

Enterprise Application Development

Lab 1

Catherine Kane, C15902129
Submission: 17th February 2019

Question 1

Code snippet to get all users:

```
/* GET /users
List all users email and sex in order of most recently created. Do not include password hash in your output
*/
app.get('/users', (req, res) => db.query("SELECT email, details FROM users ORDER BY created_at DESC").then( resp =>
{
    res.send(resp);
}))
```

Output of all users:

← → ↻ ⓘ 127.0.0.1:3000/users

```
[{"email":"Shari.Julian@yahoo.com","details":{"sex":"M"},"email":"Evelyn.Patnode@gmail.com","details":{"sex":"M"},"email":"Layne.Sarver@aol.com","details":{"sex":"M"},"email":"Quinton.Gilpatrick@yahoo.com","details":{"sex":"M"},"email":"Graciela.Kubala@yahoo.com","details":{"sex":"F"},"email":"Derek.Knittel@gmail.com","details":{"sex":"F"},"email":"Theresia.Edwin@yahoo.com","details":{"sex":"M"},"email":"Williams.Upson@gmail.com","details":{"sex":"F"},"email":"Glen.Langhear@yahoo.com","details":null,"email":"Ozella.Yoshimura@gmail.com","details":{"sex":"M"},"email":"Samatha.Hedgpeth@yahoo.com","details":{"sex":"F"},"email":"Isabel.Breeding@gmail.com","details":null,"email":"Kali.Damore@yahoo.com","details":{"sex":"F"},"email":"Gudrun.Arends@gmail.com","details":null,"email":"Ivana.Kurth@yahoo.com","details":null,"email":"Kimi.Mcqueeney@gmail.com","details":{"email":"Stacia.Schrack@aol.com","details":{"sex":"M"},"email":"Claud.Westmoreland@aol.com","details":null,"email":"Eleanor.Patnode@yahoo.com","details":{"sex":"F","state":"Florida"},"email":"Sherilyn.Hamill@gmail.com","details":{"email":"Russ.Mcclain@yahoo.com","details":{"sex":"F"},"email":"Tonette.Alba@gmail.com","details":null,"email":"Earlean.Bonaccif","email":"Shanell.Maxson@gmail.com","details":{"sex":"M"},"email":"Ozella.Roles@gmail.com","details":{"sex":"F"},"email":"Wan.Dilks@gmail.com","details":{"sex":"M"},"email":"Vivian.Westmoreland","state":"South Carolina"},"email":"Humberto.Jonson@yahoo.com","details":null,"email":"Salvatore.Arends@aol.com","details":{"sex":"F","state":"Virginia"},"email":"Angel.Lessley@yahoo","email":"Eve.Kump@yahoo.com","details":{"sex":"M"},"email":"Mauro.Pung@yahoo.com","details":{"sex":"F","state":"Illinois"},"email":"Claud.Cousineau@gmail.com","details":{"sex":"F","email":"Zita.Breeding@gmail.com","details":null,"email":"Granville.Hedgpeth@gmail.com","details":null,"email":"Harrison.Puett@yahoo.com","details":{"sex":"M"},"email":"Cortney.Strayer@gmail","state":"Virginia"},"email":"Edmund.Roles@yahoo.com","details":{"sex":"F","state":"New York"},"email":"Carmel.Bulfer@aol.com","details":{"sex":"F","state":"Florida"},"email":"Wynona.Greening@aol.com","details":{"sex":"M"},"email":"Shanell.Lichtenstein@aol.com","details":{"sex":"M"},"email":"Danny.Crays@gmail.com","details":{"sex":"M"},"email":"Samatha.Pellegrin@yahoo.com","details":null,"email":"Jeremiah.Buonocore@yahoo.com","details":null,"email":"Derek.Crenshaw@gmail.com","details":{"sex":"F"},"email":"Takako.Gilpatrick","email":"Zita.Luman@yahoo.com","details":null,"email":"CherryL.Farnowski@gmail.com","details":null,"email":"Ivana.Sosnowski@aol.com","details":{"sex":"M"},"email":"Romaine.Birdsell@aol.com",
```

Code snippet to get a specific user:

```
/*
GET /users/:id
Show above details of the specified user
*/
app.get('/users/:id', (req, res) => {
    //retrieve id
    let user_id = req.params.id;

    db.users.find({id: user_id}, {}).then(
        user => {
            res.send(user);
        }
    );
});
```

