

Server Programming Project 3

Your job is to write a RESTful service in Node.js for a company to allow them to track timecards for employees. You are required to use the provided Data Layer (see separate documentation – same as for Projects 2). There is a separate zip file for the Data Layer (see below on how to include it in your app). You need to create the Service as per below, including any validation mentioned which **should be in** your Business Layer. You can put other things in your Business Layer if you wish. **You need to use your RIT user ID for the company name whenever it is asked for.** For error output, return an appropriate error message (**not** the String “An appropriate error message.”)

Service Layer:

All methods must return a JSON String which doesn't have to be **formatted** as in the samples below (in other words, with no carriage returns/line feeds/tabs) but must contain the same information. Some methods take JSON as input, others take Query Parameters or Form Parameters.

Open terminal or cmd

- Drag the **chukzProjectThree** into visual studio code.
- Click the terminal at the top tab and type: "node server.js"
- Your server.js app should now be running on localhost:8080.
- If you want to see all data on browse, you can see at http://localhost:8080/department
- Or open the postman and type in the url: localhost:8080 /CompanyServices/?company=ijc3093, it will be displayed: “user listening” meaning it is now running.
- Then please look at screens folder of the postman below and follow example:

COMPANY

The screenshot shows the Postman interface for a GET request to `localhost:8080/CompanyServices/`. The request is saved, and the response is displayed in the 'Body' tab. The response status is 200 OK, with a time of 36 ms and a size of 239 B. The response body is 'User Listing'.

KEY	VALUE	DESCRIPTION
Key	Value	Description

1 User Listing

DEPARTMENT

Overview

POST localhost:8080 /... GET museum.mysql.da... + ...

No Environment

localhost:8080 /CompanyServices/department

Save

POSTlocalhost:8080 /CompanyServices/department

Send

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettingsCookies

noneform-datax-www-form-urlencodedrawbinaryGraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	company	ijc3093			
<input checked="" type="checkbox"/>	dept_name	Information Science Technology			
<input checked="" type="checkbox"/>	dept_no	d10			
<input checked="" type="checkbox"/>	location	Rochester			
	Key	Value	Description		

BodyCookiesHeaders (7)Test Results

Status: 200 OKTime: 1226 msSize: 376 BSave Response

PrettyRawPreviewVisualizeJSON

123456789

```
"success": {
  "company": "ijc3093",
  "dept_id": 113,
  "dept_name": "Information Science Technology ",
  "dept_no": "d10_ijc3093",
  "location": "Rochester"
}
```

localhost:8080 /CompanyServices/departments?company=ijc3093

Save

GETlocalhost:8080 /CompanyServices/departments?company=ijc3093

Send

ParamsAuthorizationHeaders (6)BodyPre-request ScriptTestsSettingsCookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	company	ijc3093			
	Key	Value	Description		

BodyCookiesHeaders (7)Test Results

Status: 200 OKTime: 216 msSize: 588 BSave Response

PrettyRawPreviewVisualizeJSON

123456789101112131415161718192021222324

```
"success": [
  {
    "company": "ijc3093",
    "dept_id": 112,
    "dept_name": "mystery",
    "dept_no": "ijc3093-bdIfT_ijc3093",
    "location": "RIT"
  },
  {
    "company": "ijc3093",
    "dept_id": 113,
    "dept_name": "Information Science Technology ",
    "dept_no": "d10_ijc3093",
    "location": "Rochester"
  },
  {
    "company": "ijc3093",
    "dept_id": 114,
    "dept_name": "IST",
    "dept_no": "d11_ijc3093",
    "location": "Rochester"
  }
]
```

GET

localhost:8080 /CompanyServices/department?company=ijc3093&dept_id=112

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	company	ijc3093			
<input checked="" type="checkbox"/>	dept_id	112			
	Key	Value	Description		

Body

Cookies

Headers (7)

