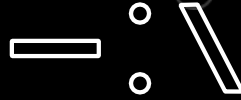


SHOUT OUR PASSION TOGETHER



PASSION

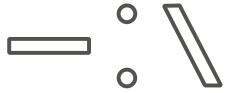
7th Seminar

EXPERIENCE

GROWTH

CHALLENGE

SHOUT OUR PASSION TOGETHER
SOPT



SHOUT OUR PASSION TOGETHER

조원

칠판

1조

2조

5조

3조

4조

6조

01 남궁권 박경선 박시현 심다은 윤혁 지현이

02 김해리 신정아 이재용 양승희 이시연 허정민

03 김강희 신윤재 손예지 양시연 양정훈 현주희

04 강영우 김민준 김채린 이소연 이소희 유희수

05 박형모 이다현 양희찬 이재현 조수민 조하담 황채연

06 박승완 박주연 이동훈 양희연 제갈윤 최소영 최영훈

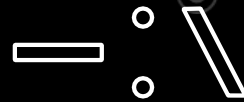
01 moment

02 cron

03 request

04 Error handling

SHOUT OUR PASSION TOGETHER



01

moment

Moment 모듈

자바스크립트에서는 날짜/시간을 다루기 위해 Date 객체를 사용

다양한 출력 함수를 지원하고 있지만 필요한 기능은 직접 만들어야 함

Ex) '2016-06-19', '07:40'와 같은 포맷을 출력하려면 별도 함수를 제작해야 함

Date객체는 Time Zone 설정 또한 까다로움

=> Moment.js는 날짜/시간을 다루기 위한 다양한 기능 제공

Moment 모듈

moment 생성

```
var yesterday = moment('2019-12-06', 'YYYY-MM-DD');
```

```
var timeAble = moment('07:45', 'hh:mm');  
console.log(timeAble.hour(), timeAble.minute());
```

moment 포맷

```
console.log(yesterday.format()); // "2015-06-15T00:00:00+09:00"  
console.log(yesterday.format('YYYY-MM-DD')); // "2015-06-15"  
console.log(yesterday.format('YYYY')); // "2015"  
console.log(yesterday.format('MM')); // "06"  
console.log(yesterday.format('DD')); // "15"
```

Moment 모듈

Timezone 설정하기

```
require('moment-timezone');  
moment.tz.setDefault("Asia/Seoul");
```