Output of retrieving a specified user:

← → ↻ ⓘ 127.0.0.1:3000/users/2

```
[{"id":2,"email":"Evelyn.Patnode@gmail.com","password":"d678656644a3f44023f90e4f1cace1f4","details":{"sex":"M"},"created_at":"2010-11-12T21:27:00.000Z","deleted_at":null,"updatedAt":null}]
```

Code snippet to list all products:

```
/*GET /products
List all products in ascending order of price*/
app.get('/products', (req, res) => db.query("SELECT * FROM products ORDER BY price ASC").then( resp =>
{
    res.send(resp);
}))
```

Output of all products:

← → ↺ ⓘ 127.0.0.1:3000/products

```
[{"id":5,"title":"Coloring Book","price":5.99,"createdAt":"2011-01-01T20:00:00.000Z","deleted_at":null,"tags":["Book","Children"],"updatedAt":null},{"id":4,"title":"Baby Book","price":7.99,"createdAt":"2011-01-01T20:00:00.000Z","deleted_at":null,"tags":["Book","Children","Baby"],"updatedAt":null},{"id":11,"title":"Classical CD","price":9.99,"createdAt":"2011-01-01T20:00:00.000Z","deleted_at":null,"tags":["Music"],"updatedAt":null},{"id":12,"title":"Holiday CD","price":9.99,"createdAt":"2011-01-01T20:00:00.000Z","deleted_at":null,"tags":["Music"],"updatedAt":null},{"id":13,"title":"Country CD","price":9.99,"createdAt":"2011-01-01T20:00:00.000Z","deleted_at":null,"tags":["Music"],"updatedAt":null},{"id":14,"title":"Electronic CD","price":9.99,"createdAt":"2011-01-01T20:00:00.000Z","deleted_at":null,"tags":["Music"],"updatedAt":null},{"id":15,"title":"Romantic","price":14.99,"createdAt":"2011-01-01T20:00:00.000Z","deleted_at":null,"tags":["Movie","Comedy"],"updatedAt":null},{"id":17,"title":"Documentary","price":14.99,"createdAt":"2011-01-01T20:00:00.000Z","deleted_at":null,"tags":["Movie"],"updatedAt":null},{"id":18,"title":"Action","price":14.99,"createdAt":"2011-01-01T20:00:00.000Z","deleted_at":null,"tags":["Movie"],"updatedAt":null},{"id":20,"title":"Python Book","price":27.99,"createdAt":"2011-01-01T20:00:00.000Z","deleted_at":null,"tags":["Book","Programming","Python"],"updatedAt":null},{"id":8,"title":"MP3 Player","price":108,"createdAt":"2011-01-01T20:00:00.000Z","deleted_at":null,"tags":["Technology","TV"],"updatedAt":null},{"id":6,"title":"Desktop Computer","price":499.99,"createdAt":"2011-01-01T20:00:00.000Z","deleted_at":null,"tags":["Technology"],"updatedAt":null}]]
```

Code snippet to get product by id:

```
/*
GET /products/:id
Show above details of the specified products
*/
app.get('/products/:id', (req, res) => {
    let pid = req.params.id;

    db.products.find({id: pid}, {}).then(
        prod => {
            res.send(prod);
        })
    });
```

Output of product retrieved by id:

← → ↺ ⓘ 127.0.0.1:3000/products/4

```
[{"id":4,"title":"Baby Book","price":7.99,"createdAt":"2011-01-01T20:00:00.000Z","deleted_at":null,"tags":["Book","Children","Baby"],"updatedAt":null}]
```

Code snippet to list purchase items using information from purchases and user:

```
/*
GET /purchases
List purchase items to include the receiver's name and,
address, the purchaser's email address and the price,
quantity and delivery status of the purchased item. Order by price in descending order
*/
app.get('/purchases', (req, res) => db.query(" SELECT " +
"      p.name\n" +
"      ,p.address\n" +
"      , u.email\n" +
"      , pi.price\n" +
"      , pi.quantity\n" +
"      , pi.state\n" +
" FROM purchase_items pi\n" +
" INNER JOIN purchases p\n" +
" on pi.purchase_id = p.id\n" +
" INNER JOIN users u\n" +
" on p.user_id = u.id").then( resp =>
{
    res.send(resp);
}))
```