Test Results

Status: 200 OK Time: 215 ms Size: 355 B Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1  {
2    "success": {
3      "company": "ijc3093",
4      "dept_id": 112,
5      "dept_name": "mystery",
6      "dept_no": "ijc3093-bdfT_ijc3093",
7      "location": "RIT"
8    }
9  }
```

localhost:8080 /CompanyServices/department

Save

PUT

localhost:8080 /CompanyServices/department

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

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

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

<

EMPLOYEE

POST

localhost:8080 /CompanyServices/employee?company=ijc3093&emp_name=Izz Chuzz&emp_no=ijc3093_e1b&hire_date=2021-09-10&job=Web Developer&dept_id=114

Send

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

none form-data x-www-form-urlencoded raw binary GraphQL

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	company	ijc3093			
<input checked="" type="checkbox"/>	emp_name	Izz Chuzz			
<input checked="" type="checkbox"/>	emp_no	ijc3093_e2b			
<input checked="" type="checkbox"/>	hire_date	2021-09-10			
<input checked="" type="checkbox"/>	job	Web Developer			
<input checked="" type="checkbox"/>	salary	60,000			
<input checked="" type="checkbox"/>	dept_id	113			
<input checked="" type="checkbox"/>	mng_id	1			

Body Cookies Headers (7) Test Results

Status: 200 OK Time: 417 ms Size: 398 B Save Response

Pretty Raw Preview Visualize JSON

```
1  {
2    "success": {
3      "emp_id": 4,
4      "emp_name": "Izz Chuzz",
5      "emp_no": "ijc3093_e2b_ijc3093",
6      "hire_date": "2021-09-10",
7      "job": "Web Developer",
8      "salary": 60,
9      "dept_id": 113,
10     "mng_id": 1
11   }
12 }
```

GET

localhost:8080 /CompanyServices/employees?company=ijc3093

Send

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

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	company	ijc3093			
	Key	Value	Description		

Body Cookies Headers (7) Test Results

Status: 200 OK Time: 304 ms Size: 555 B Save Response

Pretty Raw Preview Visualize JSON

```
1  {
2    "success": [
3      {
4        "emp_id": 4,
5        "emp_name": "Izz Chuzz",
6        "emp_no": "ijc3093_e2b_ijc3093",
7        "hire_date": "2021-09-10",
8        "job": "Web Developer",
9        "salary": 60,
10       "dept_id": 113,
11       "mng_id": 1
12     },
13     {
14       "emp_id": 1,
15       "emp_name": "Izz Chuzz",
16       "emp_no": "ijc3093_e1b_ijc3093",
17       "hire_date": "2021-09-10",
18       "job": "Web Developer",
19       "salary": 60,
20       "dept_id": 114,
21       "mng_id": null
22     }
23   ]
24 }
```

GET

localhost:8080 /CompanyServices/employee?company=ijc3093&emp_id=4

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> company	ijc3093			
<input checked="" type="checkbox"/> emp_id	4			
Key	Value	Description		

Body

Cookies

Headers (7)

Test Results

Status: 200 OK Time: 257 ms Size: 398 B Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1 1
2 2
3 3
4 4
5 5
6 6
7 7
8 8
9 9
10 10
11 11
12 12
{
  "success": {
    "emp_id": 4,
    "emp_name": "Izz Chuzz",
    "emp_no": "ijc3093_e2b_ijc3093",
    "hire_date": "2021-09-10",
    "job": "Web Developer",
    "salary": 60,
    "dept_id": 113,
    "mng_id": 1
  }
}
```

PUT

localhost:8080 /CompanyServices/employee

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

Body

Cookies

Headers (7)

Test Results

Status: 200 OK Time: 642 ms Size: 403 B Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1 1
2 2
3 3
4 4
5 5
6 6
7 7
8 8
9 9
10 10
11 11
12 12
{
  "company": "ijc3093",
  "emp_id": 4,
  "emp_name": "Izz Chuzz",
  "emp_no": "ijc3093_e2b_ijc3093",
  "hire_date": "2021-09-10",
  "job": "Web Programming",
  "salary": 60000,
  "dept_id": 113,
  "mng_id": 1
}
```

TIMECARD

POST

localhost:8080 /CompanyServices/timecard

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

<input checked="" type="checkbox"/>	company	ijc3093	
<input checked="" type="checkbox"/>	emp_id	4	
<input checked="" type="checkbox"/>	start_time	2021-09-10	
<input checked="" type="checkbox"/>	end_time	2021-12-15	
	Key	Value	Description

Body

Cookies

Headers (7)

Test Results

Status: 200 OK

Time: 457 ms

Size: 344 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1
2  "success": {
3    "emp_id": 4,
4    "timecard_id": 2,
5    "start_time": "2021-12-19 23:47:49",
6    "end_time": "2021-12-19 23:47:49"
7  }
8
```

GET

localhost:8080 /CompanyServices/timecards?company=ijc3093&emp_id=4

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

KEY	VALUE	DESCRIPTION		Bulk Edit
<input checked="" type="checkbox"/>	company	ijc3093		
<input checked="" type="checkbox"/>	emp_id	4		
	Key	Value		Description

