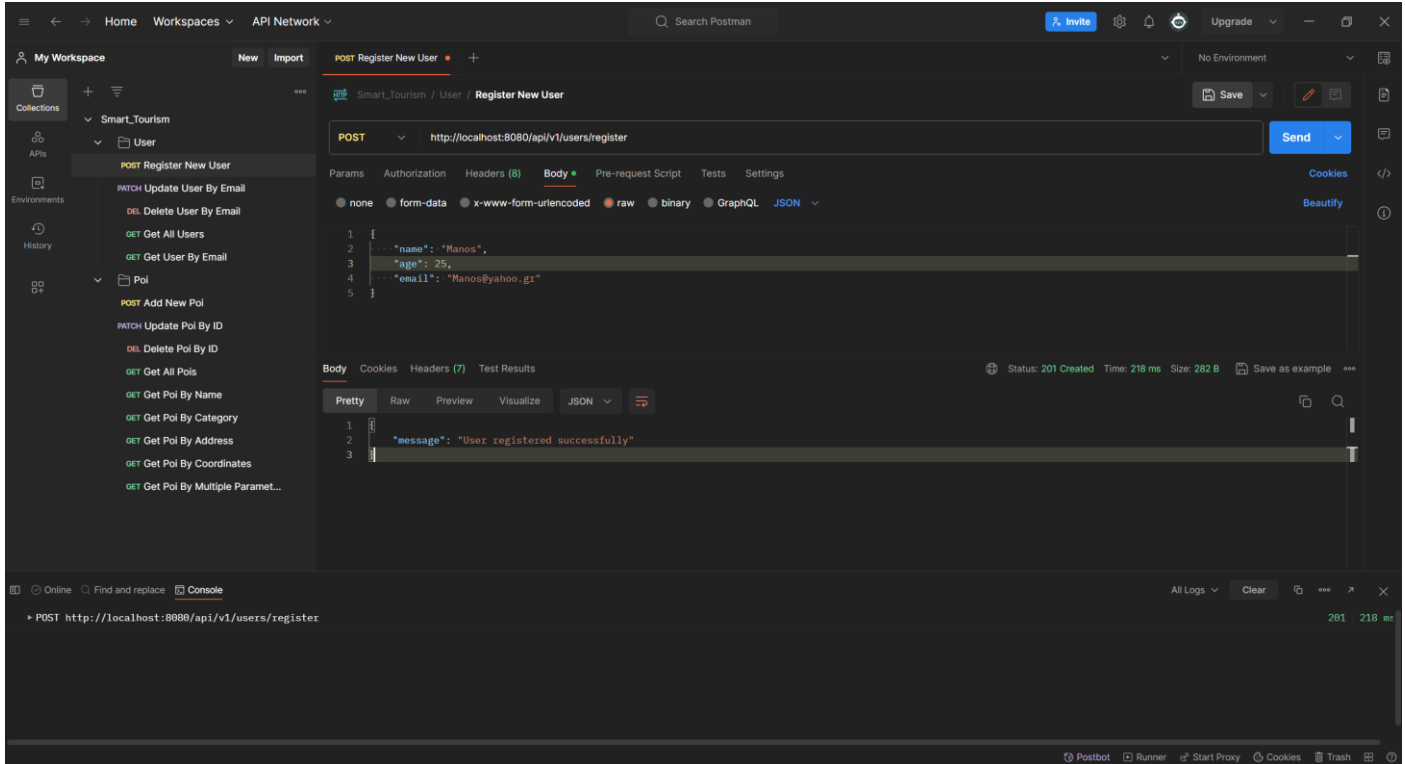


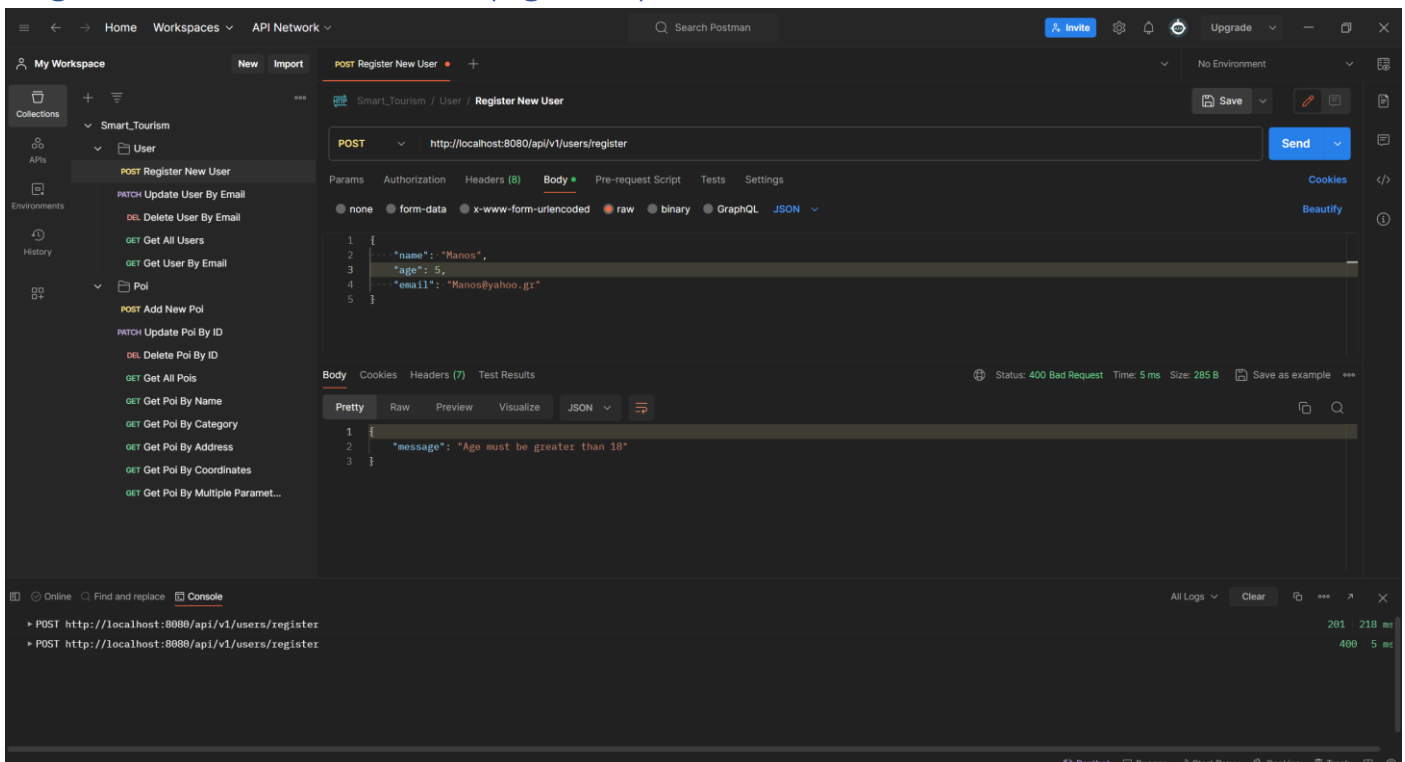
ΓΚΑΓΚΑΚΗΣ ΒΑΣΙΛΗΣ

ΑΠΑΛΛΑΚΤΙΚΗ ΕΡΓΑΣΙΑ

Register New User- Success



Register New User – Failure (Age <18)



Update User – Before

```
_id: ObjectId('65cee1f7935d8ee5c7f365fb')
name: "Manos"
age: 25
email: "manos@yahoo.gr"
__v: 0
```

Update User – Failure (Age <18)

The screenshot shows the Postman interface with a workspace named 'Smart_Tourism'. The selected collection is 'User', and the specific request is 'PATCH Update User By Email'. The URL is 'http://localhost:8080/api/v1/users/update?email=manos@yahoo.gr'. The request body is a JSON object:

```
{  "name": "Updated Manos",  "age": 2,  "email": "Manos@yahoo.gr"}
```

. The response status is '400 Bad Request' with a message:

```
{  "message": "Age must be greater than 18"}
```

. The console at the bottom shows a list of recent requests, including the failed PATCH request.

