "  } |
| 測試截圖 |  |
| 測試結果 | 通過—20201130 —OK |