Body

Cookies

Headers (7)

Test Results

Status: 200 OK

Time: 330 ms

Size: 443 B

Save Response

Pretty

Raw

Preview

Visualize

JSON

```
1
2  "success": [
3    {
4      "emp_id": 4,
5      "timecard_id": 1,
6      "start_time": "2021-12-19 23:46:53",
7      "end_time": "2021-12-19 23:46:53"
8    },
9    {
10     "emp_id": 4,
11     "timecard_id": 2,
12     "start_time": "2021-12-19 23:47:49",
13     "end_time": "2021-12-19 23:47:49"
14   }
15 ]
16
```

GET localhost:8080 /CompanyServices/timecard?company=ijc3093&timecard_id=2

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

Query Params

KEY	VALUE	DESCRIPTION
company	ijc3093	
timecard_id	2	

Body Cookies Headers (7) Test Results

Status: 200 OK Time: 215 ms Size: 344 B Save Response

Pretty Raw Preview Visualize JSON

```

1  {
2    "success": {
3      "emp_id": 4,
4      "timecard_id": 2,
5      "start_time": "2021-12-19 23:47:49",
6      "end_time": "2021-12-19 23:47:49"
7    }
8  }

```

All method signatures must match the ones listed.

You may have to use multiple Data Layer methods to accomplish each Service Layer method.

General input validation: Refer to the EER Diagram for the database for datatypes and sizes. Any additional validation/business rules will be listed below in the appropriate method.

- 1) **Root Path for Service Layer: "CompanyServices"**
- 2) **Server should listen on port 8080**
- 3) The remaining paths will be appended to the above, e.g.
localhost:8080 /CompanyServices/
- 4) **Path: /company**

Verb: DELETE

Produces: application/json

- a. Deletes all Department, Employee and Timecard records in the database for the given company. You will need to pay attention to the Foreign Key Constraints.
- b. Input is your RIT user ID as a String passed as **QueryParam**
 - i. company=company+name
where "company+name" is your RIT user ID
- c. Output:

- i. Success:

1. A JSON String:

```
{"success":"companyName's information deleted."}
```

2. A JSON String:

```
{"error":"An appropriate error message."}
```

5) Path: /department

Verb: GET

Produces: application/json

- a. Returns the requested Department as a JSON String.
- b. Input as **QueryParams:**

company=company+name&dept_id= id

where “company+name” is your RIT user ID and
“dept_id” is the record id of the department to retrieve.

c. Output:

i. Success:

1. A JSON String:

```
{  
    "dept_id":1,  
    "company":"rituserid",  
    "dept_name":"accounting",  
    "dept_no":"d10",  
    "location":"new york"  
}
```

2. A JSON String:

```
{"error":"An appropriate error message."}
```

6) Path: /departments

Verb: GET

Produces: application/json

- a. Returns the requested list of Departments.
- b. Input is your RIT user ID as a String in a **QueryParam**.
 - i. company=company+name
where “company+name” is your RIT user ID
- c. Output:

- i. Success:

- 1. A JSON String:

```
[
  {
    "dept_id":1,
    "company":"rituserid",
    "dept_name":"accounting",
    "dept_no":"d10",
    "location":"new York"
  },
  {
    "dept_id":2,
    "company":"rituserid",
    "dept_name":"research",
    "dept_no":"d20",
    "location":"dallas"
  },
  {
    "dept_id":3,
    "company":"rituserid",
    "dept_name":"sales",
    "dept_no":"d30",
    "location":"chicago"  },
  {
    "dept_id":4,
    "company":"rituserid",
    "dept_name":"operations",
    "dept_no":"d40",
    "location":"boston"  }
]
```

2. A JSON String:

```
{"error":"An appropriate error message."}
```

7) Path: /department

Verb: PUT

Consumes: application/json

Produces: application/json

a. Additional Validation:

i. dept_no must be unique among all companies, Suggestion: include company name as part of id.

ii. dept_id must be an existing record number for a department

b. Returns the updated Department as a JSON String.

c. Input: **JSON String** (Input any values you want to change plus the record id for the Department)

```
{  
  "company":"rituserid",  
  "dept_id":5,  
  "dept_name":"IT",  
  "dept_no":"d11",  
  "location":"rochester"  
}
```

where "company" is your RIT user ID.

d. Output:

i. Success:

1. A JSON String:

```
{  
  "success":{  
    "dept_id":5,
```

```
    "company":"rituserid",
    "dept_name":"IT",
    "dept_no":"d11",
    "location":"rochester"  }
}
```

2. A JSON String:

```
{"error":"An appropriate error message."}
```

8) Path: /department

Verb: POST

Produces: application/json

- a. Additional Validation:
 - i. dept_no must be unique among all companies, Suggestion: include company name as part of id.
- b. Returns the new Department as a JSON String.
- c. Input as FormParam:

```
"company" = "rituserid"
"dept_name" = "mystery"
"dept_no" = "d10"
"location" = "buffalo"
```

where “company” is your RIT user ID.