Moment 모듈

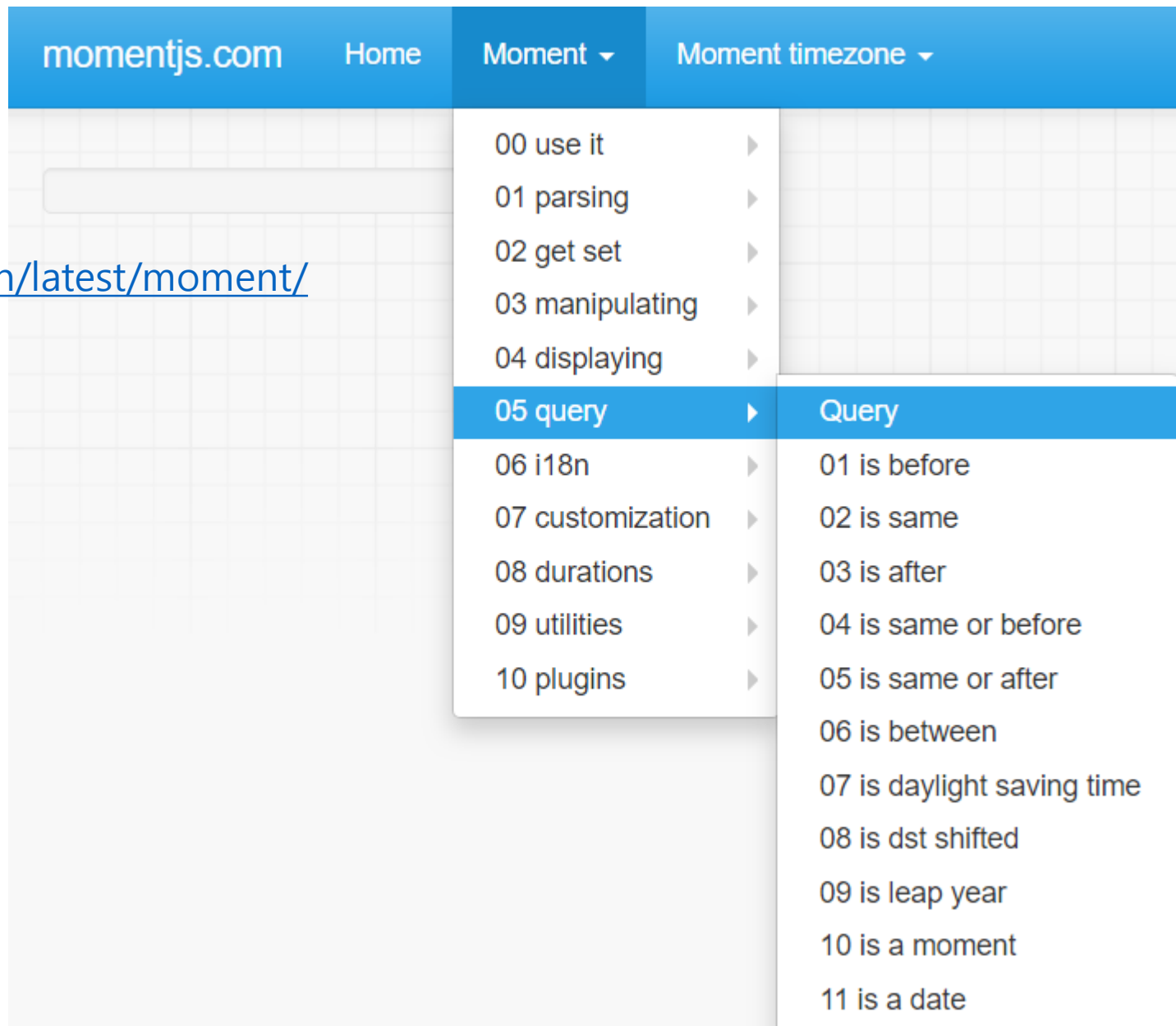
moment 쿼리

```
console.log(moment('2019-12-07').isSame('2010-12-07')); // false
console.log(moment('2019-12-12').isSame('2020-10-21', 'year')); // false
console.log(moment('2019-12-06').isSame('2019-12-06', 'month')); // true
console.log(moment('2019-12-05').isSame('2019-10-21', 'date')); // false

console.log(moment('2019-12-05').isAfter('2019-10-21')); // true
console.log(moment('2019-12-05').isSameOrBefore('2019-10-21')); // false
console.log(moment('2019-12-05').isBetween('2019-12-22', '2020-01-04')); // false
```


그 외 쿼리들은 공식 문서 참고

<https://momentjscom.readthedocs.io/en/latest/moment/05-query/06-is-between/>



Moment 모듈

moment 기간

```
var t1 = new Date(2019, 12, 22);  
var t2 = new Date(2020, 1, 4);  
var diff2 = {  
  seconds: moment.duration(t2 - t1).asSeconds(), // 1123200  
  minutes: moment.duration(t2 - t1).asMinutes(), // 18720  
  hours: moment.duration(t2 - t1).asHours() //312  
};  
console.log(diff2);
```

실행 결과

```
{ seconds: 1123200, minutes: 18720, hours: 312 }
```

EC2 인스턴스 timezone 변경

Timezone을 설정하지 않으면 UST/PST 등으로 기본 다른 국가의 시간으로 나옴

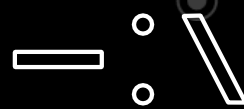
만약 명령어 `date`, `sudo cat /etc/localtime`을 입력했을 때 아래처럼 나올 경우 timezone을 변경해야 함

```
ubuntu@ip-172-31-32-34:~$ date
Fri Dec  6 01:34:40 UTC 2019
ubuntu@ip-172-31-32-34:~$ sudo cat /etc/localtime
TZif2UTCTZif2UTC
UTC0
```

Timezone 변경하는 방법

1. `ls /usr/share/zoneinfo/Asia` 입력 후 Seoul 있는 지 확인
2. `sudo rm /etc/localtime` 입력
3. `sudo ln -sf /usr/share/zoneinfo/Asia/Seoul /etc/localtime`
4. `date` 입력 후 KST 시간으로 나오면 성공