Request	Status	Time	Size
POST http://localhost:8080/api/v1/users/register	201	218 ms	
POST http://localhost:8080/api/v1/users/register	400	5 ms	
GET http://localhost:8080/api/v1/users?email=Giannis@yahoo.gr	404	74 ms	
GET http://localhost:8080/api/v1/users/all	200	62 ms	
GET http://localhost:8080/api/v1/users?email=Manos@yahoo.gr	200	86 ms	
PATCH http://localhost:8080/api/v1/users/update?email=manos@yahoo.gr	400	5 ms	

Update User – Success

The screenshot shows the Postman interface with a workspace named 'Smart_Tourism'. The selected collection is 'User', and the endpoint is 'Update User By Email'. The request is a PATCH to 'http://localhost:8080/api/v1/users/update?email=manos@yahoo.gr'. The request body is a JSON object:

```
{  "name": "Updated Manos",  "age": 27,  "email": "Manos@yahoo.gr"}
```

. The response status is 200 OK, with a time of 76 ms and a size of 389 B. The response body is a JSON object:

```
{  "message": "User updated successfully",  "updatedUser": {    "_id": "65cee1f7935d8ee5c7f365fb",    "name": "Updated Manos",    "age": 27,    "email": "manos@yahoo.gr",    "__v": 0  }}
```

. The console at the bottom shows the request and response logs.

Update User – After

```
_id: ObjectId('65cee1f7935d8ee5c7f365fb')  
name : "Updated Manos"  
age : 27  
email : "manos@yahoo.gr"  
__v : 0
```

Get All Users

The screenshot shows the Postman interface for the 'Get All Users' endpoint. The URL is `http://localhost:8080/api/v1/users/all`. The response is a JSON array of user objects, displayed in the 'Body' tab. The status is 200 OK, and the response size is 343 B.

Query Params

Key	Value	Description
Key	Value	Description

Body

```
1 {
2   {
3     "name": "Vasilis"
4   },
5   {
6     "name": "Tasos"
7   },
8   {
9     "name": "Updated Maria"
10  },
11  {
12    "name": "Dimitris"
13  },
14  {
15    "name": "Updated Manos"
16  }
17 }
```

Console

```
> POST http://localhost:8080/api/v1/users/register 490 5 ms
> GET http://localhost:8080/api/v1/users/?email=Giannis@yahoo.gr 404 74 ms
> GET http://localhost:8080/api/v1/users/all 200 62 ms
```

Get User By Email

The screenshot shows the Postman interface for the 'Get User By Email' endpoint. The URL is `http://localhost:8080/api/v1/users/?email=Manos@yahoo.gr`. The response is a JSON object with the user's name, displayed in the 'Body' tab. The status is 200 OK, and the response size is 259 B.

Query Params

Key	Value	Description
Key	Value	Description
email	Manos@yahoo.gr	

Body

```
1 {
2   "name": "Updated Manos"
3 }
```

Console

```
> PATCH http://localhost:8080/api/v1/users/update/?email=manos@yahoo.gr 200 76 ms
> GET http://localhost:8080/api/v1/users/all 200 59 ms
> GET http://localhost:8080/api/v1/users/?email=Manos@yahoo.gr 200 60 ms
```

Register User - Failure (email already exists)

The screenshot shows the Postman interface with a workspace named 'Smart_Tourism'. The selected collection is 'User', and the request is 'POST Register New User'. The URL is 'http://localhost:8080/api/v1/users/register'. The request body is a JSON object:

```
{  "name": "Manos",  "age": 30,  "email": "Manos@yahoo.gr"}
```

. The response status is '400 Bad Request' with a time of 61 ms and a size of 278 B. The response body is:

```
{  "message": "Email already exists"}
```

. The console at the bottom shows three POST requests to the same endpoint, with the last one failing with a 400 status.

Postman Interface showing a failed POST request to register a user. The request body is a JSON object with name, age, and email. The response is a 400 Bad Request with the message "Email already exists".

Delete User – Failure (User not found)

The screenshot shows the Postman interface with the same workspace and collection. The selected request is 'DELETE Delete User By Email'. The URL is 'http://localhost:8080/api/v1/users/delete?email=Madfgnos@yahoo.gr'. The query parameter 'email' has the value 'Madfgnos@yahoo.gr'. The response status is '404 Not Found' with a time of 67 ms and a size of 270 B. The response body is:

```
{  "message": "User not found"}
```

. The console at the bottom shows three requests: two POSTs and one DELETE, with the last one failing with a 404 status.

