

```
1 [HOL] Full stack Project with Vue.js
2 -----
3 Task1. Back-end Project 준비하기
4 1. In Visual Studio Code > Terminal
5 2. $ md rest_contacts
6 3. $ cd rest_contacts
7
8 4. $ npm init
9     1)package.json
10    {
11      "name": "rest_contacts",
12      "version": "1.0.0",
13      "description": "",
14      "main": "index.js",
15      "scripts": {
16        "test": "echo \"Error: no test specified\" && exit 1"
17      },
18      "author": "",
19      "license": "ISC"
20    }
21
22 5. $ npm install express mongoose body-parser cors --save
23
24 -----
25
26 Task2. Back-end Server 준비하기
27 1. index.js
28   const express = require('express')
29   const bodyparser = require('body-parser')
30   const cors = require('cors')
31   const app = express() //express app 생성
32
33   app.use(bodyparser.json()) //json 형식 parsing하기
34   app.use(cors()) //cors 적용
35
36   const dbconfig = require('./db.js')
37   const mongoose = require('mongoose')
38   //Database 연결 및 상태 경로
39   mongoose.connect(dbconfig.url, { useNewUrlParser:true})
40   .then(() => {
41     console.log('MongoDB Connect Success.')
42   }).catch(err => {
43     console.log('MongoDB Connect Failure', err)
44   })
45
46   app.get('/', (req, res) => {
47     console.log(req)
48     res.json({ 'message': 'Welcome! Hello, World!!!' })
49   })
50
51   let port = process.env.PORT || 8000 //Server Port 설정
52
53   //Client으로부터 요청 받기
54   app.listen(port, () => {
55     console.log('Port : ' + port + ' is ready...')
56   })
57
58
```

```
59 2. db.js
60
61   module.exports = {
62     url : 'mongodb://localhost:27017/contactlist'
63   }
64
65
66 3. MongoDB Server Start
67   rest_contacts> $ md .\data\db
68   rest_contacts> $ mongod --dbpath data/db
69
70
71 4. VSCode Terminal에서
72   $ node index.js
73   Port : 8000 is ready...
74   MongoDB Connect Success.
75
76
77 5. Browser에서
78   http://localhost:8000
79   {"message":"Welcome! Hello, World!!!"}
80
81
82 -----
83 Task3. Back-end Coding
84 1. contact.model.js
85   const mongoose = require('mongoose')
86
87   var ContractSchema = mongoose.Schema({ //Schema 설정
88     contactId: {type : Number, require:true},
89     name : { type:String, require : true},
90     email : { type : String, require : true},
91     gender : String,
92     phone : String,
93     register_date : { type:Date, default : Date.now}
94   })
95   module.exports = mongoose.model('Contact', ContractSchema);
96
97
98 2. router.js
99   module.exports = (app) => {
100     const contacts = require('./contact.controller.js'); //contact.controller.js 로딩
101     app.get('/contacts', contacts.findAll); //모든 연락처 검색
102     app.get('/contacts/:contactId', contacts.findOne); //특정 연락처 검색
103     app.post('/contacts', contacts.create); //새로운 연락처 추가
104     app.put('/contacts/:contactId', contacts.update); //특정 연락처 업데이트
105     app.delete('/contacts/:contactId', contacts.delete); //특정 연락처 삭제
106   }
107
108
109 3. index.js 수정
110   require('./router.js')(app); <-- 여기에 추가
111   let port = process.env.PORT || 8000 //Server Port 설정
112
113
114 4. contact.controller.js
115   const Contact = require('./contact.model.js')
116
```

```
117 exports.create = (req, res) => {
118   //새로운 연락처 만들기
119   const contact = new Contact({
120     contactId : req.body.contactId,
121     name : req.body.name,
122     email : req.body.email,
123     gender : req.body.gender,
124     phone: req.body.phone
125   });
126
127   //Database에 새로운 연락처 저장하기
128   contact.save()
129   .then(data => {
130     res.send(data);
131   })
132   .catch(err => {
133     res.status(500).send({
134       message : err.message
135     })
136   });
137 }
138
139 exports.findAll = (req, res) => {
140   Contact.find()
141   .then(contacts => {
142     res.send(contacts);
143   })
144   .catch(err => {
145     res.status(500).send({
146       message : err.message
147     })
148   });
149 }
150
151 exports.findOne = (req,res) => {
152   Contact.findOne({
153     contactId : req.params.contactId
154   })
155   .then(contact => {
156     if(!contact){
157       return res.status(404).send({
158         message:req.params.contactId + '에 해당하는 연락처가 없습니다.'
159       });
160     }
161     res.send(contact);
162   })
163   .catch(err => {
164     return res.status(500).send({
165       message : req.params.contactId + '로 검색 중 에러 발생'
166     });
167   })
168 }
169
170 exports.update = (req, res) => {
171   Contact.findOneAndUpdate (
172     {contactId:req.params.contactId},
173     {contactId:req.body.contactId, name:req.body.name, email:req.body.email,
174     gender:req.body.gender, phone:req.body.phone}, {new:true}
```

```

175     )
176     .then(contact => {
177         if(!contact){
178             return res.status(404).send({
179                 message : req.params.contactId + '에 해당하는 연락처(contact) 발견되지 않았습니다.'
180             })
181         }
182         res.send(contact);
183     })
184     .catch(err => {
185         return res.status(500).send({
186             message : err.message
187         });
188     })
189 }
190
191 exports.delete = (req, res) => {
192     Contact.findOneAndDelete({contactId:req.params.contactId})
193     .then(contact => {
194         if(!contact){
195             return res.status(404).send({
196                 message:
197                     req.params.contactId +
198                     "에 해당하는 연락처(contact)가 없습니다."
199             });
200         }
201         res.send({
202             message : '정상적으로 ' + req.params.contactId + ' 연락처가 삭제되었습니다.'
203         })
204     })
205     .catch(err => {
206         return res.status(500).send({
207             message : err.message
208         });
209     });
210 }

