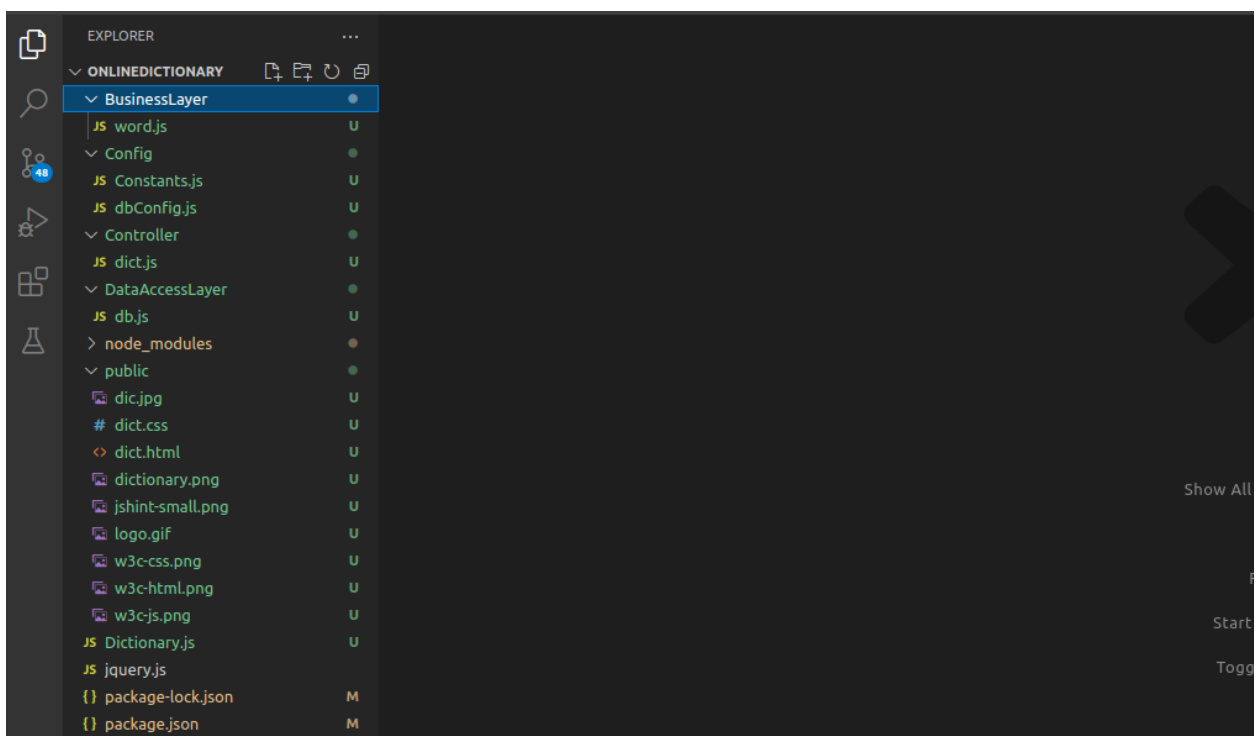
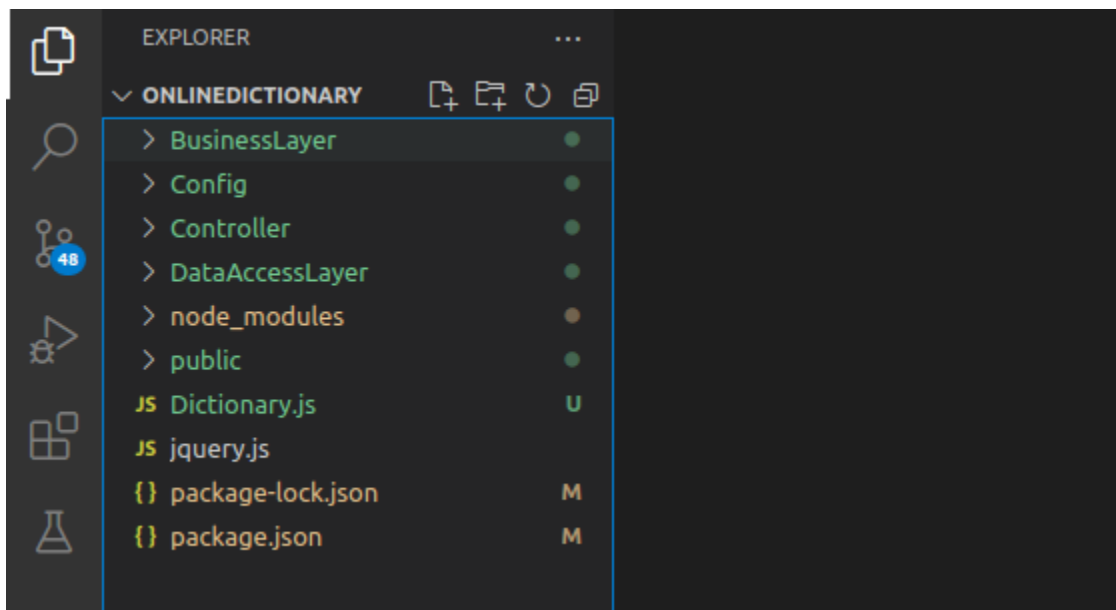


