

Library

Unless otherwise specified, assume that the library is empty before a method is called.

Get Library

Precise URL of service endpoint	library/get
Accepted operation	Get
Parameters	N/A
Description of functionality	<p>First use: creates library if not in repository already</p> <p>After first first: gets library from repository and uses same library for subsequent Operations</p>

Picture of what it returns:

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:8080/library/get`
- Method:** GET
- Status:** 200 OK
- Time:** 1122 ms
- Size:** 403 B
- Response Body (JSON):**

```
1 {
2   "libraryId": 90,
3   "librarySchedule": {
4     "scheduleId": 91,
5     "timeslots": []
6   },
7   "users": [],
8   "titles": [],
9   "rooms": [],
10  "roomReservations": null,
11  "titleReservations": null
12 }
```

Delete Library

Precise URL of service endpoint	http://localhost:8080/library/remove
Accepted operation	Post
Parameters	N/A
Description of functionality	Delete all the schedule, users, rooms, room reservations and title reservations from the library

Library before deleting

The screenshot shows a REST client interface with a GET request to `http://localhost:8080/library/get` sent successfully. The response is a JSON object with the following structure:

```
1 {
2   "libraryId": 188,
3   "librarySchedule": {
4     "scheduleId": 189,
5     "timeslots": []
6   },
7   "users": [
8     {
9       "userId": 190,
10      "username": "Liam",
11      "password": "pass123",
12      "fullname": "Liam Serour",
13      "residentialAddress": "123 avenue",
14      ...
15    }
16  ]
17 }
```

Library delete method

POST http://localhost:8080/library/remove Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
-----	-------	-------------	-----	-----------

Body Cookies Headers (7) Test Results 200 OK 393 ms 212 B Save Response

Pretty Raw Preview Visualize Text 1

Library after delete

GET http://localhost:8080/library/get Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
-----	-------	-------------	-----	-----------

Body Cookies Headers (8) Test Results 200 OK 256 ms 372 B Save Response

Pretty Raw Preview Visualize JSON 1

```

1 {
2   "libraryId": 188,
3   "librarySchedule": null,
4   "users": [],
5   "titles": [],
6   "rooms": [],
7   "roomReservations": [],
8   "titleReservations": []
9 }
```

Title

Create title

Precise URL of service endpoint	titles/create/Movie1?description=The first book&genre=Horror&isAvailable=true&titleType=Book
Accepted operation	Post

Parameters	Name Description Genre isAvailable titleType
Description of functionality	Creates title with all parameters and saves it to library

Picture of what it returns:

The screenshot shows a REST client interface with a POST request to `http://localhost:8080/titles/create/Movie1?description=The first book&genre=Horror&isAvailable=true&titleType=Book`. The request parameters are listed in a table:

Key	Value	Description
<input checked="" type="checkbox"/> description	The first book	
<input checked="" type="checkbox"/> genre	Horror	
<input checked="" type="checkbox"/> isAvailable	true	
<input checked="" type="checkbox"/> titleType	Book	

The response body is shown in JSON format:

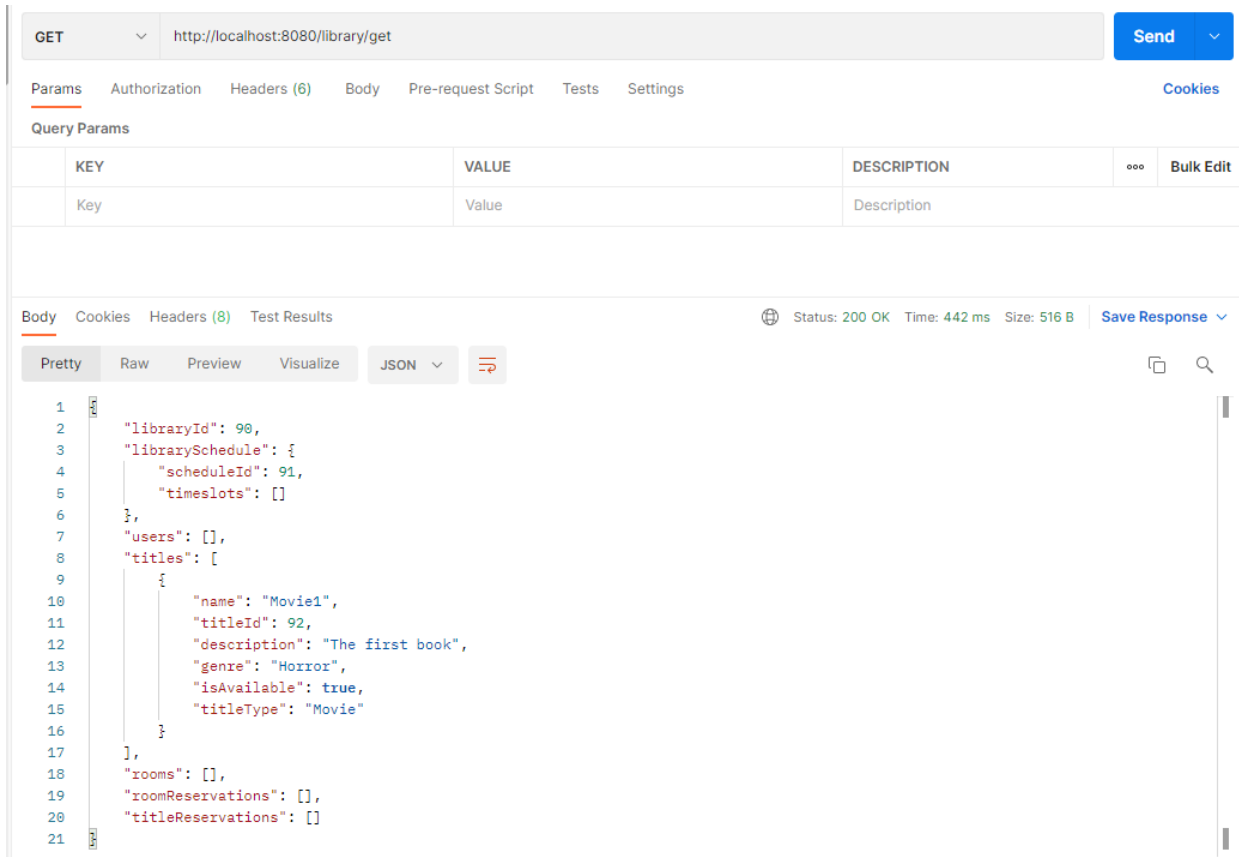
```

1  {
2    "titleId": 92,
3    "description": "The first book",
4    "genre": "Horror",
5    "titleType": "Movie",
6    "available": true
7  }

```

The status bar indicates a successful response with status 200 OK, time 489 ms, and size 352 B.

Picture of the library after:



Get title by id

Precise URL of service endpoint	titles/get/92
Accepted operation	Get
Parameters	titleId
Description of functionality	Returns title with specified titleId

Picture of what it returns:

GET
⌵
http://localhost:8080/titles/get/92
Send
⌵

Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body
Cookies
Headers (8)
Test Results