```

213 5. VSCode Terminal에서 Ctrl + C로 정지 후 다시 시작한다.

215 6. http://localhost:8000/contacts

216 1)데이터가 없기 때문에 [] 만 나타난다.

219 7. POSTMAN으로 Test

220 1)혹시 Chrome에 Postman이 없을 경우 아래 링크를 클릭해서 설치할 것

222 <https://chrome.google.com/webstore/detail/postman/fhbjgbiflinjbdgggehcdcbncdddomop?hl=ko>

225 8. POST : 연락처 추가

226 POST : http://localhost:8000/contacts

```

227 {
228     "contactId" : 1,
229     "name" : "한지민",
230     "email" : "aaa@aaa.com",
231     "gender" : "여성",

```

```

232     "phone" : "010-1234-5678"
233   }
234   -----
235   {
236     "_id": "5dfd07923276581cac4f81eb",
237     "contactId": 1,
238     "name": "한지민",
239     "email": "aaa@aaa.com",
240     "gender": "여성",
241     "phone": "010-1234-5678",
242     "register_date": "2020-06-02T13:14:18.258Z",
243     "__v": 0
244   }
245   -----
246   -http://localhost:8000/contacts와도 동일한 결과가 나온다.
247
248

```

249 9. 2명 더 연락처를 추가한다.

```

250   {
251     "contactId" : 2,
252     "name" : "박지민",
253     "email" : "bbb@bbb.com",
254     "gender" : "남성",
255     "phone" : "010-7777-8888"
256   }
257   {
258     "contactId" : 3,
259     "name" : "김지민",
260     "email" : "ccc@ccc.com",
261     "gender" : "여성",
262     "phone" : "010-5678-9876"
263   }
264
265

```

266 10. GET : 데이터 검색

267 1)전체 검색

```

268   -http://localhost:8000/contacts
269   [{ "_id": "5ed650aa7c6cdf5e587c0cc1", "contactId": 1, "name": "한지
민", "email": "aaa@aaa.com", "gender": "여
성", "phone": "010-1234-5678", "register_date": "2020-06-02T13:14:18.258Z", "__v": 0 }, { "_
id": "5ed650ff7c6cdf5e587c0cc2", "contactId": 2, "name": "박지
민", "email": "bbb@bbb.com", "gender": "남
성", "phone": "010-7777-8888", "register_date": "2020-06-02T13:15:43.801Z", "__v": 0 }, { "_
id": "5ed6510d7c6cdf5e587c0cc3", "contactId": 3, "name": "김지
민", "email": "ccc@ccc.com", "gender": "여
성", "phone": "010-5678-9876", "register_date": "2020-06-02T13:15:57.355Z", "__v": 0 } ]

```

270
271 2)특정 데이터 검색

```

272   -http://localhost:8000/contacts/2
273   { "_id": "5ed650ff7c6cdf5e587c0cc2", "contactId": 2, "name": "박지
민", "email": "bbb@bbb.com", "gender": "남
성", "phone": "010-7777-8888", "register_date": "2020-06-02T13:15:43.801Z", "__v": 0 }

```

274
275
276 11. PUT : 데이터 업데이트

```

277   -PUT : http://localhost:8000/contacts/3
278   {
279     "contactId" : 3,