- c. Output:
 - i. Success:
 - 1. A JSON String:

```
{
  "success":{
    "dept_id":1,
    "company":"rituserid",
    "dept_name":"mystery",
```

```
"dept_no":"d10",  
"location":"buffalo" }  
}
```

2. A JSON String:

```
{"error":"An appropriate error message."}
```

9) Path: /department

Verb: DELETE

Produces: application/json

- a. Returns the number of rows deleted.
- b. Input as **QueryParam:**

```
"company" = "company name"  
"dept_id" = id
```

where “company name” is your RIT user ID and
“id” is the record id of the department to delete.

c. Output:

i. Success:

1. A JSON String:

```
{  
  "success":"Department 5 from rituserid deleted."  
}
```

2. A JSON String:

```
{"error":"An appropriate error message."}
```

10) Path: /employee

Verb: GET

Produces: application/json

- a. Returns the requested Employee as a JSON String.
- b. Input: the record id of the desired Employee as a **QueryParam**
 - i. company=company+name
where "company+name" is your RIT user ID
 - ii. emp_id=#
- c. Output:
 - i. Success:
 - 1. A JSON String:


```
{
                "emp_id":2,
                "emp_name":"jones",
                "emp_no":"e2",
                "hire_date":"1981-04-01",
                "job":"manager",
                "salary":2975.0,
                "dept_id":2,
                "mng_id":1}
```
 - 2. A JSON String:


```
{"error":"An appropriate error message."}
```

11) Path: /employees

Verb: GET

Produces: application/json

- a. Returns the requested list of Employees.
- b. Input is your RIT user ID as a String as a **QueryParam**.
 - i. company=company+name
where "company+name" is your RIT user ID
- c. Output:
 - i. Success:
 - 1. A JSON String:


```
[
  {
```

```
"emp_id":1,
"emp_name":"king",
"emp_no":"e1",
"hire_date":"1981-11-16",
"job":"president",
"salary":5000.0,
"dept_id":1,
"mng_id":0
},
{
  "emp_id":2,
  "emp_name":"jones",
  "emp_no":"e2",
  "hire_date":"1981-04-01",
  "job":"manager",
  "salary":2975.0,
  "dept_id":2,
  "mng_id":1
},
{
  "emp_id":3,
  "emp_name":"ford",
  "emp_no":"e3",
  "hire_date":"1981-12-02",
  "job":"analyst",
  "salary":3000.0,
  "dept_id":2,
  "mng_id":2
},
{
  "emp_id":4,
  "emp_name":"smith",
  "emp_no":"e4",
  "hire_date":"1980-12-16",
  "job":"clerk",
  "salary":800.0,
  "dept_id":2,
  "mng_id":2
}
```

```
},
{
  "emp_id":5,
  "emp_name":"blake",
  "emp_no":"e5",
  "hire_date":"1981-04-30",
  "job":"manager",
  "salary":2850.0,
  "dept_id":3,
  "mng_id":1 },
{
  "emp_id":6,
  "emp_name":"allen",
  "emp_no":"e6",
  "hire_date":"1981-02-19",
  "job":"salesman",
  "salary":1600.0,
  "dept_id":3,
  "mng_id":5 },
{
  "emp_id":7,
  "emp_name":"ward",
  "emp_no":"e7",
  "hire_date":"1981-02-21",
  "job":"salesman",
  "salary":1250.0,
  "dept_id":3,
  "mng_id":5
},
{
  "emp_id":8,
  "emp_name":"martin",
  "emp_no":"e8",
  "hire_date":"1981-09-27",
  "job":"salesman",
  "salary":1250.0,
  "dept_id":3,
  "mng_id":5
```

```

    },
    {
      "emp_id":9,
      "emp_name":"clark",
      "emp_no":"e9",
      "hire_date":"1981-06-08",
      "job":"manager",
      "salary":2450.0,
      "dept_id":3,
      "mng_id":1 }
  ]