Output of query - printed this to console for better readability of the output.

```
email: 'Gudrun.Arends@gmail.com',
price: '899.99',
quantity: 1,
state: 'Delivered' },
{ name: 'Divina Hamill',
  address: '2103 50th Ave.',
  email: 'Gudrun.Arends@gmail.com',
  price: '7.99',
  quantity: 1,
  state: 'Delivered' },
{ name: 'Divina Hamill',
  address: '2103 50th Ave.',
  email: 'Gudrun.Arends@gmail.com',
  price: '9.99',
  quantity: 1,
  state: 'Delivered' },
{ name: 'Romaine Coderre',
  address: '6990 Washington Ave.',
  email: 'Sherilyn.Hamill@gmail.com',
  price: '7.99',
  quantity: 1,
  state: 'Delivered' },
{ name: 'Kourtney Julian',
  address: '8277 44th Ave.',
  email: 'Carmel.Bulfer@aol.com',
  price: '7.99',
  quantity: 2,
  state: 'Delivered' },
{ name: 'Danyel Styers',
  address: '8464 8th Ave.',
  email: 'Danny.Crays@gmail.com',
  price: '9.99',
  quantity: 4,
  state: 'Delivered' },
```

Question 2

Code snippet, get products by name:

```
/*
GET /products[?name=string]
extend the product indexing endpoint to allow the filtering of products by name as follows
*/
app.get('/products', (req, res) => {

  // Retrieve the tag from our URL path
  let query_string = req.query.name;
  //check if query string
  //[?name=string]
  if(query_string === undefined)
  {
    db.products.find({}, {}).then(
      product => {
        res.send(product);
      }
    )
  }
  else{
    db.products.find({title: query_string}, {}).then(
      product => {
        res.send(product);
      }
    )
  }
})
```

Testing the endpoint with the following code to show it returns the correct object

← → ↻ ⓘ 127.0.0.1:3000/products?name=Dictionary

```
[{"id":1,"title":"Dictionary","price":"9.99","created_at":"2011-01-01T20:00:00.000Z","deleted_at":null,"tags":["Book"]}]
```

```
//example of SQL injection
app.get('/products/inject', (req, res) => {
  let query_string = req.query.name;

  db.query("SELECT * FROM products WHERE title= '" + query_string + "' ").then( resp => {
    res.send(resp);
  })
})
```

Ran the following SQL inject code

<http://127.0.0.1:3000/products/inject?name=INJECTED>: UPDATE products SET title='Dictionary' WHERE title='INJECTED

Object has now been renamed

← → ↻ ⓘ 127.0.0.1:3000/products?name=INJECTED

```
[{"id":1,"title":"INJECTED","price":"9.99","created_at":"2011-01-01T20:00:00.000Z","deleted_at":null,"tags":["Book"]}]
```

Dictionary no longer exists

← → ↻ ⓘ 127.0.0.1:3000/products?name=Dictionary

```
[]
```

Changed this back to Dictionary from Injected

← → ↻ ⓘ 127.0.0.1:3000/products/inject?name=INJECTED%27;%20UPDATE%20products%20SET%20title=%27Dictionary%27%20WHERE%20title=%27INJECTED