200 OK
168 ms
352 B
Save Response
⌵

Pretty
Raw
Preview
Visualize
JSON
⌵

⌵

```

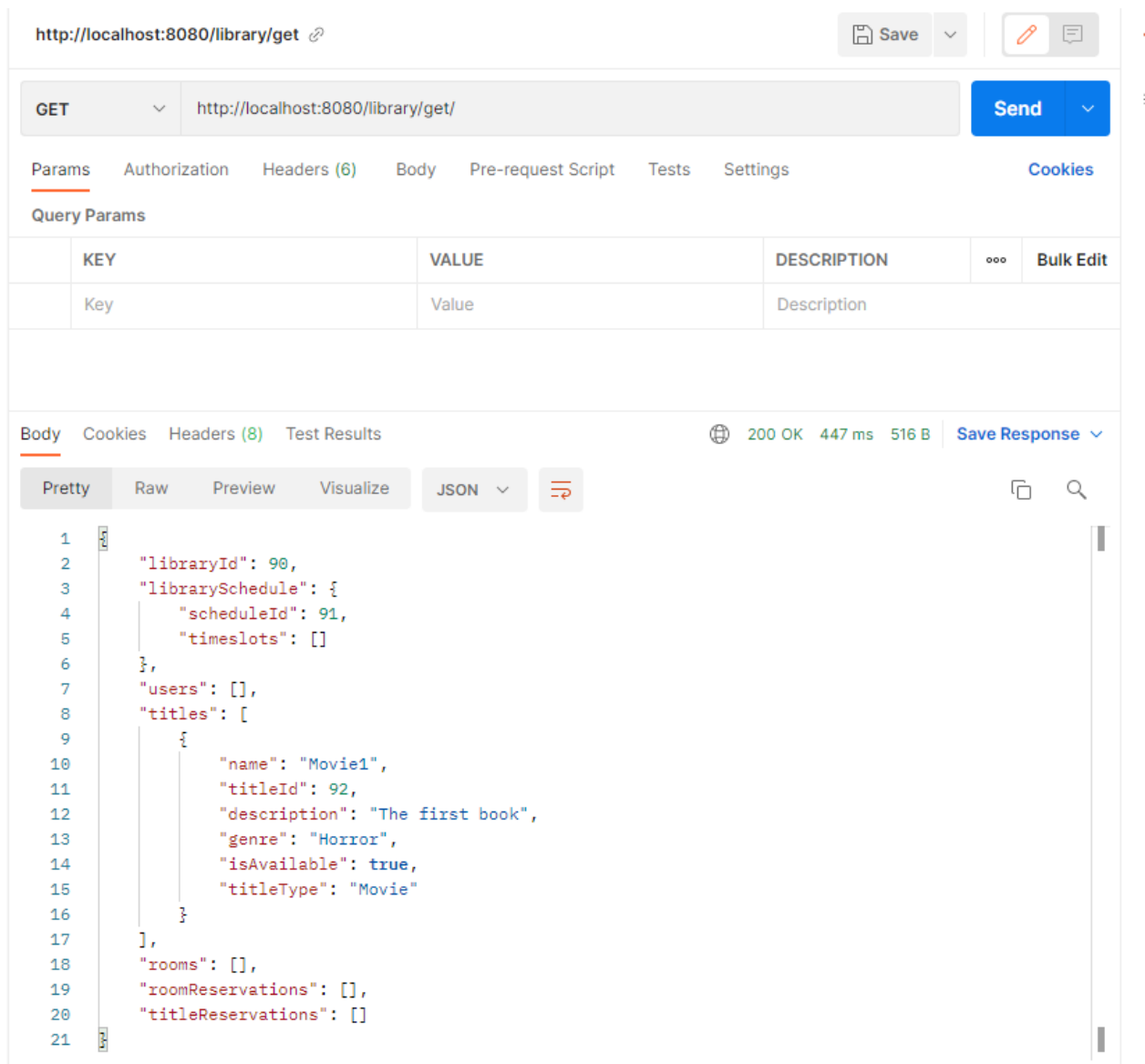
1
2
3
4
5
6
7
"tokenId": 92,
"description": "The first book",
"genre": "Horror",
"titleType": "Movie",
"available": true

```

Get all titles

Precise URL of service endpoint	titles/get/
Accepted operation	Get
Parameters	N/A
Description of functionality	Returns a list of all the titles.

Picture of what it returns:



Update title

Precise URL of service endpoint	<code>titles/update/Movie1/?description=The first book is now a movie&genre=Horror&titleType=Movie</code>
Accepted operation	post

Parameters	Name Description genre titleType
Description of functionality	Updates a title based on its name for all copies of it in the library. The name cannot be updated.

Picture of library before:

The screenshot shows a REST client interface with the following details:

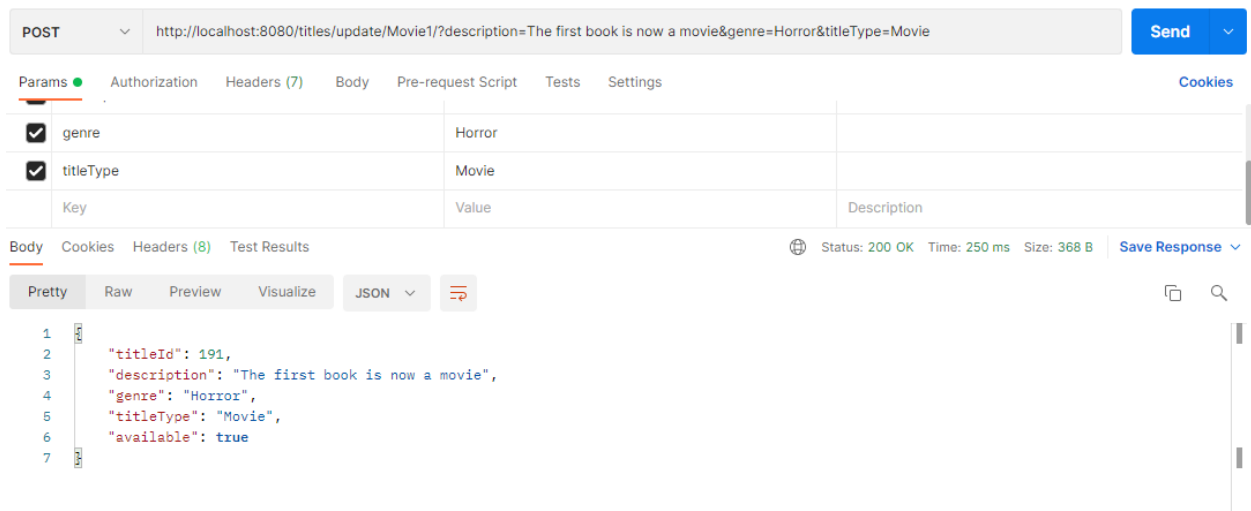
- URL:** `http://localhost:8080/library/get/`
- Method:** GET
- Response Status:** 200 OK, 447 ms, 516 B
- Response Body (JSON):**

```

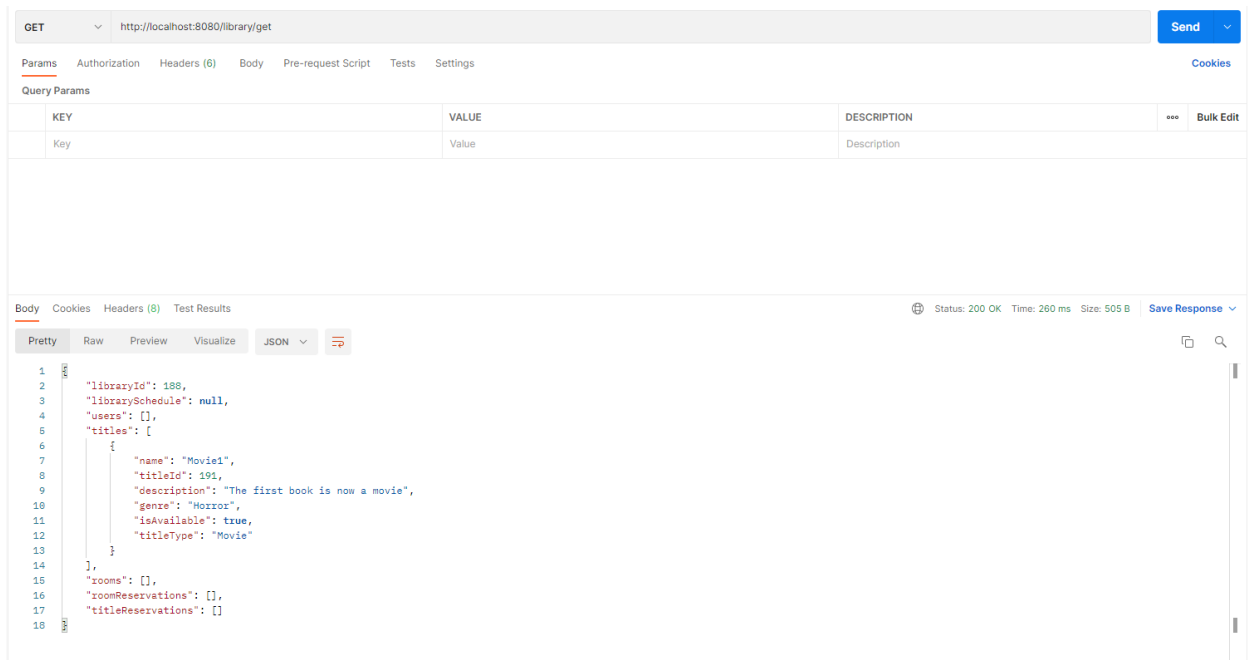
1  {
2    "libraryId": 90,
3    "librarySchedule": {
4      "scheduleId": 91,
5      "timeslots": []
6    },
7    "users": [],
8    "titles": [
9      {
10       "name": "Movie1",
11       "titleId": 92,
12       "description": "The first book",
13       "genre": "Horror",
14       "isAvailable": true,
15       "titleType": "Movie"
16     }
17   ],
18   "rooms": [],
19   "roomReservations": [],
20   "titleReservations": []
21 }

```


Picture of what it returns:



Picture of the library after:



Remove title - to do

Precise URL of service endpoint	titles/remove/298
Accepted operation	post

Parameters	titleId
Description of functionality	Removes a copy of a title by id from the library.