```

2. A JSON String:

```
{"error":"An appropriate error message."}
```

12) Path: /employee

Verb: POST

Consumes: Form Parameters

Produces: application/json

- a. Additional validations:
 - i. company – must be your RIT username
 - ii. dept_id must exist as a Department in your company
 - iii. mng_id must be the record id of an existing Employee in your company. Use 0 if the first employee or any other employee that doesn't have a manager.
 - iv. hire_date must be a valid date equal to the current date or earlier (e.g. current date or in the past)
 - v. hire_date must be a Monday, Tuesday, Wednesday, Thursday or a Friday. It **cannot** be Saturday or Sunday.
 - vi. emp_no must be unique amongst all employees in the database, **including** those of other companies. You may wish to include your RIT user ID in the employee number somehow.
- b. Returns the new Employee as a JSON String.

c. Input as FormParam:

```
"company"="yourRITid",  
  "emp_name"="french",  
  "emp_no"="rituserid-e1b",  
  "hire_date"="2018-06-16",  
  "job"="programmer",  
  "salary"=5000.0,  
  "dept_id"=1,  
  "mng_id"=2
```

d. Output:

i. Success:

1. A JSON String:

```
{  
  "success":{  
    "emp_id":15,  
    "emp_name":"french",  
    "emp_no":"rituserid-e1b",  
    "hire_date":"2018-06-16",  
    "job":"programmer",  
    "salary":5000.0,  
    "dept_id":1,  
    "mng_id":2  }  
}
```

2. A JSON String:

```
{"error":"An appropriate error message."}
```

13) Path: /employee

Verb: PUT

Consumes: application/json

Produces: application/json

- a. Additional validations same as inserting an Employee plus emp_id must be a valid record id in the database.
- b. Returns the updated Employee as a JSON String.
- c. Input(any values you want to change plus the record id for the Employee) as **JSON string (company+name is your RIT username)**:

```
{
  "company ":company+name,
  "emp_id":15,
  "emp_name":"french",
  "emp_no":"rituserid-e1b",
  "hire_date":"2018-06-16",
  "job":"programmer",
  "salary":6000.0,
  "dept_id":1,
  "mng_id":2
}
```

- d. Output:

- i. Success:

1. A JSON String:

```
{
  "success":{
    "emp_id":15,
    "emp_name":"french",
    "emp_no":"rituserid-e1b",
    "hire_date":"2018-06-16",
    "job":"programmer",
    "salary":6000.0,
    "dept_id":1,
    "mng_id":2 }
}
```

2. A JSON String:

```
{"error":"An appropriate error message."}
```

14) Path: /employee

Verb: DELETE

Produces: application/json

- a. Returns the that the employee deleted.
- b. Input: the record id of the Employee to delete as a **QueryParam**.
 - i. company=company+name
where “company+name” is your RIT user ID
 - ii. emp_id=#
- c. Output:
 - i. Success:
 - 1. A JSON String:

```
{  
  "success": "Employee 15 deleted."  
}
```
 - 2. A JSON String:

```
{"error": "An appropriate error message."}
```

15) Path: /timecard

Verb: GET

Produces: application/json

- a. Returns the requested Timecard as a JSON String.
- b. Input: the record id of the desired Timecard as a **QueryParam**
 - i. company=company+name
where “company+name” is your RIT user ID
 - ii. timecard_id=#
- c. Output:
 - i. Success:
 - 1. A JSON String:

```
{
  "timecard":{
    "timecard_id":1,
    "start_time":"2018-06-14 11:30:00",
    "end_time":"2018-06-14 15:30:00",
    "emp_id":2
  }
}
```

2. A JSON String:

```
{"error":"An appropriate error message."}
```

16) Path: /timecards

Verb: GET

Produces: application/json

- a. Returns the requested list of Timecards.
- b. Input is the record id of the employee you want to see the Timecards for as a **QueryParam**.

- i. company=company+name

where "company+name" is your RIT user ID

- ii. emp_id=#

c. Output:

- i. Success:

1. A JSON String:

```
[
  {
    "timecard_id":3,
    "start_time":"2018-06-14 11:30:00",
    "end_time":"2018-06-14 15:30:00",
    "emp_id":4
  },
  {
    "timecard_id":4,
    "start_time":"2018-06-13 11:30:00",
```

```
        "end_time":"2018-06-13 15:30:00",
        "emp_id":4
    },
    {
        "timecard_id":6,
        "start_time":"2018-06-12 11:30:00",
        "end_time":"2018-06-12 15:30:00",
        "emp_id":4 }
]
```