Ooops

This is also showing how I was able to obtain all of the emails of users by running email='x'
OR 'x'='x

```
{
  "id": 1, "email": "Earlean.Bonaccii@yahoo.com", "password": "029761dd44fec0b14825843add0dfce", "details": null, "created_at": "2009-12-20T20:36:00.000Z", "deleted_at": null,
  "id": 2, "email": "Evelyn.Patnode@gmail.com", "password": "d67865644a344023590e4f1cace1f4", "details": {"sex": "M", "created_at": "2010-11-12T21:27:00.000Z", "deleted_at": null},
  "id": 3, "email": "Derek.Crenshaw@gmail.com", "password": "5ab7bc1590631065b41059097f0efe", "details": {"sex": "F", "created_at": "2009-03-08T03:06:00.000Z", "deleted_at": null},
  "id": 4, "email": "Sharil.Jullian@yahoo.com", "password": "9d38d622b73d4363e9814e5f", "details": {"sex": "M", "created_at": "2010-11-20T10:58:00.000Z", "deleted_at": null},
  "id": 5, "email": "Zita.Breeding@gmail.com", "password": "7a1c8d1d80d75da38ef8d03f38847d", "details": null, "created_at": "2009-08-11T22:33:00.000Z", "deleted_at": null,
  "id": 6, "email": "Samatha.Hedgpeth@yahoo.com", "password": "e129316bf01b0440247414b75726956", "details": {"sex": "F", "created_at": "2010-07-18T10:40:00.000Z", "deleted_at": null},
  "id": 7, "email": "Quinton.Gilpatrick@yahoo.com", "password": "7c63f3c25ee52041c2b9aacc321a96b6", "details": {"sex": "M", "created_at": "2010-09-02T21:56:00.000Z", "deleted_at": null},
  "id": 8, "email": "Wivian.Westmoreland@yahoo.com", "password": "100945c1684d6926dafca9d97aed9", "details": {"sex": "F", "state": "South Carolina", "created_at": "2009-10-01T11:34:00.000Z", "deleted_at": null},
  "id": 9, "email": "Danny.Craves@gmail.com", "password": "511e3229996147ae68f83bab75b9733e", "details": {"sex": "M", "created_at": "2009-04-22T07:30:00.000Z", "deleted_at": null},
  "id": 10, "email": "Edmund.Rolies@yahoo.com", "password": "5aacc2309a2b01e9177564126a6f8393", "details": {"sex": "F", "state": "New York", "created_at": "2009-07-07T21:01:00.000Z", "deleted_at": null},
  "id": 11, "email": "Shanell.Lichtenstein@aol.com", "password": "98ac14b2c6b0ef8a53b5c54ee5f2b", "details": {"sex": "M", "created_at": "2009-05-22T00:18:00.000Z", "deleted_at": null},
  "id": 12, "email": "Romaine.Birdsell@aol.com", "password": "4571853f5ee73e305ac152c765ad2915", "details": {"sex": "F", "created_at": "2009-01-14T05:07:00.000Z", "deleted_at": null},
  "id": 13, "email": "Zita.Lumany@yahoo.com", "password": "7467fa8332bc45a15ad2c7003c963ea2", "details": null, "created_at": "2009-02-04T14:49:00.000Z", "deleted_at": null},
  "id": 14, "email": "Claud.Cousineau@gmail.com", "password": "82b0cc0c63c01a9bbae5dc7b9fabcc", "details": {"sex": "F", "created_at": "2009-08-17T18:48:00.000Z", "deleted_at": null},
  "id": 15, "email": "Kali.Damore@yahoo.com", "password": "66327b7500c1b4115910260481f8582", "details": {"sex": "F", "created_at": "2010-07-07T10:28:00.000Z", "deleted_at": null},
  "id": 16, "email": "Graciela.Kubalicki@yahoo.com", "password": "85dbdc9f9f08c157d7d1055009e98ff", "details": {"sex": "F", "created_at": "2010-08-19T05:42:00.000Z", "deleted_at": null},
  "id": 17, "email": "Theresa.Edwin@yahoo.com", "password": "87b2ae03da521142d376766a6376a", "details": {"sex": "M", "created_at": "2010-08-11T08:21:00.000Z", "deleted_at": null},
  "id": 18, "email": "Ozella.Yoshimura@gmail.com", "password": "df68a6070ac1f8ce7a16baa96922948", "details": {"sex": "M", "created_at": "2010-07-23T16:03:00.000Z", "deleted_at": null},
  "id": 19, "email": "Wynona.Greening@aol.com", "password": "176c818b6c6324925f2f6c274667e3e8f", "details": {"sex": "M", "created_at": "2009-05-24T14:25:00.000Z", "deleted_at": null},
  "id": 20, "email": "Kimi.Mcqueeny@gmail.com", "password": "588169a5e191c0e99b087a392b03ada", "details": null, "created_at": "2010-06-22T15:16:00.000Z", "deleted_at": null},
  "id": 21, "email": "Cherry.Karnowski@gmail.com", "password": "15981f660fede4e69f9e21cc146aa6", "details": null, "created_at": "2009-01-26T09:56:00.000Z", "deleted_at": null},
  "id": 22, "email": "Isabel.Breeding@gmail.com", "password": "a32fbb2e28f4ce747d0ee30aaf9ae5", "details": null, "created_at": "2010-07-11T13:28:00.000Z", "deleted_at": null},
  "id": 23, "email": "Ivana.Kurthy@yahoo.com", "password": "ca72fafe92a1ef1520b6b53e2532571", "details": null, "created_at": "2010-06-25T08:36:00.000Z", "deleted_at": null},
  "id": 24, "email": "Bumberto.Tonony@yahoo.com", "password": "642a91737480d3bbbf2168963ee3", "details": null, "created_at": "2009-09-23T13:09:00.000Z", "deleted_at": null},
  "id": 25, "email": "Ivana.Sosnowski@aol.com", "password": "f12980358430ee7ae192b01aa6ac05d", "details": {"sex": "M", "created_at": "2009-01-16T11:55:00.000Z", "deleted_at": null},