Postman Interface showing a failed DELETE request to delete a user by email. The response is a 404 Not Found with the message "User not found".

Users before delete

The screenshot shows the MongoDB Compass interface for the 'smart_tourism' database. The 'users' collection is selected, and the 'Documents' tab is active. There are 4 documents in the collection. The documents are as follows:

_id	name	age	email
65ced2e973d8b37af8e307f1	"rasl11s"	20	"rasl11s@yahoo.gr"
65ced458c47ebc6c15a37b71	"Tassos"	60	"tassos@yahoo.gr"
65ced588c47ebc6c15a37c71	"Updated Maria"	21	"Maria3@yahoo.gr"
65ced6c4441d1fce83a333f71	"Gialtris"	23	"Gialtris@yahoo.gr"

Users after delete

The screenshot shows the MongoDB Compass interface for the 'smart_tourism' database. The 'users' collection is selected, and the 'Documents' tab is active. There are 3 documents in the collection. The documents are as follows:

_id	name	age	email
65ced2e973d8b37af8e307f1	"rasl11s"	20	"rasl11s@yahoo.gr"
65ced458c47ebc6c15a37b71	"Tassos"	60	"tassos@yahoo.gr"
65ced588c47ebc6c15a37c71	"Updated Maria"	21	"Maria3@yahoo.gr"

Delete User- Success

The screenshot shows the Postman interface with a workspace named "Smart_Tourism". The "Delete User By Email" collection is selected, and the "DELETE" method is chosen for the endpoint `http://localhost:8080/api/v1/users/delete?email=Manos@yahoo.gr`. The "Query Params" section shows a parameter `email` with the value `Manos@yahoo.gr`. The "Body" tab is selected, showing a JSON response:

```
1 {
2   "message": "User deleted successfully",
3   "deletedUser": {
4     "_id": "65cee1f7935d8ee5c7f365fb",
5     "name": "Updated Manos",
6     "age": 27,
7     "email": "manos@yahoo.gr",
8     "__v": 0
9   }
10 }
```

The status bar indicates a 200 OK response with a time of 86 ms and a size of 389 B. The console shows the following logs:

```
> GET http://localhost:8080/api/v1/users/all 200 59 ms
> GET http://localhost:8080/api/v1/users/?email=Manos@yahoo.gr 200 68 ms
> DELETE http://localhost:8080/api/v1/users/delete?email=Manos@yahoo.gr
```

Add New Poi

The screenshot shows the Postman interface with a workspace named "Smart_Tourism". The "Add New Poi" collection is selected, and the "POST" method is chosen for the endpoint `http://localhost:8080/api/v1/pois/add`. The "Body" tab is selected, showing a JSON request:

```
1 {
2   "name": "Asteroskopeio",
3   "category": "Mouseio",
4   "address": "Kalamata",
5   "coordinates": "Paraliaka",
6   "photo": "test4"
7 }
```

The status bar indicates a 201 Created response with a time of 131 ms and a size of 276 B. The console shows the following logs:

```
> GET http://localhost:8080/api/v1/users/?email=Manos@yahoo.gr 200 68 ms
> DELETE http://localhost:8080/api/v1/users/delete?email=Manos@yahoo.gr 200 86 ms
> POST http://localhost:8080/api/v1/pois/add
```

Get All Pois

The screenshot shows the Postman interface with a GET request to `http://localhost:8080/api/v1/pois/all`. The response is a JSON array of 4 POI objects. The status is 200 OK, time is 62 ms, and size is 830 B.

```
7 {
8   "coordinates": "Updated Kentro",
9   "photo": "test",
10  "___v": 0
11 },
12 {
13   "_id": "65ced86979a5d687b56a81e0",
14   "name": "Fisiko",
15   "category": "Museum",
16   "address": "Irakleio",
17   "coordinates": "Kentro",
18   "photo": "test2",
19   "___v": 0
20 },
21 {
22   "_id": "65ced88178a5d687b56a81e3",
23   "name": "Xiniko",
24   "category": "Museum",
25   "address": "Thessaloniki",
26   "coordinates": "Paraliaka",
27   "photo": "test3",
28   "___v": 0
29 },
30 {
31   "_id": "65ced88935d687b56a81e5",
32   "name": "Asteroskopeio",
33   "category": "Museum",
34   "address": "Kalamata",
35   "coordinates": "Paraliaka",
36   "photo": "test4",
37   "___v": 0
38 }
```