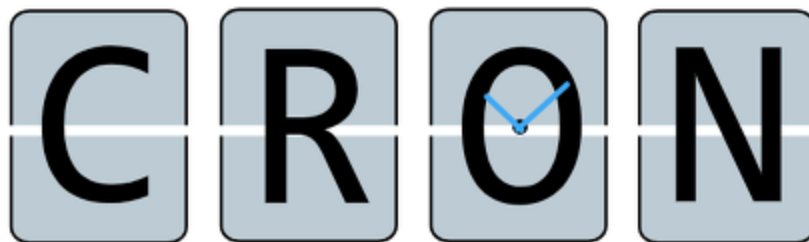
SHOUT OUR PASSION TOGETHER



02

cron

cron



Cron은 유닉스 계열 컴퓨터 운영 체제의 시간 기반 Job 스케줄러
고정된 시간, 날짜, 간격에 주기적으로 실행할 수 있도록 스케줄링하기 위해 사용

node-cron 모듈

GNU crontab을 기반으로 하는 node.js용 순수 자바 스크립트의 가벼운 작업 스케줄러

설치

```
npm install node-cron
```

```
# _____ second (optional)
# | _____ minute
# | | _____ hour
# | | | _____ day of month
# | | | | _____ month
# | | | | | _____ day of week
# | | | | |
# | | | | |
# * * * * *
```

field	value
second	0-59
minute	0-59
hour	0-23
day of month	1-31
month	1-12 (or names)
day of week	0-7 (or names, 0 or 7 are sunday)

node-cron 실행 주기 예제

```
const cron = require('node-cron');
const moment = require('moment');

cron.schedule('* / 10 * * * *', () => console.log('매 10초마다 실행', moment().format()));

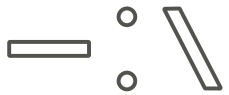
cron.schedule('0-30 / 7 * * * *', () => console.log('0~30사이에 매 7초마다 실행', moment().format()));

cron.schedule('25 * * * *', () => console.log('25초에 실행', moment().format()));

cron.schedule('26,27,28 * * * *', () => console.log('26 또는 27 또는 28초에 실행', moment().format()));

cron.schedule('0 12 * * *', () => console.log('매일 12시에 실행', moment().format()));

cron.schedule('0 12 * * Jan', () => console.log('매주 일요일 12시에 실행', moment().format()));
```



cronManager Module

```
const cron = require('node-cron');
const jobList = [];
module.exports = {
  addTask: (syntax, task, immediate = false) => {
    const job = cron.schedule(syntax, task, {scheduled: immediate});
    const idx = jobList.push(job) - 1;
    return idx;
  },
  startTask: (idx) => jobList[idx].start();
  stopTask: (idx) => jobList[idx].stop();
  destroy: (idx) => jobList[idx].destroy();
  validate: (idx, syntax) => jobList[idx].validate(syntax);
  clear: () => {
    jobList.forEach(it => it.destroy());
    jobList.length = 0;
  }
}
```

Schedule()를 통해서
ScheduledTask객체 생성

```
ScheduledTask {
  start: [Function],
  stop: [Function],
  getStatus: [Function],
  destroy: [Function],
  task: [Function],
  tick: null }
```

Start()
Stop()
Destroy()

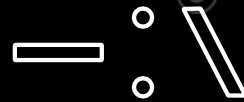
위의 메소드로 제어 가능

cronManager Module Example

```
let lapse = 0;
const idx1 = csvManager.addTask('* / 1 * * * *', () => console.log(`${lapse++}초`, moment().format()));
const idx2 = csvManager.addTask('* / 5 * * * *', () => console.log('매 5초마다 발생', moment().format()), true);

csvManager.startTask(idx1);
setTimeout(() => {
  csvManager.stopTask(idx1);
}, 10000);
setTimeout(() => {
  csvManager.startTask(idx1);
  csvManager.destroy(idx2);
}, 20000);
setInterval(() => {
  csvManager.clear();
}, 50000)
```

SHOUT OUR PASSION TOGETHER



03

request

Request 모듈

서버 내부에서 다른 서버로 request를 보낼 때 사용하는 모듈
다른 서버에 request를 보내 데이터를 받아 옴

```
npm install request
```

```
const request = require('request');
```

```
const url = 'http://www.naver.com';
request.get({
  url: url,
  json: true
}, (err, httpResponse, body) => {
  if (err) {
    console.log(body);
    return;
  }
  console.log(body);
})
```

```
const url = 'http://www.naver.com/...';
request.post({
  url: url,
  body: jsonData,
  json: true
}, (err, httpResponse, body) => {
  if (err) {
    console.log(body);
    return;
  }
  console.log(body);
})
```