## Project Structure

(<https://github.com/tsitotaw/tsitotaw.github.io/tree/master/nodejs/onlineDictionary>)



## Code Snippets

### Dictionary.js

```
1  const express = require('express');
2
3  const Entry = require('./BusinessLayer/word');
4  const path = require('path');
5  const constants = require('./Config/Constants');
6
7  const app = express();
8
9  app.use(express.static('public'));
10 app.use(express.static(__dirname));
11
12 app.get('/', (req, res) => {
13   res.status(200).sendFile(path.join(__dirname + '/public/dict.html'));
14 });
15
16 app.get('/entry/:word?', (req, res, next) => {
17   let word = req.params.word;
18   if (word) {
19     Entry.findByWord(word, (err, data) => {
20       if (err) {
21         res.status(404).send(err);
22       } else {
23         res.send(data);
24       }
25     });
26   } else {
27     res.send([]);
28     next();
29   }
30 });
31
32 app.listen(constants.PORT, () => {
33   console.log(`Server is listening on port ${constants.PORT}`);
34 });
35
36
37
```

```

1  <!DOCTYPE html>
2  <html lang="en-US">
3      <head>
4          <meta charset="UTF-8">
5
6          <link rel="stylesheet" type="text/css" href="dict.css">
7          <title>Online English Dictionary</title>
8      </head>
9      <body>
10         <div class="header">
11             
12         </div>
13         <div id="wordinput">
14             <label for="txtword">Term:</label>
15             <input type="text" name="txtword" id="txtword" placeholder="word" />
16             <button id="btnLookup">Lookup</button>
17         </div>
18
19         <ol id="definitions"></ol>
20
21         <script src="../../jquery.js" type="text/javascript"></script>
22         <script src="../../Controller/dict.js" type="text/javascript"></script>
23     </body>
24 </html>
25

```

```

1  const mysql = require('mysql');
2  const dbConfig = require("../Config/dbConfig");
3
4  const connection = mysql.createConnection({
5      host: dbConfig.HOST,
6      user: dbConfig.USER,
7      password: dbConfig.PASSWORD,
8      database: dbConfig.DB
9  });
10
11  connection.connect(error => {
12      if (error) throw error;
13      console.log("Successfully connected to the database.");
14  });
15
16  module.exports = connection;

```

```

1  const sql = require("../DataAccessLayer/db");
2
3  const Entry = function(entry){
4      this.word = entry.word;
5      this.wordtype = entry.wordtype;
6      this.definition = entry.definition;
7  };
8
9  Entry.findByWord = (word, result) => {
10      sql.query(`SELECT * FROM entries WHERE word = '${word}'`, (err, res) => {
11          if(err){
12              console.error(err);
13              result(err, null);
14              return;
15          }
16          result(null, res);
17      });
18  };
19
20  module.exports = Entry;

```

Controller > JS dict.js > ready() callback > clickHandler > success

```
1  "use strict";
2  $(document).ready(function(){
3
4      let clickHandler = function(){
5          let word = $("#txtword").val();
6          $.ajax({
7              url: 'http://localhost:8081/entry/' + word,
8              type: 'GET',
9              success: (words) => {
10                 $('#definitions').html("");
11
12                 if(words.length < 1){
13                     $('#definitions').append("<p> no definition found for the specified term - or no term specified </p>");
14                     return false;
15                 }
16
17                 for(let i=0; i< words.length; i++) {
18                     $('#definitions').append(`
19                         <li>
20                             wordtype: ${words[i]["wordtype"]} <br>
21                             definition: ${words[i]["definition"]}
22                         </li>
23                         <hr>`
24                     );
25                 }
26             },
27             error: function(error){
28                 $('#definitions').append(error);
29             },
30         });
31     });
32 }
33
34 $("#btnLookup").on({
35     'click':clickHandler,
36     'keypress': (e) => {
37         if(e.which == 13) {
38             clickHandler()
39         }
40     }
41 });
42
43
```

Config > JS Constants.js > ...

```
1  const port = 8081;
2  const url = `http://localhost:${port}/entry/`;
3  exports.PORT = port;
4  exports.URL = url;
```

JS dbConfig.js U

Config > JS dbConfig.js > <unknown> > PASSWORD

```
1  module.exports = {
2      HOST: "localhost",
3      USER: "root",
4      PASSWORD: "*****",
5      DB: "entries"
6  };
```



Term:

[Lookup](#)



Term:

[Lookup](#)

- 
1. wordtype: n.  
definition : Example; pattern.

---

  2. wordtype: n.  
definition : A part of anything presented for inspection, or shown as evidence of the quality of the whole; a specimen; as, goods are often purchased by samples.

---

  3. wordtype: v. t.  
definition : To make or show something similar to; to match.

---

  4. wordtype: v. t.  
definition : To take or to test a sample or samples of; as, to sample sugar, teas, wools, cloths.

---



Term:

[Lookup](#)

- 
1. wordtype: a.  
definition : Endowed with the faculty of understanding or reason; as, man is an intelligent being.

---

  2. wordtype: a.  
definition : Possessed of intelligence, education, or judgment; knowing; sensible; skilled; marked by intelligence; as, an intelligent young man; an intelligent architect; an intelligent answer.

---

  3. wordtype: a.  
definition : Gognizant; aware; communicate.

---



Term:

[Lookup](#)

---

no definition found for the specified term - or no term specified