Get Poi By Name

The screenshot shows the Postman interface with a GET request to `http://localhost:8080/api/v1/pois/?name=fisiko`. The response is a single POI object. The status is 200 OK, time is 73 ms, and size is 377 B.

```
1 {
2   "_id": "65ced86979a5d687b56a81e0",
3   "name": "Fisiko",
4   "category": "Museum",
5   "address": "Irakleio",
6   "coordinates": "Kentro",
7   "photo": "test2",
8   "___v": 0
9 }
```


The screenshot shows the Postman interface with a REST client request for 'Get Poi By Category'. The request is a GET call to 'http://localhost:8080/api/v1/pois/category-mouseio'. The response is a JSON array of three poi objects. The first object has name 'Akiropoli', category 'Mouseio', address 'Athina', coordinates 'Updated Kentro', and photo 'test'. The second object has name 'Fisiko', category 'Mouseio', address 'Irakleio', coordinates 'Kentro', and photo 'test'. The third object has name 'Ximiko', category 'Mouseio', address 'Irakleio', coordinates 'Kentro', and photo 'test'.

The screenshot displays the Postman application interface. On the left, a sidebar shows a collection named 'Smart_LTourism' with various endpoints. The 'Get Poi By Address' endpoint is selected. The main panel shows the request details for a GET request to 'http://localhost:8080/api/v1/pois/?address=Athina'. The response body is visible, showing a JSON object with the following structure:

```

{
  "id": "60cedd979ac0d6979de81dd",
  "name": "Updated Akropolis",
  "category": "Museum",
  "address": "Athina",
  "coordinates": "Updated Kentro",
  "photo": "test",
  "_v": 0
}

```

The status bar at the bottom indicates a 200 OK response with a time of 58 ms and a size of 392 B. The console at the bottom shows the request logs.

Get Poi By Coordinates

The screenshot shows the Postman interface with a GET request to `http://localhost:8080/api/v1/pois/coordinates=Kentro`. The request is successful with a status of 200 OK, a time of 59 ms, and a size of 533 B. The response body is a JSON array of two objects, each representing a point of interest.

Query Params

Key	Value	Description
coordinates	Kentro	

Body

```
1 {
2   "_id": "65ced85d78e5d687b56a81dd",
3   "name": "Updated Akropolis",
4   "category": "Museum",
5   "address": "Athina",
6   "coordinates": "Updated Kentro",
7   "photo": "test",
8   "__v": 0
9 }
10
11 {
12   "_id": "65ced85d78e5d687b56a81e0",
13   "name": "Fisika",
14   "category": "Museum",
15   "address": "Irakleio",
16   "coordinates": "Kentro",
17   "photo": "test2",
18   "__v": 0
19 }
20 }
```

Console

```
> GET http://localhost:8080/api/v1/pois/category=museum 200 66 ms
> GET http://localhost:8080/api/v1/pois/address=Athina 200 58 ms
> GET http://localhost:8080/api/v1/pois/coordinates=Kentro 200 59 ms
```

Get Poi By Multiple Parameters

The screenshot shows the Postman interface with a GET request to `http://localhost:8080/api/v1/pois/name=fisika&coordinates=kentro`. The request is successful with a status of 200 OK, a time of 69 ms, and a size of 377 B. The response body is a JSON array of one object representing a point of interest.