Picture of library before:

GET

http://localhost:8080/library/get

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

KEY	VALUE	DESCRIPTION	***	Bulk Edit
Key	Value	Description		

Body

Cookies

Headers (8)

Test Results

Status: 200 OK

Time: 274 ms

Size: 489 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

"libraryId": 293,

"librarySchedule": null,

"users": [],

"titles": [

{

"name": "Movie1",

"titleId": 298,

"description": "The first book",

"genre": "Horror",

"isAvailable": true,

"titleType": "Book"

},

"rooms": [],

"roomReservations": [],

"titleReservations": []

]

Picture of what it returns:

POST

http://localhost:8080/titles/remove/298

Send

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

KEY	VALUE	DESCRIPTION	***	Bulk Edit
Key	Value	Description		

Body

Cookies

Headers (7)

Test Results

Status: 200 OK

Time: 249 ms

Size: 212 B

Save Response

Pretty

Raw

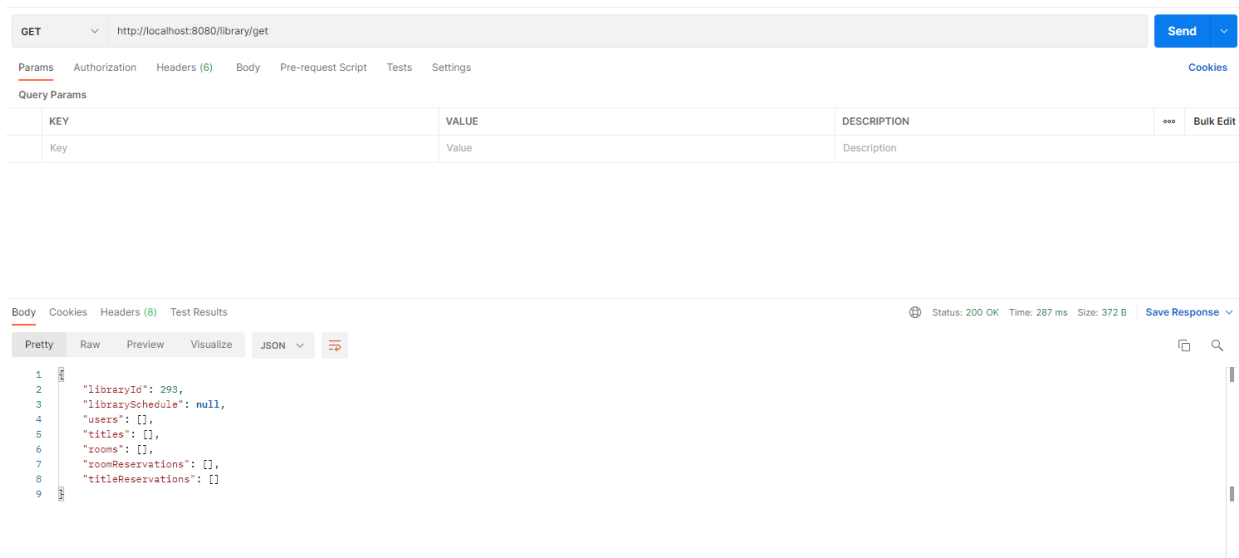
Preview

Visualize

Text

1

Picture of the library after:



Client

Create client

Precise URL of service endpoint	clients/create/Liam?password=pass123&fullName=Liam Serour&residentialAddress=123 avenue&email=liam@gmail.com&isResident=true&isOnline=yes
Accepted operation	Post
Parameters	Username Password fullName residentialAddress Email isResident isOnline
Description of functionality	Creates client and adds them to the library.

Picture of what it returns:

POST ▼ http://localhost:8080/clients/create/Liam?password=pass123&fullName=Liam Serour&residentialAddress=123 avenue&email=liam@gmail.com&isResident=true&isOnline=yes Send ▼

Params ● Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

<input checked="" type="checkbox"/>	residentialAddress	123 avenue	
<input checked="" type="checkbox"/>	email	liam@gmail.com	
<input checked="" type="checkbox"/>	isResident	true	
<input checked="" type="checkbox"/>	isOnline	yes	
	Key	Value	Description

Body Cookies Headers (8) Test Results 200 OK 374 ms 395 B Save Response ▼

Pretty Raw Preview Visualize JSON ▼ ≡

```
1  {
2    "residentialAddress": "123 avenue",
3    "email": "liam@gmail.com",
4    "userId": 93,
5    "username": "Liam",
6    "password": "pass123",
7    "online": false,
8    "resident": true
9  }
```

Picture of the library after:

GET
http://localhost:8080/library/get/
Send

Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body
Cookies
Headers (8)
Test Results

Status: 200 OK
Time: 427 ms
Size: 687 B
Save Response

Pretty
Raw
Preview
Visualize
JSON

```

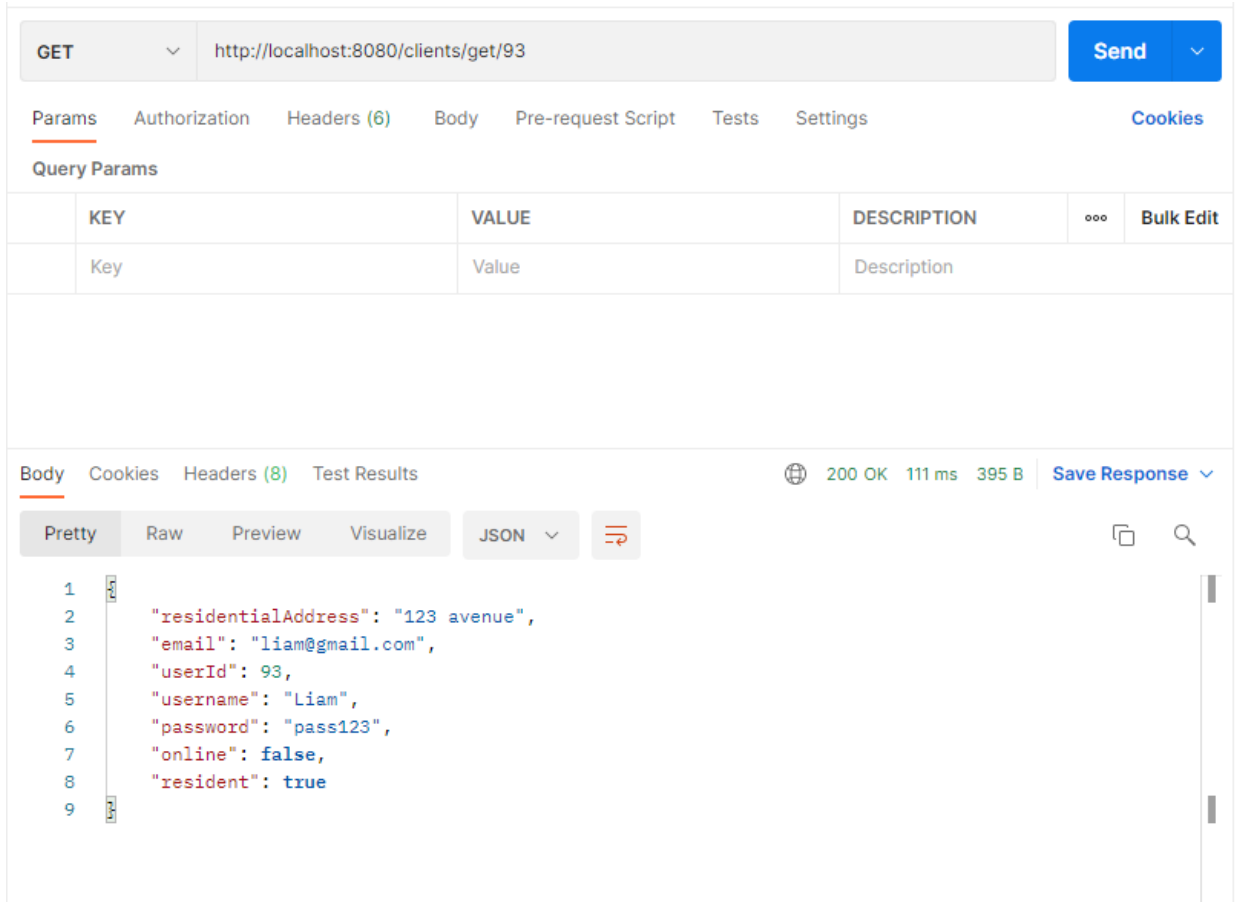
1
2  "libraryId": 90,
3  "librarySchedule": {
4    "scheduleId": 91,
5    "timeslots": []
6  },
7  "users": [
8    {
9      "userId": 93,
10     "username": "Liam",
11     "password": "pass123",
12     "fullname": "Liam Serour",
13     "residentialAddress": "123 avenue",
14     "email": "liam@gmail.com",
15     "isResident": true,
16     "isOnline": false
17   }
18 ],

```

Get client by ID

Precise URL of service endpoint	clients/get/93
Accepted operation	Get
Parameters	id
Description of functionality	Gets a specified client by id

Picture of what it returns:



Get clients

Precise URL of service endpoint	clients/get/
Accepted operation	Get
Parameters	N/A
Description of functionality	Gets all clients in the library

Picture of what it returns:

GET

http://localhost:8080/clients/get/

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body

Cookies

Headers (8)

Test Results

200 OK 135 ms 397 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```

1  [
2    {
3      "residentialAddress": "123 avenue",
4      "email": "liam@gmail.com",
5      "userId": 93,
6      "username": "Liam",
7      "password": "pass123",
8      "online": false,
9      "resident": true
10   }
11 ]

```

Update clients - to do

Precise URL of service endpoint	update/Liam?password=changed password&fullName=Liam Serour&residentialAddress=456 avenue&email=hi@hi.com&isResident=true&isOnline=true
Accepted operation	post

Parameters	Username password Fullname Residential address Is Resident Is Online
Description of functionality	Updates client information in the library. Cannot update the username.

Picture before :

GET

⌵

http://localhost:8080/library/get

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Query Params

	KEY	VALUE
	Key	Value

Body

Cookies

Headers (8)

Test Results

Pretty

Raw

Preview

Visualize

JSON ⌵

⌵