Request로 공공데이터 가져오기

공공데이터 포털 가입 -> 서비스 신청 -> 인증키 발급

URL 생성 -> request로 요청

```
const jsonData = {
  ServiceKey: '1Q0QwNz1pQ9tHVUTtXXSvIUaELeSDfRLDAMJk722CSiM6LL7e68V8mmNUeqVnurFinZxM8%2FZf1cEkDeJyRPU%2Fg%3D%3D',
  numOfRows: 10,
  pageNo: 1,
  sidoName: '서울',
  searchCondition: 'HOUR'
};

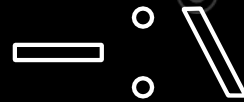
var url = 'http://openapi.airkorea.or.kr/openapi/services/rest/ArpltnInforInquireSvc/getCtprvnMeasureSidoList';
var queryParams = '?' + encodeURIComponent('ServiceKey') + `=${jsonData.ServiceKey}`; /* Service Key*/
queryParams += '&' + encodeURIComponent('numOfRows') + '=' + encodeURIComponent(jsonData.numOfRows); /* 한 페이지 결과 수 */
queryParams += '&' + encodeURIComponent('pageNo') + '=' + encodeURIComponent(jsonData.pageNo); /* 페이지 번호 */
queryParams += '&' + encodeURIComponent('sidoName') + '=' + encodeURIComponent(jsonData.sidoName); /* 시도 이름 (서울, 부산, 대구, 인천, 광주, 대전, 울산, 경기, 강원, 충북, 충남, 전북, 전남, 경북, 경남, 제주, 세종) */
queryParams += '&' + encodeURIComponent('searchCondition') + '=' + encodeURIComponent(jsonData.searchCondition); /* 요청 데이터기간 (시간 : HOUR, 하루 : DAILY) */
queryParams += '&' + encodeURIComponent('_returnType') + '=' + encodeURIComponent('json');
```

Request로 공공데이터 가져오기

응답 수신 후 정보 가공

```
request.get(url + queryParams, (error, response, body) => {  
  console.log('error:', error);  
  console.log('statusCode:', response && response.statusCode);  
  const data = JSON.parse(body);  
  const result = data.list.map(it => {  
    const {cityName, coValue, no2Value, o3Value, pm10Value, pm25Value} = it;  
    return {cityName, coValue, no2Value, o3Value, pm10Value, pm25Value};  
  })  
  console.log(result);  
});
```

SHOUT OUR PASSION TOGETHER



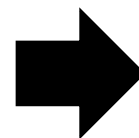
04

Error handling

기존 방식
status 코드와 json을 전달 하면서 처리

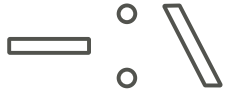
```
return {  
  code: statusCode.OK,  
  json: authUtil.successTrue('메시지', data)  
};
```

```
res.status(statusCode.BAD_REQUEST)  
  .send(authUtil.successFalse('에러 메세지'));
```