  "id": 26, "email": "Courtney.Struvey@gmail.com", "password": "d80da950209cfdb96c76648d45f5b87", "details": {"sex": "M", "state": "Virginia", "created_at": "2009-07-19T06:08:00.000Z", "deleted_at": null},
  "id": 27, "email": "Williams.Upson@gmail.com", "password": "de9a71ad16ed44395d3b64f686430a", "details": {"sex": "F", "created_at": "2010-08-10T05:48:00.000Z", "deleted_at": null},
  "id": 28, "email": "Jeremiah.Buayocore@yahoo.com", "password": "1994c6611461fc9d11683b50e540d701", "details": null, "created_at": "2009-03-19T07:49:00.000Z", "deleted_at": null},
  "id": 29, "email": "Ozella.Rolies@gmail.com", "password": "8bee01c9b64ed4ca3e683c3502e1d85", "details": null, "created_at": "2009-10-09T09:44:00.000Z", "deleted_at": null},
  "id": 30, "email": "Salvatore.Arendse@aol.com", "password": "8664e4bf1574238287230f6d014664", "details": {"sex": "F", "state": "Virginia", "created_at": "2009-09-05T01:55:00.000Z", "deleted_at": null},
  "id": 31, "email": "Layne.Sarver@aol.com", "password": "296ca911a6fc78b43b5f5927c16cfd", "details": {"sex": "M", "created_at": "2010-09-26T08:00:00.000Z", "deleted_at": null},
  "id": 32, "email": "Takako.Gilpatrick@aol.com", "password": "3abe3e825f6e749dca1b8193d5f15215", "details": {"sex": "M", "created_at": "2009-02-22T15:46:00.000Z", "deleted_at": null},
  "id": 33, "email": "Russ.Mcclain@yahoo.com", "password": "d1f1dc7c23e4a93f5ed8e0352aa5f", "details": {"sex": "F", "state": "Florida", "created_at": "2010-01-12T17:27:00.000Z", "deleted_at": null},
  "id": 34, "email": "Claud.Westmoreland@aol.com", "password": "631f77eeef3e13c8aad646fd73c03a", "details": null, "created_at": "2010-06-11T17:21:00.000Z", "deleted_at": null},
  "id": 35, "email": "Derek.Knitte@gmail.com", "password": "ce3ce9650891124de7f449e8433ff71", "details": {"sex": "F", "created_at": "2010-08-16T21:09:00.000Z", "deleted_at": null},
  "id": 36, "email": "Eleanor.Patnode@yahoo.com", "password": "c20912ab068921f869ee26724bd0e081", "details": {"sex": "F", "state": "Florida", "created_at": "2010-06-06T01:27:00.000Z", "deleted_at": null},
  "id": 37, "email": "Carmel.Bulfer@aol.com", "password": "1526fe5da06c6fcd010c90f8a744b3", "details": {"sex": "F", "state": "Florida", "created_at": "2009-06-06T20:13:00.000Z", "deleted_at": null},
  "id": 38, "email": "Maurio.Pung@yahoo.com", "password": "4e625168e5a9b5d48c303d20ccc95b5", "details": {"sex": "F", "state": "Illinois", "created_at": "2009-08-20T02:34:00.000Z", "deleted_at": null},
  "id": 39, "email": "Sherilyn.Hamill@gmail.com", "password": "2f313c4006182796faf17d83ef73312b", "details": {"sex": "M", "created_at": "2010-04-01T23:39:00.000Z", "deleted_at": null},
  "id": 40, "email": "Glen.Lamphear@yahoo.com", "password": "66563168a637a3c5f43bdcf3f2e931a", "details": null, "created_at": "2010-08-06T15:14:00.000Z", "deleted_at": null},
  "id": 41, "email": "Stacia.Schrack@aol.com", "password": "8a918b3f9c9d9aefb5d5df2b2a07", "details": {"sex": "M", "created_at": "2010-06-14T19:28:00.000Z", "deleted_at": null},
  "id": 42, "email": "Tonette.Albagack@aol.com", "password": "9e74217f6d41b88a4c7734267a8a0", "details": null, "created_at": "2009-12-28T10:21:00.000Z", "deleted_at": null},
  "id": 43, "email": "Eve.Kump@yahoo.com", "password": "30b9c55bf6e5454178e1ef08228498a", "details": {"sex": "M", "created_at": "2009-08-20T09:45:00.000Z", "deleted_at": null},
  "id": 44, "email": "Shanell.Maxon@gmail.com", "password": "e47a8b0056427f189f146889457f5c2", "details": {"sex": "M", "created_at": "2009-11-21T06:28:00.000Z", "deleted_at": null},
  "id": 45, "email": "Gudrun.Arendse@gmail.com", "password": "735dba8760996aaf1b08b443c65f49", "details": null, "created_at": "2010-06-30T12:30:00.000Z", "deleted_at": null},
  "id": 46, "email": "Annelise.Feuer@yahoo.com", "password": "870f6d39e2a9b8d30b1309936744cb", "details": {"sex": "F", "created_at": "2009-08-21T17:06:00.000Z", "deleted_at": null},
  "id": 47, "email": "Harrison.Puett@yahoo.com", "password": "ff9e46a0aca39a2c3bbd804307826a", "details": {"sex": "M", "created_at": "2009-07-21T15:20:00.000Z", "deleted_at": null},
  "id": 48, "email": "Granville.Hedgpeth@gmail.com", "password": "87f0bd39e2a9b8d30b1309936744cb", "details": null, "created_at": "2009-08-03T14:54:00.000Z", "deleted_at": null},
  "id": 49, "email": "Samatha.Pellegrini@yahoo.com", "password": "4e2a5f4b46263dc07519bc49841822", "details": null, "created_at": "2009-03-25T20:17:00.000Z", "deleted_at": null},
  "id": 50, "email": "Wan.Dillks@gmail.com", "password": "0650f5923e2abce4172id3d9ab37cc54", "details": {"sex": "M", "created_at": "2009-10-08T22:43:00.000Z", "deleted_at": null}}