Query Params

Key	Value	Description
name	fisika	
coordinates	kentro	

Body

```
1 {
2   "_id": "65ced86978e5d687b56a81e0",
3   "name": "Fisika",
4   "category": "Museum",
5   "address": "Irakleio",
6   "coordinates": "Kentro",
7   "photo": "test2",
8   "__v": 0
9 }
10
11 }
```

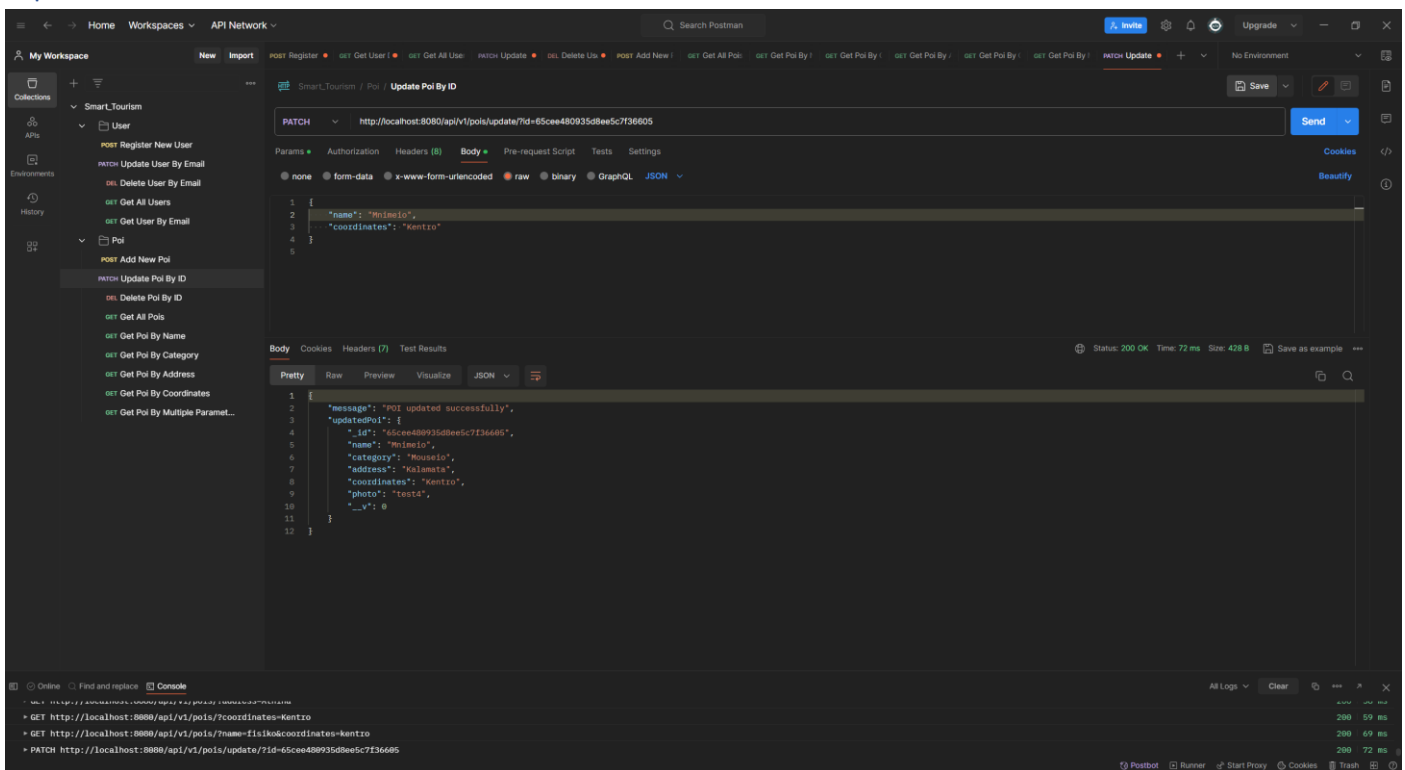
Console

```
> GET http://localhost:8080/api/v1/pois/address=Athina 200 58 ms
> GET http://localhost:8080/api/v1/pois/coordinates=Kentro 200 59 ms
> GET http://localhost:8080/api/v1/pois/name=fisika&coordinates=kentro 200 69 ms
```

Update Poi - Before

```
_id: ObjectId('65cee480935d8ee5c7f36605')
name: "Asteroskopeio"
category: "Mouseio"
address: "Kalamata"
coordinates: "Paraliaka"
photo: "test4"
__v: 0
```