```
1  {
2    "libraryId": 238,
3    "librarySchedule": null,
4    "users": [
5      {
6        "userId": 241,
7        "username": "Liam",
8        "password": "pass123",
9        "fullname": "Liam Serour",
10       "residentialAddress": "123 avenue",
11       "email": "liam@gmail.com",
12       "isResident": true,
13       "isOnline": false
14     }
15   ],
16   "titles": [],
17   "rooms": [],
18   "roomReservations": [],
19   "titleReservations": []
20 }
```

Picture of what it returns:

POST [http://localhost:8080/clients/update/Liam?password=changed password&fullName=Liam Serour&residentialAddress=456 avenue&email=hi@hi.com&isResident=true&isOnline=true](http://localhost:8080/clients/update/Liam?password=changed%20password&fullName=Liam%20Serour&residentialAddress=456%20avenue&email=hi@hi.com&isResident=true&isOnline=true) **Send**

Params Authorization Headers (7) Body Pre-request Script Tests Settings **Cookies**

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> password	changed password	
<input checked="" type="checkbox"/> fullName	Liam Serour	
<input checked="" type="checkbox"/> residentialAddress	456 avenue	
<input checked="" type="checkbox"/> email	hi@hi.com	
<input checked="" type="checkbox"/> isResident	true	
<input checked="" type="checkbox"/> isOnline	true	
Key	Value	Description

Body Cookies Headers (8) Test Results Status: 200 OK Time: 209 ms Size: 400 B **Save Response**

Pretty Raw Preview Visualize JSON

```

1  {
2    "residentialAddress": "456 avenue",
3    "email": "hi@hi.com",
4    "userId": 241,
5    "username": "Liam",
6    "password": "changed password",
7    "online": false,
8    "resident": true
9  }

```

Picture of the library after:

GET <http://localhost:8080/library/get> **Send**

Params Authorization Headers (6) Body Pre-request Script Tests Settings **Cookies**

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (8) Test Results Status: 200 OK Time: 271 ms Size: 548 B **Save Response**

Pretty Raw Preview Visualize JSON

```

1  {
2    "libraryId": 238,
3    "librarySchedule": null,
4    "users": [
5      {
6        "userId": 241,
7        "username": "Liam",
8        "password": "changed password",
9        "fullName": "Liam Serour",
10       "residentialAddress": "456 avenue",
11       "email": "hi@hi.com",
12       "isResident": true,
13       "isOnline": false
14     }
15   ],
16   "titles": [],
17   "rooms": [],
18   "roomReservations": [],
19   "titleReservations": []
20 }

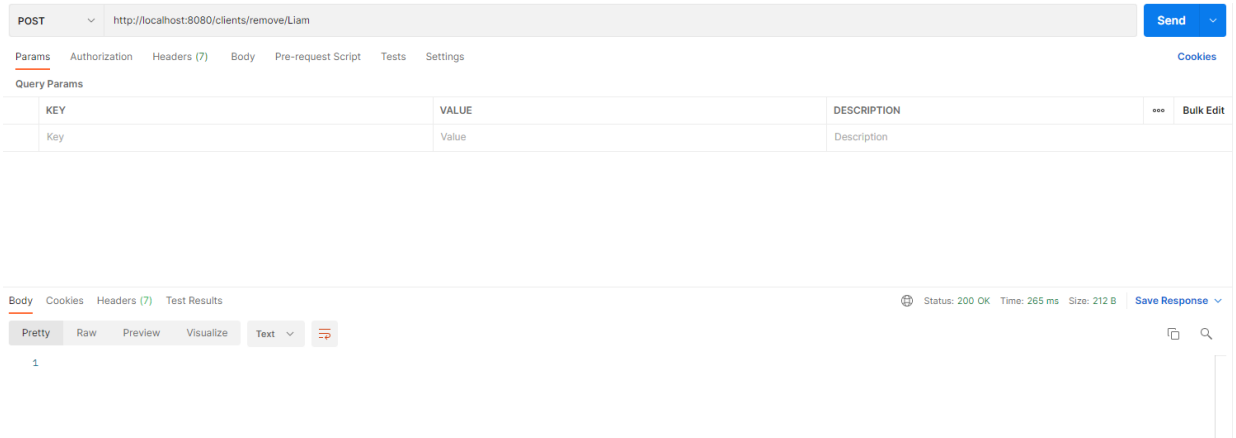
```

Remove client

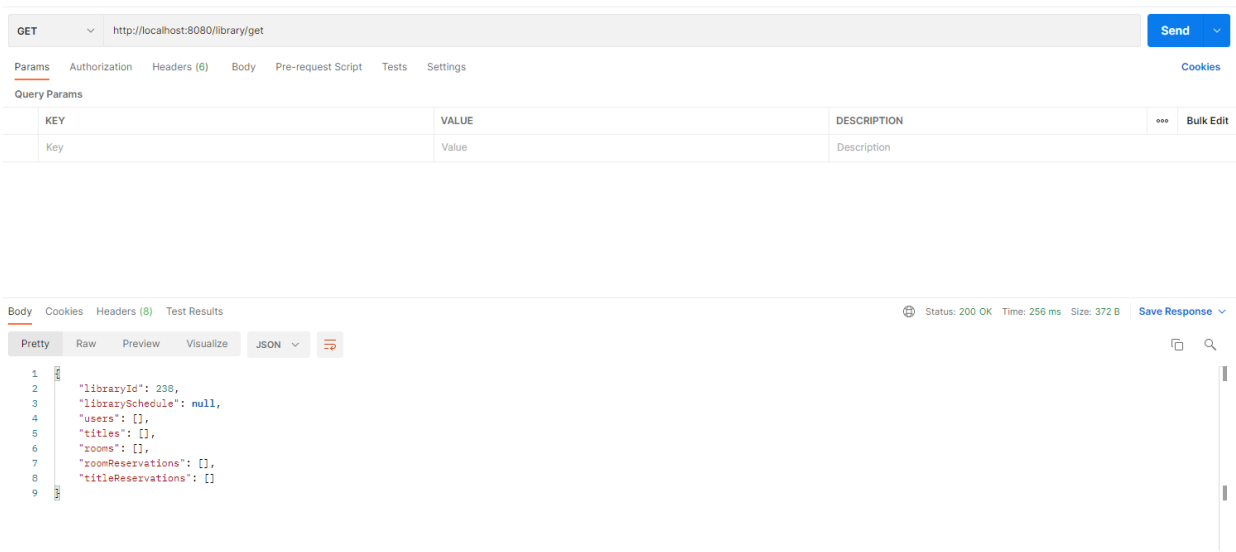
Precise URL of service endpoint	clients/remove/Liam
---------------------------------	---------------------

Accepted operation	post
Parameters	username
Description of functionality	Removes a client based on username from the library.

Picture of what it returns:



Picture of the library after:



Librarian

Create Librarian

Precise URL of service endpoint	librarians/create/Avi?password=password 456&fullName=aviSomech&isHeadLibrarian=true
Accepted operation	Post
Parameters	Username Password Fullname isHeadLibrarian
Description of functionality	Creates a librarian . Can only be a head librarian if no other head librarians exists.

Picture of what it returns:

The screenshot shows a REST client interface with a POST request to `http://localhost:8080/librarians/create/Avi?password=password 456&fullName=aviSomech&isHeadLibrarian=true`. The request parameters are:

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> password	password 456	
<input checked="" type="checkbox"/> fullName	aviSomech	
<input checked="" type="checkbox"/> isHeadLibrarian	true	

The response body is a JSON object:

```
1 {
2   "userId": 296,
3   "username": "Avi",
4   "password": "password 456",
5   "fullName": "aviSomech",
6   "headLibrarian": true
7 }
```

The status is 200 OK, Time: 347 ms, Size: 354 B.

Picture of the library after:

GET
http://localhost:8080/library/get
Send

Params
Authorization
Headers (6)
Body
Pre-request Script
Tests
Settings
Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Ed
Key	Value	Description		

Body
Cookies
Headers (8)
Test Results
Status: 200 OK
Time: 374 ms
Size: 554 B
Save Response

Pretty
Raw
Preview
Visualize
JSON