2. A JSON String:

```
{"error":"An appropriate error message."}
```

17) Path: /timecard

Verb: POST

Consumes: form parameters

Produces: application/json

a. Additional validations:

- i. company must be your RIT id
- ii. emp_id must exist as the record id of an Employee in your company.
- iii. start_time must be a valid date and time equal to the current date or back to the Monday prior to the current date if the current date is not a Monday.
- iv. end_time must be a valid date and time at least 1 hour greater than the start_time and be on the same day as the start_time.
- v. start_time and end_time must be a Monday, Tuesday, Wednesday, Thursday or a Friday. They **cannot** be Saturday or Sunday.
- vi. start_time and end_time must be between the hours (in 24 hour format) of 08:00:00 and 18:00:00 inclusive.
- vii. start_time must not be on the same day as any other start_time for that employee.

- b. Returns the new Timecard as a JSON String.
- c. Input all Timecard values as **FormParams**:

```

"company="your RIT ID",
    "emp_id"=1,
"start_time"="2018-06-15 12:30:00",
"end_time"="2018-06-15 15:30:00"

```

- d. Output:

- i. Success:

- 1. A JSON String:

```

{
  "success":{
    "timecard_id":1,
    "start_time":"2018-06-14 11:30:00",
    "end_time":"2018-06-14 15:30:00",
    "emp_id":2
  }
}

```

- 2. A JSON String:

```

{"error":"An appropriate error message."}

```

18) Path: /timecard

Verb: PUT

Consumes: application/json

Produces: application/json

- a. Additional validations same as inserting a Timecard plus timecard_id must be a valid record id in the database.
- b. Returns the updated Timecard as a JSON String.
- c. Input(any values you want to change plus the record id for the Timecard) as **JSON string (company is your RIT username)**:

```
{
  "company":"your RIT ID",
    timecard_id":2,
  "start_time":"2018-06-14 11:30:00",
  "end_time":"2018-06-14 15:30:00",
  "emp_id":1
}
```

d. Output:

i. Success:

1. A JSON String:

```
{
  "success":{
    "timecard_id":0,
    "start_time":"2018-06-15 12:30:00",
    "end_time":"2018-06-15 15:30:00",
    "emp_id":2
  }
}
```

2. A JSON String:

```
{"error":"An appropriate error message."}
```

19) Path: /timecard

Verb: DELETE

Produces: application/json

- a. Returns the number of rows deleted.
- b. Input: the record id of the Timecard to delete as a **QueryParam**.
 - i. company=company+name
where "company+name" is your RIT user ID
 - ii. timecard_id=#

c. Output:

i. Success:

1. A JSON String:

```
{  
  "success": "Timecard 1 deleted."  
}
```

2. A JSON String:

```
{"error": "An appropriate error message."}
```

Deliverables:

Put the following in the dropbox for Project 3 by the due date on the dropbox:

1) A zip file of all of your source files. (don't zip up your node_modules folder)

Hints:

1) In addition to the Data Layer, use npm install --save when installing.

2) To convert from Timestamp to String (for putting in JSON String), take a look at the date-fns module (format method) or the moment.js module.

3) When creating a Timecard, use the string representation of the date as a parameter to the constructor in the format of yyyy-mm-dd hh:mm:ss

4) Use: app.use(express.json()) for processing JSON body input to get the fields from req.body

5) Use: app.use(express.urlencoded({extended:false})) for processing POST form input to get the fields from req.body

6) App structure for using the provided Data Layer:

- Unzip the companydata.zip file
- Make a directory for your node project

- “touch server.js”
- Copy the companydata folder created by the unzip above into this folder
- npm init
- npm install --save <path to the unzipped company folder>
- in server.js add:
 - var DataLayer = require("./companydata/index.js");
 - var dl = new DataLayer("yourusername");
 - Continue with the rest of your code.
 - To create a Department/Employee/Timecard: new dl.Department(...), new dl.Employee(...), new Timecard(...)

- 7) To test using Postman:
- a. For DELETE method, make sure any text under raw/body is deleted, all form fields are unchecked and uncheck any header fields and the correct id is set as a parameter.
 - b. For POST method, select x-www-form-urlencoded with fields for each item you want to pass in.
 - c. For PUT, make sure Content-Type header = application/json

Rubric:

	Possible Points	Actual Points
All required methods with correct inputs and outputs:	40	
All validations in Business Layer:	25	
Appropriate error messages:	5	
Correct Node.js structure and it runs:	15	

Good code structure (DRY, etc):	15	
Total:	100	