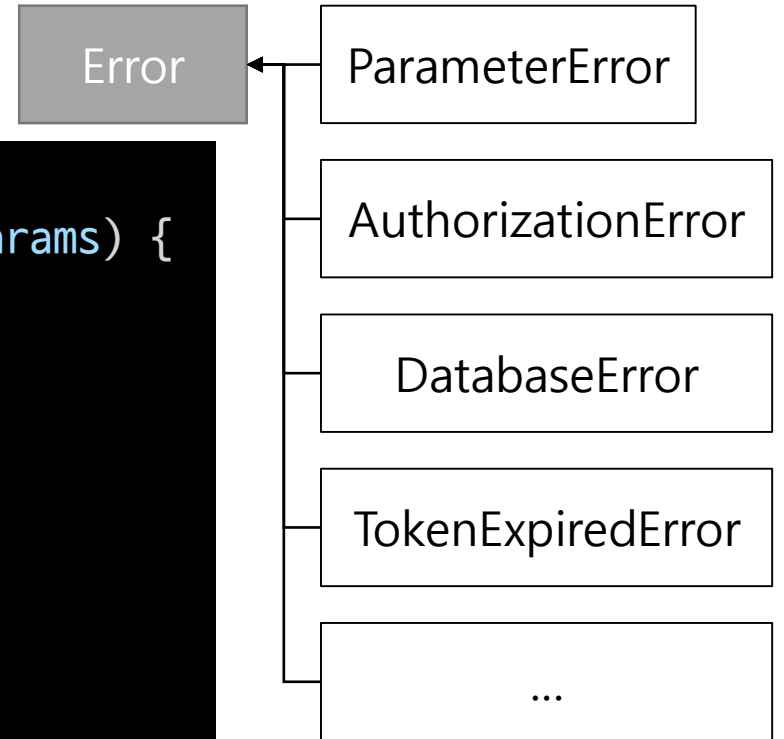
```
if(code && json) { ... }
```

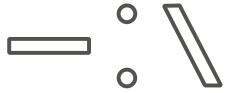
에러 상황인지 확인하기 위해서
매번 code와 json을 체크해야 하는
상황 발생



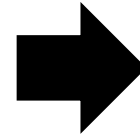
Error를 상속한 Class 정의

```
class DatabaseError extends Error {  
  constructor(code = 'GENERIC', status = statusCode.DB_ERROR, ...params) {  
    super(...params);  
    if (Error.captureStackTrace) {  
      Error.captureStackTrace(this, DatabaseError);  
    }  
    this.code = code;  
    this.status = status;  
    this.message = 에러 메시지;  
  }  
}
```





```
return {  
    code: statusCode.BAD_REQUEST,  
    json: authUtil.successTrue('디비 오류', data)  
};
```



```
throw new DatabaseError();
```

```
return data;
```

에러 상황에서 새로 정의한 Error class를 throw해준다.
또한 code & json 을 응답할 필요 없이 성공한 **결과값**만 바로 반환해준다.

즉 함수는 실패한 경우에는 Error가 발생하며 성공한 경우에만 값을 반환한다.

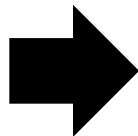
Default error handling in app.js

```
// error handler
app.use(function(err, req, res, next) {
  // set locals, only providing error in development
  res.locals.message = err.message;
  res.locals.error = req.app.get('env') === 'development' ? err : {};

  // render the error page
  res.status(err.status || 500);
  res.render('error');
});
```

변경

```
res.render('error');
```



```
res.send(util.successFalse(err.message));
```

하나의 catch로 모든 에러 처리 가능

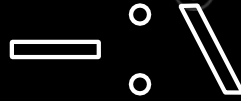
Promise 경우

```
.catch(err => {  
  console.log(err);  
  res.status(err.status || 500);  
  res.send(util.successFalse(err.message));  
});
```

try-catch 경우

```
catch(err) {  
  console.log(err);  
  res.status(err.status || status.INTERNAL_SERVER_ERROR)  
    .send(util.successFalse(err.message));  
}
```

SHOUT OUR PASSION TOGETHER



과제 안내

https://docs.google.com/spreadsheets/d/1hRx_bmkXR3zC91i1cPr7DAo0U4CeDjDVOYG4HpHb_Kw/edit?usp=drive_web&ouid=108665243993139220562