```

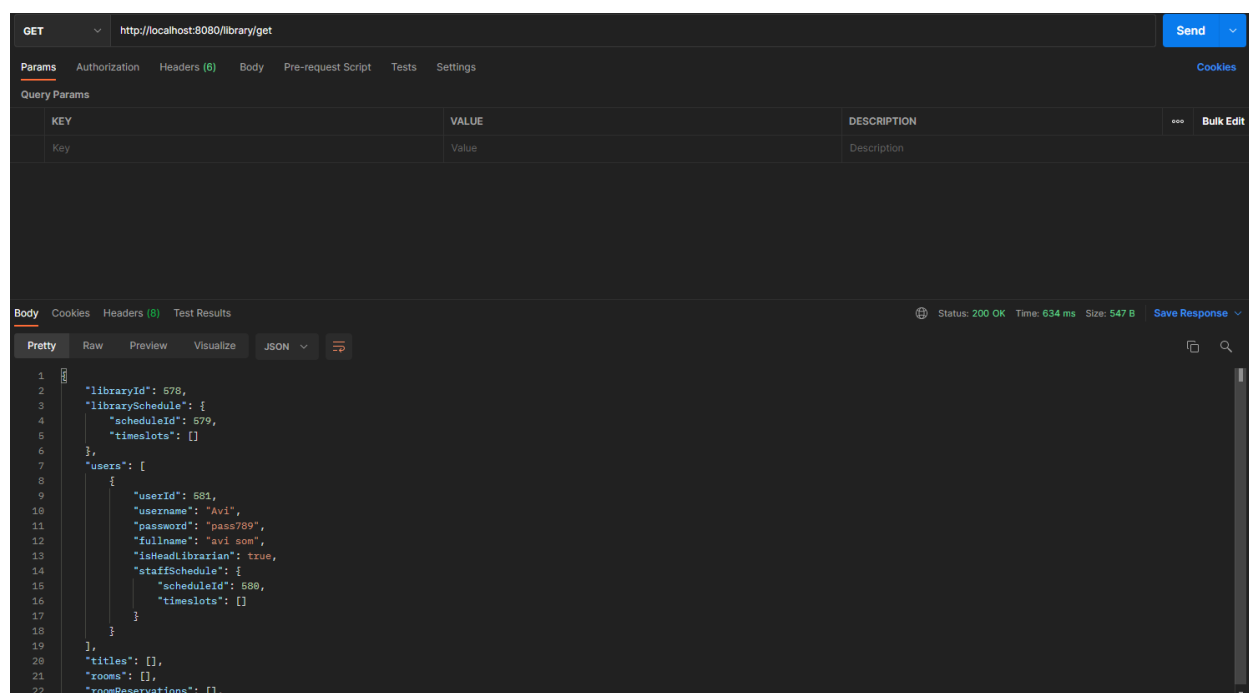
1  {
2    "libraryId": 299,
3    "librarySchedule": {
4      "scheduleId": 294,
5      "timeslots": []
6    },
7    "users": [
8      {
9        "userId": 296,
10       "username": "Avi",
11       "password": "password 456",
12       "fullname": "aviSomech",
13       "isHeadLibrarian": true,
14       "staffSchedule": {
15         "scheduleId": 296,
16         "timeslots": []
17       }
18     }
19   ],
20   "titles": [],
21   "rooms": [],
22   "roomReservations": [],
23   "staffReservations": []
24 }

```

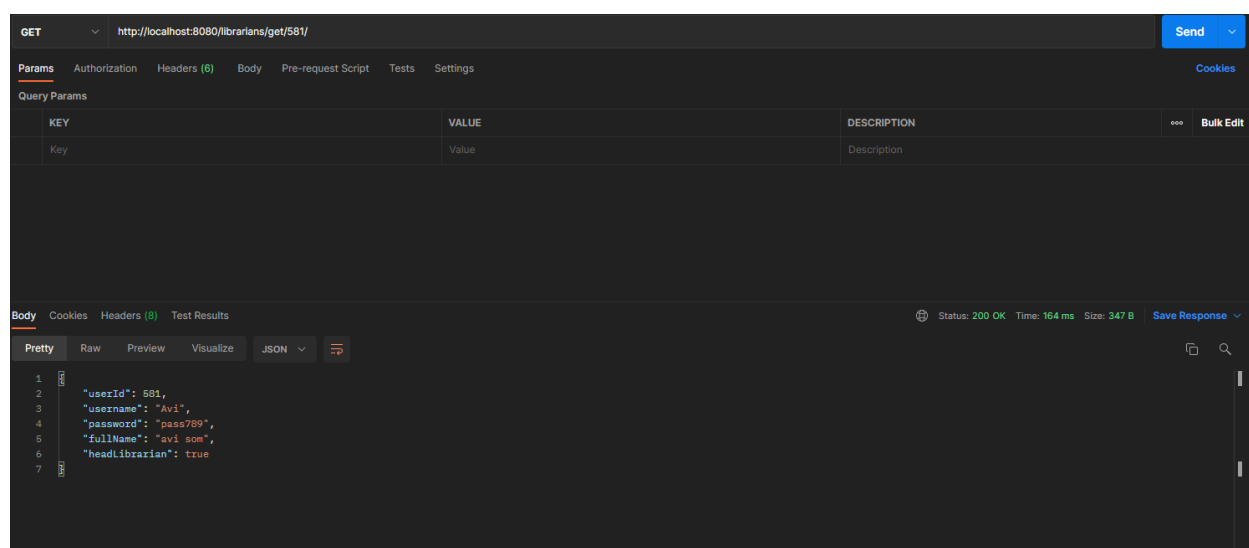
Get librarian by ID

Precise URL of service endpoint	librarians/get/581
Accepted operation	Get
Parameters	User ID
Description of functionality	Gets a librarian by a specified username

Picture of library before:



Picture of what it returns:



Get all librarians

Precise URL of service endpoint	librarians/get
Accepted operation	Get

Parameters	N/A
Description of functionality	Gets all the librarians in the library.

Library before:

GET

http://localhost:8080/library/get

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body

Cookies

Headers (8)

Test Results

Status: 200 OK Time: 345 ms Size: 554 B Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1  {
2    "libraryId": 482,
3    "librarySchedule": {
4      "scheduleId": 483,
5      "timeslots": []
6    },
7    "users": [
8      {
9        "userId": 485,
10       "username": "Avi",
11       "password": "password 456",
12       "fullName": "aviSomech",
13       "isHeadLibrarian": true,
14       "staffSchedule": {
15         "scheduleId": 484,
16         "timeslots": []
17       }
18     }
19   ],
20   "titles": [],
21   "rooms": [],
22   "roomReservations": []
23 }
```

Picture of what it returns:

GET

http://localhost:8080/librarians/get

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body

Cookies

Headers (8)

Test Results

Status: 200 OK Time: 132 ms Size: 356 B Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1  {
2    "userId": 486,
3    "username": "Avi",
4    "password": "password 456",
5    "fullName": "aviSomech",
6    "headLibrarian": true
7  }
```

Update librarian

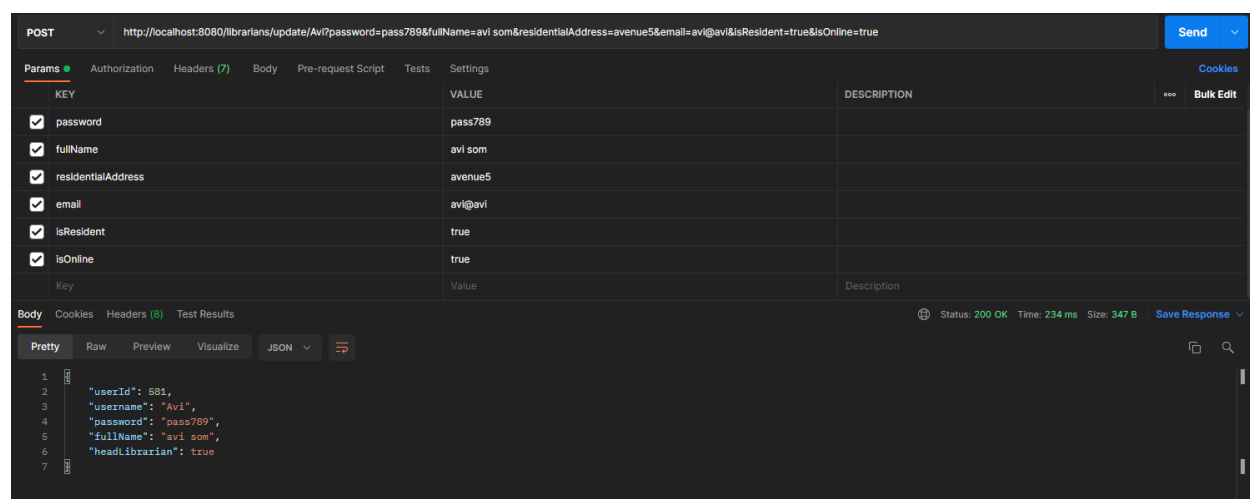
Precise URL of service endpoint	update/Avi?password=pass789&fullName=avisom&residentialAddress=avenue5&email=avi@avi&isResident=true&isOnline=true
Accepted operation	post
Parameters	Username Password Fullname Residential Address Email is Resident Is Online
Description of functionality	Updates a librarian account to the desired fields.

Library before

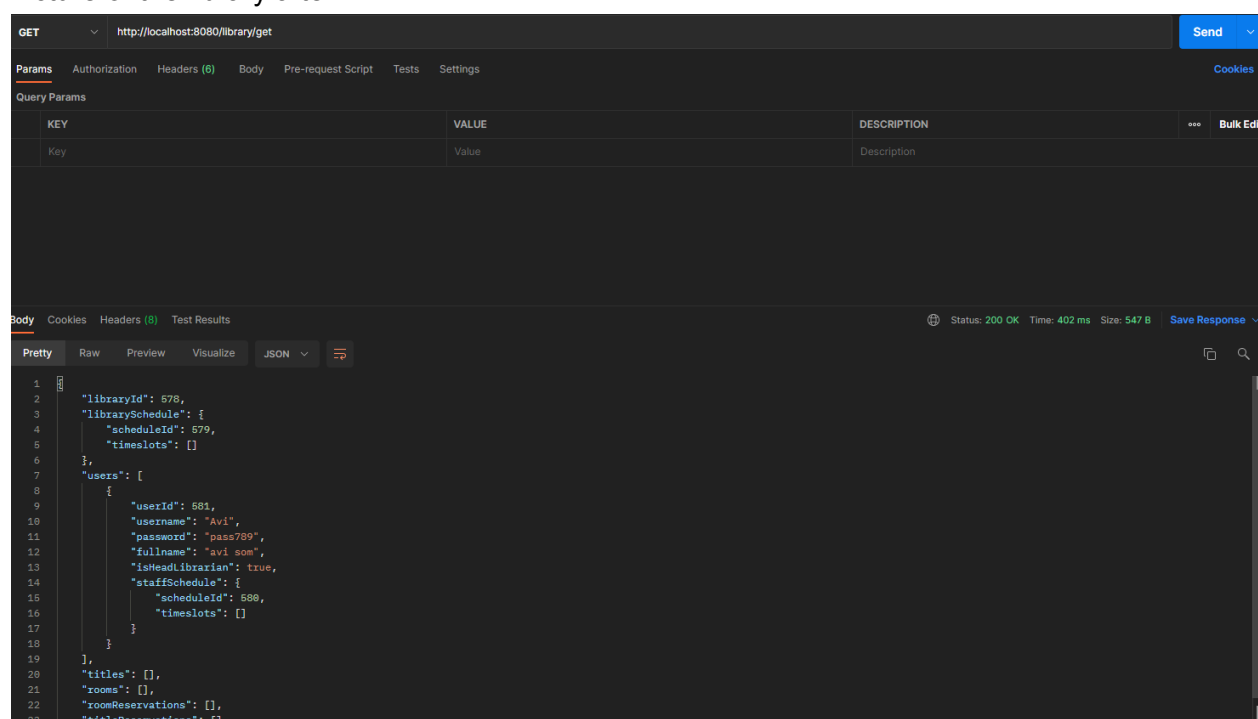
The screenshot shows a REST client interface with a GET request to `http://localhost:8080/library/get`. The response is a JSON object with the following structure:

```
1 {
2   "libraryId": 578,
3   "librarySchedule": {
4     "scheduleId": 579,
5     "timeslots": []
6   },
7   "users": [
8     {
9       "userId": 581,
10      "username": "Avi",
11      "password": "password 456",
12      "fullName": "aviSomech",
13      "isHeadLibrarian": true,
14      "staffSchedule": {
15        "scheduleId": 580,
16        "timeslots": []
17      }
18    }
19  ],
}
```

Picture of what it returns:



Picture of the library after:

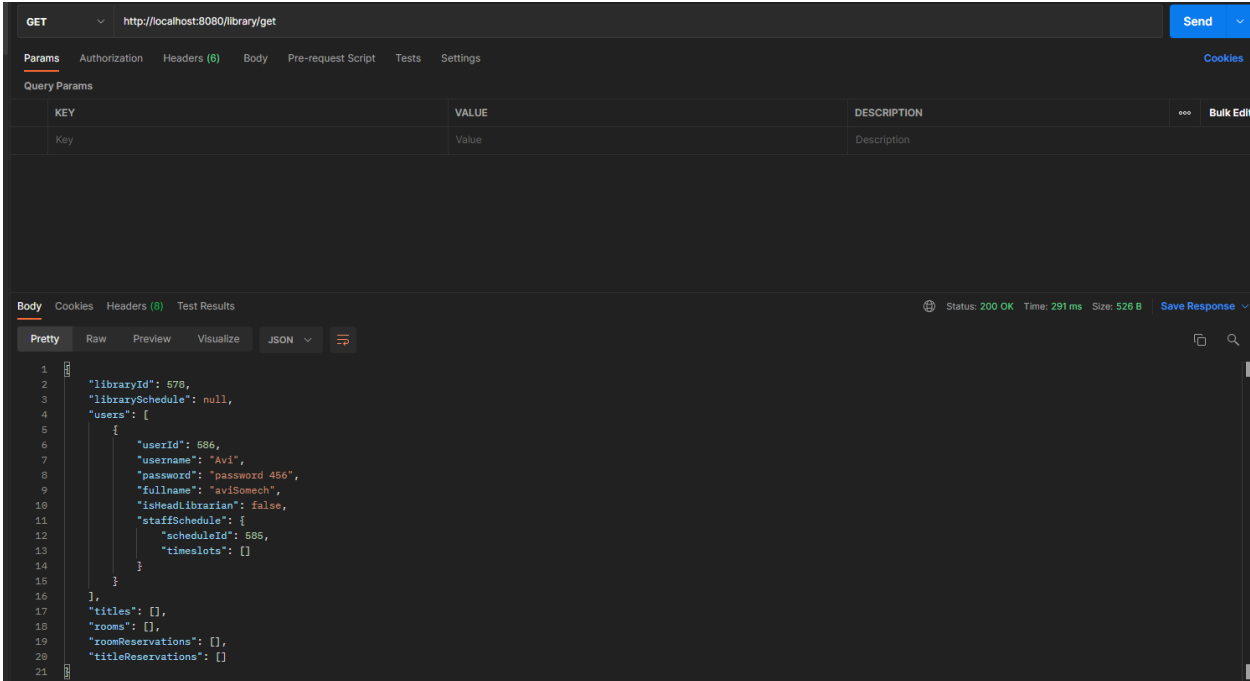


Remove Librarian

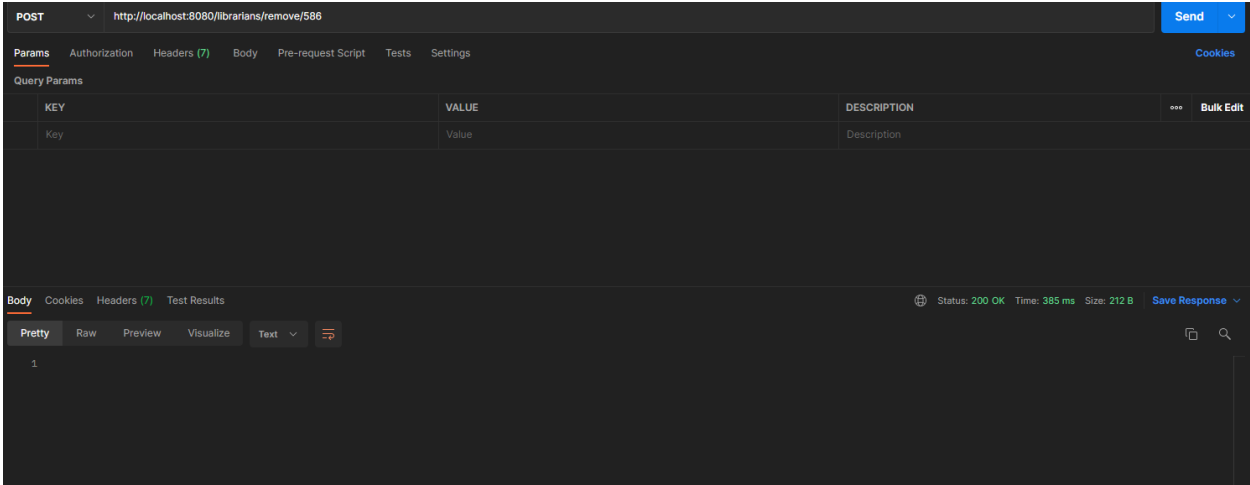
Precise URL of service endpoint	
Accepted operation	Post

Parameters	UserId
Description of functionality	Removes a librarian by userId. Note you cannot fire a head librarian.

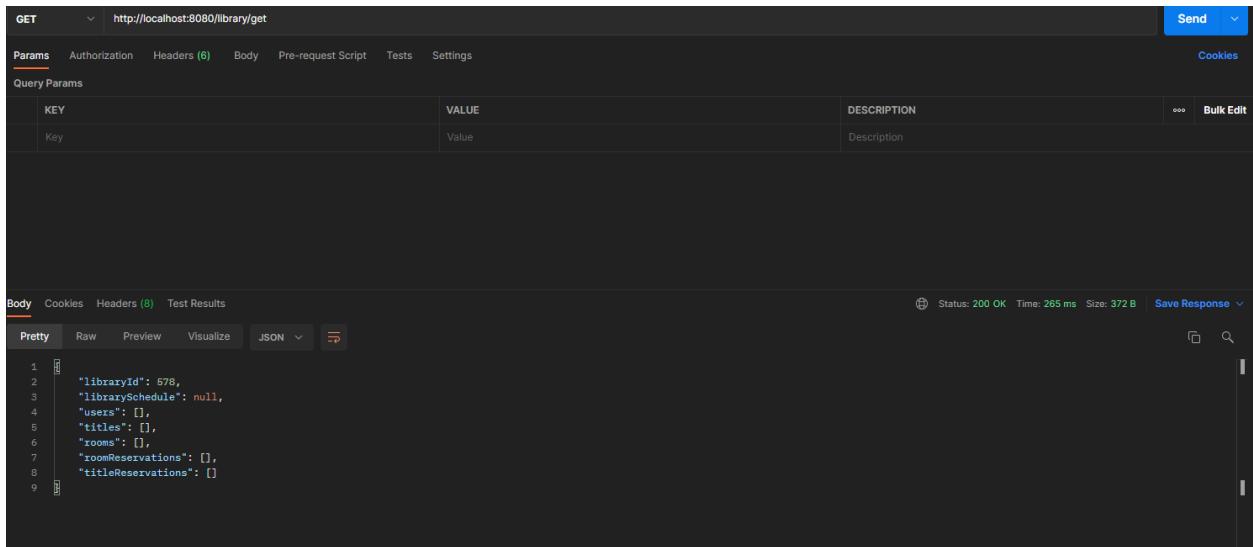
Picture of library before:



Picture of what it returns:



Picture of the library after:



Room

Precise URL of service endpoint	
Accepted operation	
Parameters	
Description of functionality	

Picture of what it returns:

Picture of the library after:

Create Room

Precise URL of service endpoint	rooms/create/5?isAvailable=true&roomType=Study
Accepted operation	post

Parameters	Capacity isAvailable roomType
Description of functionality	Creates a room with the required fields and uses a boolean to keep track of the rooms availability.

Picture of what it returns:

POST

http://localhost:8080/rooms/create/5?isAvailable=true&roomType=Study

Send

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> isAvailable	true			
<input checked="" type="checkbox"/> roomType	Study			
Key	Value	Description		

Body

Cookies

Headers (8)

Test Results

Status: 200 OK

Time: 269 ms

Size: 316 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

"roomId": 299,

"capacity": 5,

"roomType": "Study",

"available": true

Picture of the library after:

GET

http://localhost:8080/library/get

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body

Cookies

Headers (8)

Test Results

Status: 200 OK