```

Question 3

Parameterised Query

Here I have ran the same query (as above), however I have added \$1 where title is included to ensure the string isn't parsed prior to entering the database. The result is nothing is returned this time.

```
//using a parametrised query
app.get('/products/sql-protect', (req, res) => {
  let query_string = req.query.name;

  db.query("SELECT * FROM products WHERE title=$1", query_string ).then(resp => {
    res.send(resp);
  })
})
```

11

Stored Procedure

It is useful to use a stored procedure when sending a query via the URL, to ensure it is not possible to pass a SQL statement or other malicious code. These procedures are stored in the database script and are precompiled before running, which means they're different from the previous run-time queries, due to the point in which they are executed.

```
//using a stored procedure
app.get('/users/sql-protect-it', (req, res) =>
{
    let _email = req.query.email;
    let stmt = "CREATE OR REPLACE FUNCTION authenticate(_email TEXT)" +
        " RETURNS details AS $$" +
        " SELECT * FROM users WHERE email=_email";

    db.query(stmt, _email ).then(resp => {
        res.send(resp);
    })
})
})
```

Question 4 - Setting up models in Sequelize

Users Model

```
'use strict';
module.exports = function(sequelize, DataTypes) {
    const users = sequelize.define('users', {
        email: DataTypes.STRING,
        password: DataTypes.STRING,
        details: DataTypes.STRING,
        deleted_at: {
            type: 'TIMESTAMP',
            defaultValue: sequelize.literal('CURRENT_TIMESTAMP'),
            allowNull: false
        }
    }, {
        classMethods: {
            associate: function (models) {
                // associations can be defined here
            }
        }
    });
    return users;
};
```


Products Model

```
'use strict';
module.exports = function (sequelize, DataTypes) {
  const products = sequelize.define('products', {
    title: DataTypes.STRING,
    price: DataTypes.DOUBLE,
    deleted_at: {
      type: 'TIMESTAMP',
      defaultValue: sequelize.literal('CURRENT_TIMESTAMP'),
      allowNull: false
    },
    tags: {
      type: DataTypes.ARRAY(DataTypes.STRING)
    }
  }, {
    classMethods: {
      associate: function (models) {
        // associations can be defined here
      }
    }
  });
  return products;
};
```