```

```

280     "name" : "김지민",
281     "email" : "ccc@ccc.com",
282     "gender" : "여성",
283     "phone" : "010-9999-9999" <--전화번호 수정
284 }
285
286 -3번의 전화번호가 변경된 것이 Browser에서 확인
287 -http://localhost:8000/contacts/3
288 { "_id": "5ed6510d7c6cdf5e587c0cc3", "contactId": 3, "name": "김지
    민", "email": "ccc@ccc.com", "gender": "여
    성", "phone": "010-9999-9999", "register_date": "2020-06-02T13:15:57.355Z", "__v": 0 }
289
290
291 12. DELETE : 데이터 삭제
292 -DELETE : http://localhost:8000/contacts/2
293
294 {
295     "message": "정상적으로 2 연락처가 삭제되었습니다."
296 }
297
298 -----
299 Browser 에서 http://localhost:8000/contacts/2로 검색하면
300
301 {"message": "2에 해당하는 연락처가 없습니다."}
302
303
304 -----
305
306 Task4. Front-end 준비하기
307 1. contactscient Project 생성하기
308     1) 새로운 Command 창에서
309
310     $ vue create contactscient --default
311     ...
312     ...
313     $ cd contactscient
314     $ npm run serve
315     -Web Browser에서 확인
316
317 2) 여기서 새 Visual Studio Code 열기
318
319     contactscient> $ code .
320
321
322 2. Vuetify 설치
323     1) Google Material Design에 기반을 둔 Vue.js에 특화된 UI 컴포넌트 라이브러리.
324     2) Command 창에서
325         $ vue add vuetify
326     3) Default(recommended) 로 설치한다.
327     4) 설치 후 npm run serve로 확인
328
329
330 3. App.vue
331
332     ...
333     ...
334     <v-content>
335     <h2>연락처 리스트</h2>

```

```

336     <data-table></data-table>
337   </v-content>
338 </template>
339
340 <script>
341 import DataTable from './components/DataTable';
342
343 export default {
344   name: 'App',
345
346   components: {
347     DataTable
348   },
349
350   data: () => ({
351     //
352   }),
353 };
354 </script>
355
356 <style scoped>
357 h2 {
358   margin-top:20px;
359   margin-left:10px;
360   color:blue;
361 }
362 </style>

```

4. axios 설치

```
$ npm install --save axios
```

```
-----
```

Task5. Front-end Coding

1. DataTable.vue

```

372 <template>
373   <div id="app">
374     <v-data-table :headers="headers" :items="items" class="elevation-1">
375       <template v-slot:top>
376         <v-dialog v-model="dialog" persistent max-width="500px">
377           <v-card>
378             <v-card-title>
379               <span class="headline">{{ dialogTitle }}</span>
380             </v-card-title>
381             <v-card-text>
382               <v-container grid-list-md>
383                 <v-layout wrap>
384                   <v-flex xs12>
385                     <v-text-field label="번호" v-model="contactInfo.contactId"
386                       required>
387                   </v-text-field>
388                 </v-flex>
389                 <v-flex xs12>
390                   <v-text-field label="이름" v-model="contactInfo.name"
391                     required>

```

```

392         <v-flex xs12>
393             <v-text-field label="이메일" v-model="contactInfo.email"
394                 required>
395             </v-text-field>
396         </v-flex>
397         <v-flex xs12>
398             <v-text-field label="성별" v-model="contactInfo.gender"
399                 required>
400             </v-text-field>
401         </v-flex>
402         <v-flex xs12>
403             <v-text-field label="전화" v-model="contactInfo.phone"
404                 required>
405             </v-text-field>
406         </v-flex>
407         <v-flex xs12>
408             <v-text-field label="등록일" v-model="contactInfo.register_date"
409                 required>
410             </v-text-field>
411         </v-flex>
412     </v-container>
413 </v-card-text>
414 <v-card-actions>
415     <v-spacer></v-spacer>
416     <v-btn color="blue darken-1" text @click="btnClick($event)">취소
417     </v-btn>
418     <v-btn color="blue darken-1" text @click="btnClick($event)">확인
419     </v-btn>
420 </v-card-actions>
421 </v-card>
422 </v-dialog>
423 </template>
424 <template v-slot:item.action="{ item }">
425     <v-btn color="primary" class="mr-2" v-on:click.native="addContact()">추가
426     </v-btn>
427     <v-btn color="primary" class="mr-2" v-on:click.native="updateContact(item)">
428     수정</v-btn>
429     <v-btn color="primary" class="mr-2" v-on:click.native="deleteContact(item)">삭
430     제</v-btn>
431 </template>
432 </v-data-table>
433 </div>
434 </template>
435 <script>
436 import axios from 'axios'
437 export default {
438     data(){
439         return {
440             urlinfo:'http://localhost:8000/contacts',
441             contactInfo : {
442                 _id : null,
443                 contactId:null,
444                 name:null,
445                 email:null,
446                 gender:null,
447                 phone:null,
448                 register_date:null