Time: 286 ms

Size: 437 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

"libraryId": 293,

"librarySchedule": null,

"users": [],

"titles": [],

"rooms": [

{

"roomId": 299,

"capacity": 5,

"isAvailable": true,

"roomType": "Study"

}

],

"roomReservations": [],

"titleReservations": []

Get all Rooms

Precise URL of service endpoint	rooms/get
Accepted operation	get
Parameters	N/A
Description of functionality	Gets all the rooms in the library

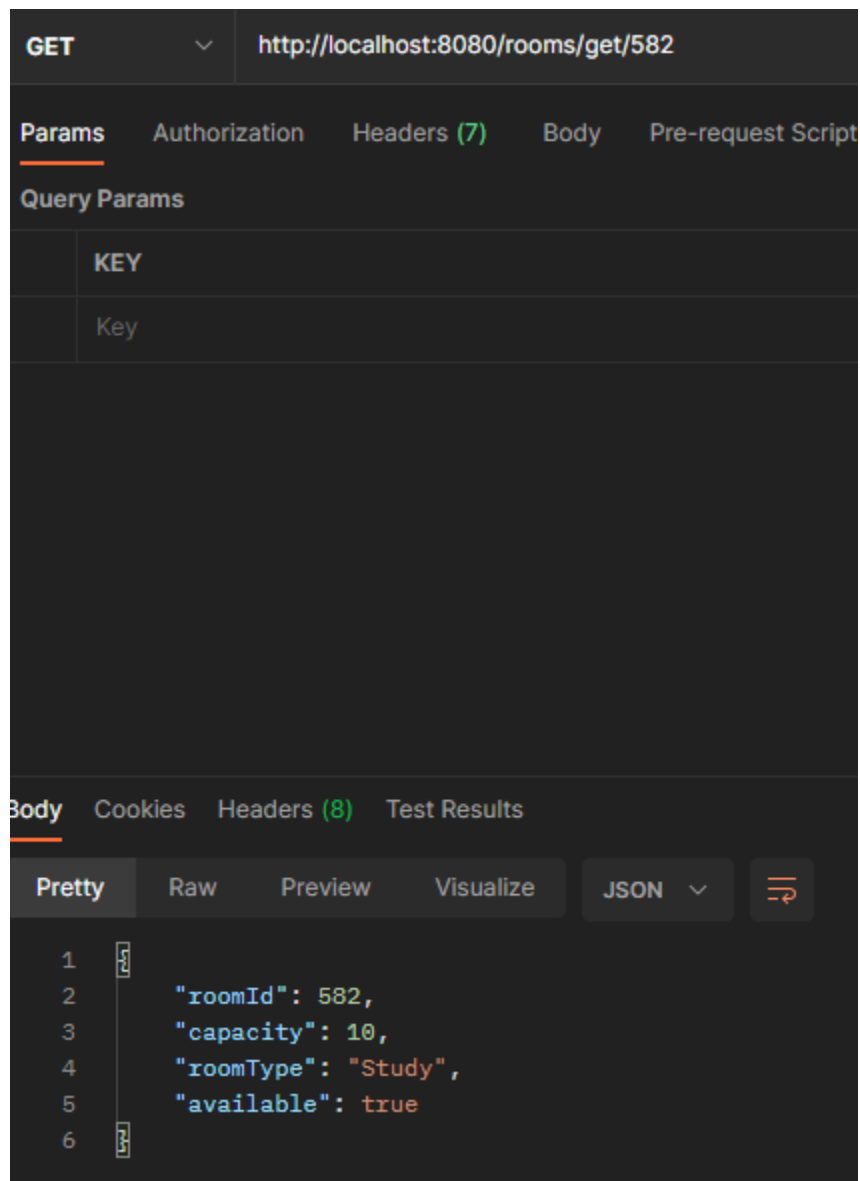
Picture of what it returns:

The screenshot shows a REST client interface. At the top, a GET request is configured to `http://localhost:8080/rooms/get`. Below the URL bar, tabs for Params, Authorization, Headers (6), Body, Pre-request Script, Tests, and Settings are visible. The 'Query Params' section is expanded, showing a table with columns KEY, VALUE, and DESCRIPTION. The table contains one entry: 'Key' with 'Value' and 'Description'. The 'Body' tab is selected, showing the response in 'Pretty' format. The response is a JSON array with one object: `[{"roomId": 299, "capacity": 5, "roomType": "Study", "available": true}]`. The status bar at the bottom indicates 'Status: 200 OK', 'Time: 101 ms', and 'Size: 318 B'.

Get room by Id

Precise URL of service endpoint	/rooms/get/582
Accepted operation	get
Parameters	roomId
Description of functionality	Gets a room by room id.

Picture of what it returns:



Update Room- to do

Precise URL of service endpoint	
Accepted operation	post
Parameters	Capacity roomType

	is Available
Description of functionality	Only the availability of a room can be updated

Picture of what it returns:

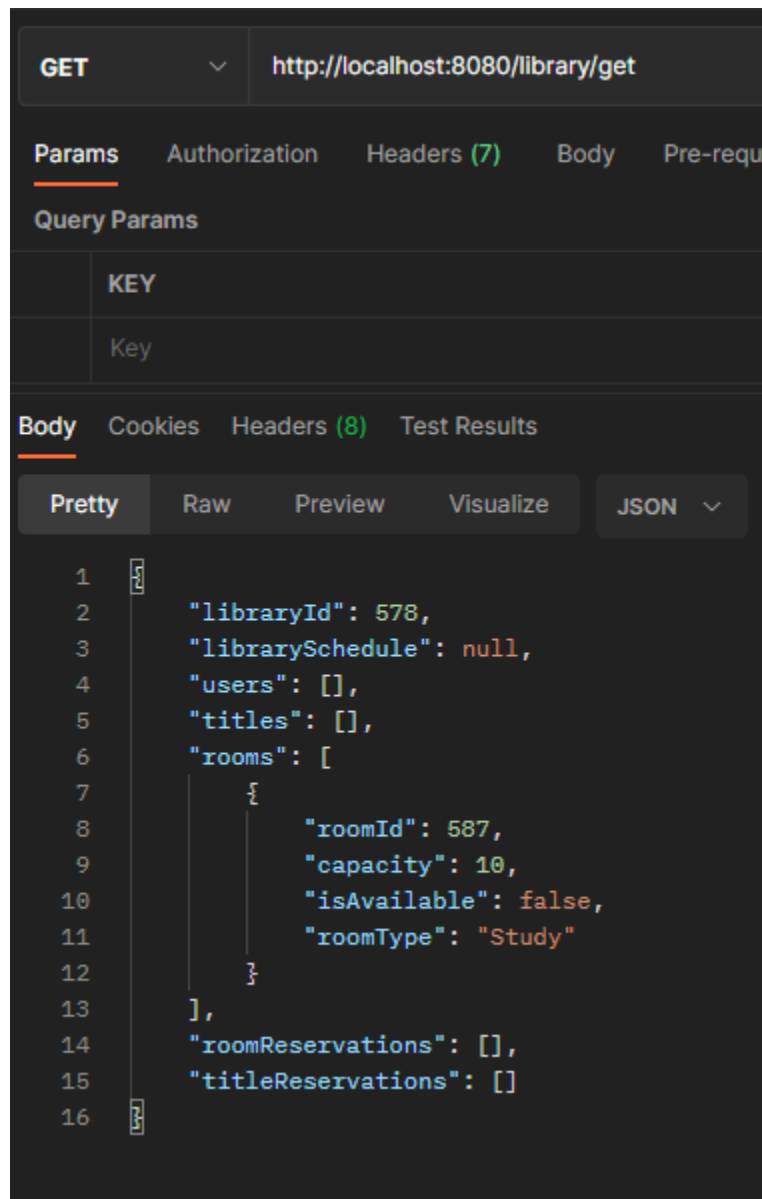
The screenshot shows a REST client interface with a POST request to `http://localhost:8080/rooms/update/587?isAvailable=false`. The 'Params' tab is active, showing a query parameter `isAvailable` with a value of `false`. The 'Body' tab is also active, showing a JSON response in 'Pretty' format:

```

1  {
2    "roomId": 587,
3    "capacity": 10,
4    "roomType": "Study",
5    "available": false
6  }

```

Picture of the library after:



Remove Room

Not necessary since a room cannot be removed. If it is not available to the public then the availability is updated.

RoomReservation

Create RoomReservation- to do

Precise URL of service endpoint	<code>/roomReservations/create/588?userId=589&startHour=10&startMin=40&endHour=11&endMin=43&year=2020&month=6&day=11</code>
Accepted operation	Post
Parameters	roomId userId startHour startMin endHour endMin Year Month day
Description of functionality	Creates a room reservation using the room id and the user who requests to reserve it.

Picture of library before:

POST ⌵ http://localhost:8080/roomReservations/create/588?userId=589&startHour=10&startMin=40&endHour=11&endMin=43&year=2020&month=6&day=11

Params ● Authorization Headers (8) Body Pre-request Script Tests Settings

<input checked="" type="checkbox"/>	userId	589
<input checked="" type="checkbox"/>	startHour	10
<input checked="" type="checkbox"/>	startMin	40
<input checked="" type="checkbox"/>	endHour	11
<input checked="" type="checkbox"/>	endMin	43
<input checked="" type="checkbox"/>	year	2020
<input checked="" type="checkbox"/>	month	6
<input checked="" type="checkbox"/>	day	11
	Key	Value

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON ⌵ ⌵

```
1  [
2    "room": {
3      "roomId": 588,
4      "capacity": 5,
5      "roomType": "Event",
6      "available": true
7    },
8    "client": {
9      "residentialAddress": "123 avenue",
10     "email": "liam@gmail.com",
11     "userId": 589,
12     "username": "Liam",
13     "password": "pass123",
14     "online": false,
15     "resident": true
16   }
17 ]
```

Picture of what it returns:

```
GET http://localhost:8080/library/get

Params Authorization Headers (7) Body Pre-request Script
Query Params
KEY

Body Cookies Headers (8) Test Results
Pretty Raw Preview Visualize JSON
19         roomId : 588,
20         "capacity": 5,
21         "isAvailable": true,
22         "roomType": "Event"
23     }
24 ],
25     "roomReservations": [
26     {
27         "startTime": "10:40:00",
28         "endTime": "11:43:00",
29         "date": "3920-07-11",
30         "room": {
31             "roomId": 588,
32             "capacity": 5,
33             "isAvailable": true,
34             "roomType": "Event"
35         },
36         "client": {
37             "userId": 589,
38             "username": "Liam",
39             "password": "pass123",
40             "fullname": "Liam Serour",
41             "residentialAddress": "123 avenue",
42             "email": "liam@gmail.com",
43             "isResident": true,
44             "isOnline": false
45         },
46         "timeSlotId": 590
47     }
48 ],
49     "titleReservations": []
50 }
```

Picture of the library after:

Update RoomReservation- to do

Note a reservation cannot be updated but it can be deleted and a new one can be made.

Remove RoomReservation-to do

Precise URL of service endpoint	
Accepted operation	post
Parameters	roomId userId
Description of functionality	Removes a reservation that a user has made.

Picture before:

```
GET http://localhost:8080/library/get

Params Authorization Headers (7) Body Pre-request Script

Body Cookies Headers (8) Test Results

Pretty Raw Preview Visualize JSON ↕

15      ],
16      "titles": [],
17      "rooms": [
18        {
19          "roomId": 588,
20          "capacity": 5,
21          "isAvailable": true,
22          "roomType": "Event"
23        }
24      ],
25      "roomReservations": [
26        {
27          "startTime": "10:40:00",
28          "endTime": "11:43:00",
29          "date": "3910-07-11",
30          "room": {
31            "roomId": 588,
32            "capacity": 5,
33            "isAvailable": true,
34            "roomType": "Event"
35          },
36          "client": {
37            "userId": 589,
38            "username": "Liam",
39            "password": "pass123",
40            "fullname": "Liam Serour",
41            "residentialAddress": "123 avenue",
42            "email": "liam@gmail.com",
43            "isResident": true,
44            "isOnline": false
45          },
46          "timeSlotId": 592
47        }
48      ]
49    }
50  ]
51}
```

Picture of what it returns:

Picture of the library after:

Schedule

Note that a library schedule and staff schedule work identically. For the sake of the tests we will use staff schedule.

Create Schedule- to do

Precise URL of service endpoint	
Accepted operation	Post
Parameters	librarianId startHour startMin endHour endMin Year Month day
Description of functionality	Creates a schedule of a librarian using timeslots.

Picture of what it returns:

Picture of the library after:

Picture of what it returns:

Update Schedule- to do

Precise URL of service endpoint	
Accepted operation	
Parameters	
Description of functionality	

Picture of what it returns:

Picture of the library after:

Remove Schedule-to do

Precise URL of service endpoint	
Accepted operation	post
Parameters	librarianId startHour startMin endHour endMin Year Month
Description of functionality	Removes a timeslot in the librarians schedule.

Picture of what it returns:

Picture of the library after:

TimeSlot

Create Timeslot - to do

Precise URL of service endpoint	
Accepted operation	
Parameters	
Description of functionality	

Picture of what it returns:

Picture of the library after:

Update Timeslot - to do

Precise URL of service endpoint	
Accepted operation	
Parameters	
Description of functionality	

Picture of what it returns:

Picture of the library after:

Remove Timeslot -to do

Precise URL of service endpoint	
Accepted operation	
Parameters	
Description of functionality	

Picture of what it returns:

Picture of the library after:

Use cases

Reserve title

- Precise URL of service endpoint
`http://localhost:8080/titles/reserve/Movie1?clientUsername=Liam`

- Accepted operation

post

Parameters

Client username

Title name

Description of functionality

Reserve a title for a client. Changes `isAvailable` to `false`, creates a title reservation class with the given title and client and adds it to the library, updates the return date to 2 weeks later than the current day.

POST

http://localhost:8080/titles/reserve/Movie1?clientUsername=Liam

Send

Params

Authorization

Headers (7)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	clientUsername	Liam			
	Key	Value	Description		

Body

Cookies

Headers (8)

Test Results

Status: 200 OK

Time: 492 ms

Size: 353 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```

1  {
2    "titleId": 92,
3    "description": "The first book",
4    "genre": "Horror",
5    "titleType": "Movie",
6    "available": false
7  }
```

GET

http://localhost:8080/library/get/

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Body

Cookies

Headers (8)

Test Results

Status: 200 OK

Time: 455 ms

Size: 1.04 KB

Save Response

Pretty

Raw

Preview

Visualize

JSON

```

27  {
28    },
29    "rooms": [],
30    "roomReservations": [],
31    "titleReservations": [
32      {
33        "returnDate": "2021-11-25",
34        "isCheckedOut": false,
35        "title": {
36          "name": "Movie1",
37          "titleId": 92,
38          "description": "The first book",
39          "genre": "Horror",
40          "isAvailable": false,
41          "titleType": "Movie"
42        },
43        "client": {
44          "userId": 93,
45          "username": "Liam",
46          "password": "pass123",
47          "fullname": "Liam Seroux",
48          "residentialAddress": "123 avenue",
49          "email": "liam@gmail.com",
50          "isResident": true,
51          "isOnline": false
52        },
53        "titleReservationId": 94
54      }
55    ]
56  }
```

Reserve title

- Precise URL of service endpoint
http://localhost:8080/titles/reserve/Movie1?clientUsername=Liam
- Accepted operation

post

Parameters

Client username

Title name

Description of functionality

Reserve a title for a client. Changes isAvailable to false, creates a title reservation class with the given title and client and adds it to the library, updates the return date to 2 weeks later than the current day.