Purchase Items Model

```
'use strict';
module.exports = function(sequelize, DataTypes) {
  const purchase_items = sequelize.define('purchase_items', {
    purchase_id: DataTypes.INTEGER,
    product_id: DataTypes.INTEGER,
    price: DataTypes.DOUBLE,
    quantity: DataTypes.INTEGER,
    state: DataTypes.STRING
  }, {
    classMethods: {
      associate: function (models) {
        // associations can be defined here
        purchase_items.belongsTo(models.purchases);
        purchase_items.belongsTo(models.products);
      }
    }
  });
  return purchase_items;
};
```

Purchases Model

```
'use strict';
module.exports = function(sequelize, DataTypes) {
  const purchases = sequelize.define('purchases', {
    name: DataTypes.STRING,
    address: DataTypes.STRING,
    state: DataTypes.STRING,
    zipcode: DataTypes.INTEGER
  }, {
    classMethods: {
      associate: function(models) {
        // associations can be defined here
        purchases.belongsTo(models.users);
      }
    }
  });
  return purchases;
};
```

Question 5 - example updating model with test data

```
//post new purchase item
router.post('/purchases', function (req, res) {
  models.purchases.create({
    title: req.body.title,
    name: req.body.name,
    address: req.body.address,
    state: req.body.state,
    zipcode: req.body.zipcode,
  }).then(function (purchases) {
    res.json(purchases);
  });
});
```

Code snippet for updating purchases with test data:

```
Catherines-MacBook-Pro-2:Lab 1 Part 2 CatherineKane$ curl --data "title=Test&name=Tests&address=123TestStreet&state=FL&zipcode=123456" http://127.0.0.1:3000
{"id":1001,"name":"Tests","address":"123TestStreet","state":"FL","zipcode":123456,"updatedAt":"2019-02-17T22:14:10.688Z","createdAt":"2019-02-17T22:14:10.688Z"}
```

Output for new test data in purchases:

```
{"id":1001,"name":"Tests","address":"123TestStreet","state":"FL","zipcode":123456,"createdAt":"2019-02-17T22:14:10.688Z","updatedAt":"2019-02-17T22:14:10.688Z"}
```


Question 6

Code snippet to list all products:

```
// list all products
router.get('/products', function (req, res) {
  models.products.findAll({}).then(function (products) {
    res.json(products);
  });
});
```

Output list of all products:

← → ↺ ⓘ localhost:5000/products

```
[{"id":2,"title":"Python Book","price":29.99,"deleted_at":null,"tags":["Book","Programming","Python"],"createdAt":"2011-01-01T20:00:00.000Z","updatedAt":"2011-01-01T20:00:00.000Z"}, {"id":3,"title":"Ruby Book","price":27.99,"deleted_at":null,"tags":["Book","Programming","Ruby"],"createdAt":"2011-01-01T20:00:00.000Z","updatedAt":"2011-01-01T20:00:00.000Z"}, {"id":4,"title":"Baby Book","price":7.99,"deleted_at":null,"tags":["Book","Children","Baby"],"createdAt":"2011-01-01T20:00:00.000Z","updatedAt":"2011-01-01T20:00:00.000Z"}, {"id":5,"title":"Children Book","price":5.99,"deleted_at":null,"tags":["Book","Children"],"createdAt":"2011-01-01T20:00:00.000Z","updatedAt":"2011-01-01T20:00:00.000Z"}, {"id":6,"title":"Computer","price":499.99,"deleted_at":null,"tags":["Technology"],"createdAt":"2011-01-01T20:00:00.000Z","updatedAt":"2011-01-01T20:00:00.000Z"}, {"id":7,"title":"Computer","price":899.99,"deleted_at":null,"tags":["Technology"],"createdAt":"2011-01-01T20:00:00.000Z","updatedAt":"2011-01-01T20:00:00.000Z"}, {"id":8,"title":"Player","price":108,"deleted_at":null,"tags":["Technology","Music"],"createdAt":"2011-01-01T20:00:00.000Z","updatedAt":"2011-01-01T20:00:00.000Z"}, {"id":9,"title":"TV","price":499,"deleted_at":null,"tags":["Technology","TV"],"createdAt":"2011-01-01T20:00:00.000Z","updatedAt":"2011-01-01T20:00:00.000Z"}, {"id":10,"title":"TV","price":529,"deleted_at":null,"tags":["Technology","TV"],"createdAt":"2011-01-01T20:00:00.000Z","updatedAt":"2011-01-01T20:00:00.000Z"}, {"id":11,"title":"CD","price":9.99,"deleted_at":null,"tags":["Music"],"createdAt":"2011-01-01T20:00:00.000Z","updatedAt":"2011-01-01T20:00:00.000Z"}, {"id":12,"title":"Country CD","price":9.99,"deleted_at":null,"tags":["Music"],"createdAt":"2011-01-01T20:00:00.000Z","updatedAt":"2011-01-01T20:00:00.000Z"}, {"id":13,"title":"Country CD","price":9.99,"deleted_at":null,"tags":["Music"],"createdAt":"2011-01-01T20:00:00.000Z","updatedAt":"2011-01-01T20:00:00.000Z"}, {"id":14,"title":"Pop CD","price":9.99,"deleted_at":null,"tags":["Music"],"createdAt":"2011-01-01T20:00:00.000Z","updatedAt":"2011-01-01T20:00:00.000Z"}, {"id":15,"title":"Electronic CD","price":9.99,"deleted_at":null,"tags":["Music"],"createdAt":"2011-01-01T20:00:00.000Z","updatedAt":"2011-01-01T20:00:00.000Z"}, {"id":16,"title":"Movie","price":14.99,"deleted_at":null,"tags":["Movie","Comedy"],"createdAt":"2011-01-01T20:00:00.000Z","updatedAt":"2011-01-01T20:00:00.000Z"}, {"id":17,"title":"Documentary","price":14.99,"deleted_at":null,"tags":["Movie"],"createdAt":"2011-01-01T20:00:00.000Z","updatedAt":"2011-01-01T20:00:00.000Z"}, {"id":18,"title":"Romantic","price":14.99,"deleted_at":null,"tags":["Movie"],"createdAt":"2011-01-01T20:00:00.000Z","updatedAt":"2011-01-01T20:00:00.000Z"}, {"id":19,"title":"Drama","price":14.99,"deleted_at":null,"tags":["Movie"],"createdAt":"2011-01-01T20:00:00.000Z","updatedAt":"2011-01-01T20:00:00.000Z"}, {"id":20,"title":"Action","price":14.99,"deleted_at":null,"tags":["Movie"],"createdAt":"2011-01-01T20:00:00.000Z","updatedAt":"2011-01-01T20:00:00.000Z"}, {"id":1,"title":"Dictionary","price":9.99,"deleted_at":null,"tags":["Book"],"createdAt":"2011-01-01T20:00:00.000Z","updatedAt":"2011-01-01T20:00:00.000Z"}, {"id":27,"title":"Wallet","price":13.45,"deleted_at":"2019-02-17T20:30:45.976Z","tags":null,"createdAt":"2019-02-17T20:30:45.976Z","updatedAt":"2019-02-17T20:30:45.976Z"}, {"id":28,"title":"TestProduct","price":20.19,"deleted_at":"2019-02-17T20:40:53.443Z","tags":null,"createdAt":"2019-02-17T20:40:53.443Z","updatedAt":"2019-02-17T20:40:53.443Z"}]
```

Code snippet to list single product:

```
// list a single product
router.get('/products/:id', function (req, res) {
  models.products.find({
    where: {
      id: req.params.id
    }
  }).then(function (products) {
    res.json(products);
  });
});
```

Output of single product:

← → ↻ ⓘ localhost:5000/products/20

```
{"id":20,"title":"Action","price":14.99,"deleted_at":null,"tags":["Movie"],"createdAt":"2011
```

Code snippet to create a new product:

```
Catherines-MacBook-Pro-2:Lab 1 Part 2 CatherineKane$ curl --data "title=TestP
```

Output of new product create:

```
{"id":28,"title":"TestProduct","price":20.19,"deleted_at":"2019-02-17
```

Code snippet to update a product:

```
Catherines-MacBook-Pro-2:Lab 1 Part 2 CatherineKane$ curl -X PUT --data "titl  
{"id":28,"title":"NewTestProduct","deleted_at":"2019-02-17T20:40:53.443Z","ta
```

Output of update to product:

← → ↻ ⓘ localhost:5000/products/28

```
{"id":28,"title":"NewTestProduct","price":20.19,
```

Code snippet to delete a product:

```
Catherines-MacBook-Pro-2:Lab 1 Part 2 CatherineKane$ curl -X DELETE http://127.0.0.1:5000/products/28
```

Output to check product is deleted:

← → ↻ ⓘ localhost:5000/products/28

```
null
```