Update User – Success



Update Poi - After

```
_id: ObjectId('65cee480935d8ee5c7f36605')
name: "Mnimeio"
category: "Mouseio"
address: "Kalamata"
coordinates: "Kentro"
photo: "test4"
__v: 0
```

Delete Poi - Before

smart_tourism.pois

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or [Generate query](#)

ADD DATA EXPORT DATA UPDATE DELETE

1-4 of 4

```
{ "_id": "63ce80597a0d0700a0a1a02", "name": "Updated Akropolis", "category": "Museum", "address": "Aktion", "coordinates": "Updated Kentro", "photo": "test", "__v": 0 }
```

```
{ "_id": "63ce80597a0d0700a0a1a03", "name": "Fistula", "category": "Museum", "address": "Iraklio", "coordinates": "Kentro", "photo": "test", "__v": 0 }
```

```
1 { "_id": "63ce80597a0d0700a0a1a03", "name": "Fistula", "category": "Museum", "address": "Iraklio", "coordinates": "Paralika", "photo": "test", "__v": 0 }
```

```
2 { "_id": "63ce80597a0d0700a0a1a04", "name": "Updated Akropolis", "category": "Museum", "address": "Aktion", "coordinates": "Updated Kentro", "photo": "test", "__v": 0 }
```

```
3 { "_id": "63ce80597a0d0700a0a1a05", "name": "Updated Akropolis", "category": "Museum", "address": "Aktion", "coordinates": "Updated Kentro", "photo": "test", "__v": 0 }
```

CANCEL UPDATE

Delete Poi - After

smart_tourism.pois

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } or [Generate query](#)

ADD DATA EXPORT DATA UPDATE DELETE

1-3 of 3

```
{ "_id": "63ce80597a0d0700a0a1a02", "name": "Updated Akropolis", "category": "Museum", "address": "Aktion", "coordinates": "Updated Kentro", "photo": "test", "__v": 0 }
```

```
{ "_id": "63ce80597a0d0700a0a1a03", "name": "Fistula", "category": "Museum", "address": "Iraklio", "coordinates": "Kentro", "photo": "test", "__v": 0 }
```

```
1 { "_id": "63ce80597a0d0700a0a1a04", "name": "Updated Akropolis", "category": "Museum", "address": "Aktion", "coordinates": "Updated Kentro", "photo": "test", "__v": 0 }
```

```
2 { "_id": "63ce80597a0d0700a0a1a05", "name": "Updated Akropolis", "category": "Museum", "address": "Aktion", "coordinates": "Updated Kentro", "photo": "test", "__v": 0 }
```

Delete Poi

My Workspace

Smart_Tourism

User

POST Register New User

PATCH Update User By Email

DELETE Delete User By Email

GET Get All Users

GET Get User By Email

Poi

POST Add New Poi

PATCH Update Poi By ID

DELETE Delete Poi By ID

GET Get All Ppis

GET Get Poi By Name

GET Get Poi By Category

GET Get Poi By Address

GET Get Poi By Coordinates

GET Get Poi By Multiple Paramet...

DELETE

http://localhost:8080/api/v1/pois/delete?id=65ced88170a5d607b50a81e3

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Query Params

☒

Key

Value

Description

Bulk Edit

☒

id

65ced88170a5d607b50a81e3

Key

Value

Description

Body

Cookies

Headers (7)

Test Results

Status: 200 OK

Time: 64 ms

Size: 434 B

Save as example

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

message: "POI deleted successfully",

deletedPoi: {

id: "65ced88170a5d607b50a81e3",

name: "Ximiko",

category: "Museum",

address: "Messaloniki",

coordinates: "Paralimni",

photo: "test3",

__v: 0

}

Online

Find and replace

Console

GET http://localhost:8080/api/v1/pois/?name=fisiko&coordinates=kantro 200 69 ms

PATCH http://localhost:8080/api/v1/pois/update/?id=65ced88170a5d607b50a81e3 200 72 ms

DELETE http://localhost:8080/api/v1/pois/delete/?id=65ced88170a5d607b50a81e3 200 64 ms

All Logs

Clear

Postbot

Runner

Start Proxy

Cookies

Trash