[1차] 라우팅 과제	[2차] node 정리문서	[2차] express 과제	[3차] 블로그/게시글/댓글 구현 과제	[4차] 기능 확장
https://github.com/WITH-SOPT/assignment1				
https://github.com/WITH-SOPT/assignment2		https://github.com/WITH-SOPT/assignment2	https://github.com/WITH-SOPT/assignment2	https://github.com/WITH-SOPT/assignment2
https://github.com/WITH-SOPT/assignment3	https://github.com/luciferkala/SHARED-LEARNING	https://github.com/luciferkala/assignment3	https://github.com/luciferkala/assignment3	https://github.com/luciferkala/assignment3
https://github.com/WITH-SOPT/assignment4				
https://github.com/WITH-SOPT/assignment5	https://github.com/WITH-SOPT/assignment5	https://github.com/siyeons/SOPT_SERVER	https://github.com/siyeons/SOPT_SERVER	https://github.com/siyeons/SOPT_SERVER
https://github.com/seunghyeon/SOPT_SERVER				
https://github.com/leeda66/SOPT_SERVER				
https://github.com/funnkid/SOPT_SERVER				
https://github.com/Ju-Yeon/Backend_Study		https://github.com/Ju-Yeon/Backend_Study	https://github.com/Ju-Yeon/Backend_Study	
https://github.com/ChoHa/SOPT_SERVER				
https://github.com/ooeunz/SOPT_SERVER			https://github.com/ooeunz/submit_server	
https://github.com/kkoon9/SOPT_SERVER	https://github.com/WITH-SOPT_SERVER/SHARED_LEARNING	https://github.com/kkoon9/SOPT_SERVER	https://github.com/kkoon9/SOPT_SERVER	
https://github.com/donghu/SOPT_SERVER	https://github.com/3%EC%A1%B0.md			
https://github.com/chaelin/SOPT_SERVER	https://github.com/WITH-SOPT_SERVER/SHARED_LEARNING			
https://github.com/ganghee/SOPT_SERVER	https://github.com/WITH-SOPT_SERVER/SHARED_LEARNING	https://github.com/ganghee/SOPT_SERVER	https://github.com/ganghee/SOPT_SERVER	https://github.com/ganghee/SOPT_SERVER
https://github.com/ChoSo/SOPT_SERVER	https://github.com/WITH-SOPT_SERVER/SHARED_LEARNING			
https://github.com/artiicy/SOPT_SERVER	https://github.com/WITH-SOPT_SERVER/SHARED_LEARNING	https://github.com/artiicy/SOPT_SERVER	https://github.com/artiicy/SOPT_SERVER	https://github.com/artiicy/SOPT_SERVER
https://github.com/ujusy/SOPT_SERVER	https://github.com/WITH-SOPT_SERVER/SHARED_LEARNING		https://github.com/ujusy/SOPT25_Server/tree/master/sopthw3/hw3	
https://github.com/yoohee/SOPT_SERVER	https://github.com/WITH-SOPT_SERVER/SHARED_LEARNING	https://github.com/yooheesoo/SOPT_SERVER		
https://github.com/tape22/SOPT_SERVER	https://github.com/WITH-SOPT_SERVER/SHARED_LEARNING		https://github.com/tape22/SOPT_SERVER	https://github.com/tape22/SOPT_SERVER
https://github.com/dudgns/SOPT_SERVER	https://github.com/WITH-SOPT_SERVER/SHARED_LEARNING		https://github.com/dudgns3tp/Sopt25th-Ng	https://github.com/dudgns3tp/Sopt25th-Ng
https://github.com/dlthgml/SOPT_SERVER	https://github.com/WITH-SOPT_SERVER/SHARED_LEARNING	https://github.com/dlthgml1997/Sopt25_S	https://github.com/dlthgml1997/Sopt25_S	https://github.com/dlthgml1997/Sopt25_S

검사 완료

과제 검사 완료

Good Case

좋은 시도

try more

조금 더 수정!

retry

잘못된 제출

과제는 데모 데이까지 받음.

제출만 하면 과제 인정
(단 Retry는 무효)

https://docs.google.com/spreadsheets/d/1hRx_bmkXR3zC91i1cPr7DAo0U4CeDjDVOYG4HpHb_Kw/edit?usp=drive_web&ouid=108665243993139220562