```



```

441     },
442     dialog:false,
443     dialogTitle :null,
444     headers : [
445         { text:'번호', align:'right', value:'contactId'},
446         { text:'이름', align:'center', value:'name'},
447         { text:'이메일', align:'right', value:'email'},
448         { text:'성별', align:'center', value:'gender'},
449         { text:'전화', align:'center', value:'phone'},
450         { text:'등록일', align:'right', value:'register_date'},
451         { text:'작업', align:'center', value:'action'}
452     ],
453     items : []
454 },
455 },
456 created(){
457     axios.get(this.urlinfo) //Server에 요청하기
458     .then((res) => {
459         //console.log(res.data);
460         this.items = res.data;
461     })
462     .catch((err) => {
463         alert('Error 발생:', err)
464     });
465 },
466 methods : {
467     addContact(){
468         this.dialog = true;
469         this.dialogTitle = "추가";
470         this.contactInfo.register_date = Date.now();
471     },
472     updateContact(data){
473         this.dialog = true;
474         this.dialogTitle = "수정";
475         this.contactInfo.contactId = data.contactId;
476         this.contactInfo.name = data.name;
477         this.contactInfo.email = data.email;
478         this.contactInfo.gender = data.gender;
479         this.contactInfo.phone = data.phone;
480         this.contactInfo.register_date = data.register_date;
481     },
482     deleteContact(data){
483         this.dialog = true;
484         this.dialogTitle = "삭제";
485         this.contactInfo.contactId = data.contactId;
486         //console.log('deleteContact : ' + data);
487     },
488     btnClick($event){
489         this.dialog = false;
490         if($event.target.innerHTML == "확인"){
491             if(this.dialogTitle == "추가"){
492                 axios.post(this.urlinfo,{
493                     contactId: this.contactInfo.contactId, name: this.contactInfo.name, email:
494                     this.contactInfo.email,
495                     gender: this.contactInfo.gender,
496                     phone:this.contactInfo.phone //, register_date:this.contactInfo.register_date
497                 })
498                 .then(() => {

```

```
498     axios.get(this.urlinfo) //서버에 요청하기
499     .then((res) => {
500         //console.log(res.data); //성공시
501         this.items = res.data;
502         alert("연락처 추가 성공");
503     })
504     .catch((err) => {
505         alert('에러 발생: ' + err); //에러 발생
506     });
507
508 })
509 .catch((err) => {
510     alert('에러 발생: ' + err); //에러 발생
511 });
512 }
513 else if(this.dialogTitle=="수정"){
514     //alert("입력된 정보 : " + "이메일 : " + this.contactInfo.email + " 패스워드 : " +
515     this.contactInfo.password);
516     axios.put(this.urlinfo + '/' + this.contactInfo.contactId, {
517         contactId: this.contactInfo.contactId, name: this.contactInfo.name, email:
518         this.contactInfo.email,
519         gender: this.contactInfo.gender,
520         phone: this.contactInfo.phone, register_date: this.contactInfo.register_date
521     })
522     .then(() => {
523         axios.get(this.urlinfo) //서버에 요청하기
524         .then((res) => {
525             //console.log(res.data); //성공시
526             this.items = res.data;
527             alert("업데이트 성공");
528         })
529         .catch((err) => {
530             alert('에러 발생: ' + err); //에러 발생
531         });
532     })
533     .catch((err) => {
534         alert('에러 발생: ' + err); //에러 발생
535     });
536 }
537 else {
538     axios.delete(this.urlinfo + '/' + this.contactInfo.contactId, { data: { contactId:
539     this.contactInfo.contactId } })
540     .then(() => {
541         //console.log("삭제 후" + result); //성공시
542         axios.get(this.urlinfo) //서버에 요청하기
543         .then((res) => {
544             this.items = res.data;
545             alert("삭제 성공");
546         })
547         .catch((err) => {
548             alert('삭제 후 데이터 가져오는 중 에러 발생: ' + err); //에러 발생
549         });
550     })
551     .catch((err) => { alert('에러 발생: ' + err); });
552 }
553 }
554 this.contactInfo.contactId = null;
555 this.contactInfo.name = null;
```

```
553         this.contactInfo.email = null;
554         this.contactInfo.gender = null;
555         this.contactInfo.phone = null;
556         this.contactInfo.register_date = null;
557     }
558 }
559 }
560 </script>
561 <style scoped>
562     div {
563         margin:0 5px 0 5px;
564     }
565 </style>
566
567
568 2. Test
569
570 $ npm run serve
```