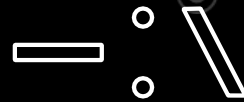
[1차] 라우팅 과제	[2차] node 정리문서	[2차] express 과제	[3차] 블로그/게시글/댓글 구현 과제
https://github.com/WITH-S...			
https://github.com/WITH-S...		https://github.com/WITH-SOPT-SERVER-...	https://github.com/WITH-SOPT-SERVER-...
https://github.com/WITH-S...	https://github.com/luciferkala/SHARED-LEAR...	https://github.com/luciferkala/...	https://github.com/luciferkala/assignment3...
https://github.com/WITH-S...			
https://github.com/WITH-S...	https://github.com/WITH-SOPT-SERVER/SH...	https://github.com/siyeons/SO...	https://github.com/siyeons/SOPT-SERVEI...
https://github.com/seungh...			
https://github.com/leeda8...			
https://github.com/funnkid...			
https://github.com/Ju-Yeo...		https://github.com/Ju-Yeon/Ba...	https://github.com/Ju-Yeon/Backend_Stud...
https://github.com/ChoHa...			
https://github.com/ooeunz...			https://github.com/ooeunz...
https://github.com/kkoon9...	https://github.com/WITH-SOPT-SERVER/SH...	https://github.com/kkoon9/SO...	https://github.com/kkoon9/SOPT-SERVEI...
https://github.com/donghu...	https://github.com/...3%EC%A1%B0.md		
https://github.com/chaelin...	https://github.com/WITH-SOPT-SERVER/SH...		
https://github.com/ganghee...	https://github.com/WITH-SOPT-SERVER/SH...	https://github.com/ganghee/St...	https://github.com/ganghee/SOPT_semin...
https://github.com/ChoSo...	https://github.com/WITH-SOPT-SERVER/SH...		
https://github.com/artiicy/...	https://github.com/WITH-SOPT-SERVER/SH...	https://github.com/artiicy/SOF...	https://github.com/artiicy/SOPT_25th_ass...
https://github.com/ujsy/S...	https://github.com/WITH-SOPT-SERVER/SH...		https://github.com/ujsy/SOPT25_Server/tree...
https://github.com/yoohee...	https://github.com/WITH-SOPT-SERVER/SH...	https://github.com/yooheesoo...	
https://github.com/tape22...	https://github.com/WITH-SOPT-SERVER/SH...		https://github.com/tape22/SOPT_SERVER...
https://github.com/dudgns...	https://github.com/WITH-SOPT-SERVER/SH...		https://github.com/dudgns3tp/Sopt25th-Nr...
https://github.com/dlthgm1...	https://github.com/WITH-SOPT-SERVER/SH...	https://github.com/dlthgm1...	https://github.com/dlthgm1997/Sopt25_S...

파일 수정 보기 삽입 서식 데이터 도구 부가기능 도움말 드라이브에서 모든 변경사항이 저장되었습니다.

100% W % .00 .00 123 기본값 (Ari... 10 B I S A

	A	B	C	D	E	F	G	H	I	J
1	조	이름	[1차] 이름 추가	[1차] 라우팅 과제	[2차] node 정리문서	[2차] express 과제	[3차] 블로그/게시글	[4차] 기능 확장	[5차] 기능 확장2	[6차] 합동세미나
2	1	강영우	Q	Q	X	X	X	X	X	X
3	1	박경선	Q	Q	X	Q	Q	Q	X	X
4	1	양정훈	Q	Q	Q	Q	Q	Q	Q	Q
5	1	박시현	Q	Q	X	X	X	X	X	X
6	1	이시연	Q	Q	Q	Q	Q	Q	X	X
7	2	양승희	Q	Q	X	X	X	X	X	X
8	2	이다현	Q	Q	X	X	X	X	X	X
9	2	황재연	Q	Q	X	X	X	X	X	X
10	2	박주연	Q	Q	X	Q	Q	X	X	X
11	2	조하담	Q	Q	X	X	X	X	X	X
12	2	신윤재	Q	Q	X	X	Q	X	X	X
13	2	남궁권	Q	Q	Q	Q	Q	X	X	X
14	3	이동훈	Q	Q	X	X	X	X	X	X
15	3	김채린	Q	Q	Q	X	X	X	X	X
16	3	김강희	Q	Q	Q	Q	Q	Q	X	X
17	3	조수민	Q	Q	Q	X	X	X	X	X
18	3	김민준	Q	Q	Q	Q	Q	Q	X	X
19	3	이소연	Q	Q	Q	X	Q	X	X	X
20	3	유희수	Q	Q	Q	Q	X	X	X	X
21	4	허정민	Q	Q	Q	X	Q	Q	X	X
22	4	최영훈	Q	Q	Q	X	Q	Q	X	X
23	4	이소희	Q	Q	Q	Q	Q	Q	X	X
24	4	윤혁	Q	X	X	X	X	X	X	X
25	4	신정아	Q	X	X	X	X	X	X	X
26	4	이재용	Q	Q	Q	X	X	X	X	X

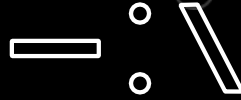
SHOUT OUR PASSION TOGETHER



Let's do

SHOUT OUR PASSION TOGETHER
SOPT

SHOUT OUR PASSION TOGETHER



PASSION

Thank You :)

EXPERIENCE

GROWTH

CHALLENGE

SHOUT OUR PASSION TOGETHER
SOPT