MantaRay API 목차

1. 기술 정보

Mantaray 서비스를 기업 단위로 개발에 사용하기 위해서는 ShTheCode에서
기업용 Token을 발급 받고, 발급받은 Token으로 API호출 시 인증을 해야합니다.

발급 문의
TEL. 070-4077-0327 E-MAIL. help@sothecode.com http://www.sothecode.com/

Bearer 인증 방식을 사용하기 때문에
`Authorization: Bearer <token>`형식으로 HTTP 요청 헤더에 포함합니다.
`Authorization: Bearer a9ace025c90c0da2161075daaadwwfsdcs26d5as13`와 같은
형식으로 작성할 수 있습니다.

모바일 앱 개발 시 서버와의 주요 연동 개발은 다음과 같이 요약할 수 있습니다.

[회원 관리]

연동 명칭	요청 시 파라미터	응답(JSON)	설명
아이디 중복 확인	-중복 확인 ID	-중복 확인 결과	POST방식
회원가입	-회원가입 ID -비밀번호 -기기 고유의 IMEI	Data 멤버 고유번호 회사 고유번호 아이디 암호화된 비밀번호 멤버 IMEI 생성 날짜/시간 Result Success 혹은 fail	·POST방식 ·회원가입 전 아이디 중복 확인이 필요합니다. ·아이디 혹은 IMEI가 동일한 계정이 있을 시 회원가입이 이루어지지 않습니다.
로그인	-회원 ID -비밀번호	User 멤버 고유번호 회사 고유번호 아이디 암호화된 비밀번호	·POST방식 ·로그인에 대한 결과는 Success와 Fail로 나타납니다. ·개발 시 IMEI 로그인 계정의 IMEI로 사용해야 타 기기에서 로그인 시

		회원 상태 생성 일자 업데이트 일자 Result Success 혹은 fail	정상적으로 API 호출이 이루어질 수 있습니다.
회원 IMEI 업데이트	-멤버 고유번호 -변경할 IMEI	Data [1] Result Success 혹은 fail	·POST방식 ·동일한 IMEI로 변경은 불가합니다.
회원 상태 변경	-멤버 고유번호 -변경할 상태	Data [1] Result Success 혹은 fail	·POST방식 ·기본적으로 Mantaray 서비스에서는 'P' : 사용중 , 'S' : 정지, 'D' : 사용안함 이렇게 세 가지 파라미터를 사용합니다.

[디바이스 관리]

연동 명칭	요청 시 파라미터	응답(JSON)	설명
디바이스 전체 리스트	-회원 IMEI	Data 디바이스 리스트 Result Success 혹은 fail	·GET방식 ·회원이 등록한 모든 디바이스에 대한 목록을 가져옵니다. ·디바이스 목록에는 디바이스 정보와 디바이스에 귀속된 센서의 리스트가 출력이 됩니다.
디바이스 타입별 리스트	-회원 IMEI -디바이스 유형	Data 디바이스 리스트 Result Success 혹은 fail	·GET방식 ·회원이 등록한 디바이스 중 파라미터로 넣은 유형에 대한 목록을 가져옵니다. ·디바이스 유형은 "ph"와 "temp"로 구분됩니다
센서 전체 리스트	-회원 IMEI -디바이스 유형 -디바이스 고유 번호	Data 디바이스 정보 디바이스에 귀속된 센서 Result Success 혹은 fail	·GET방식 ·회원이 등록한 디바이스 중 파라미터에 넣은 디바이스 고유 번호에 해당하는 디바이스의 정보와 디바이스에 귀속된 센서 목록을 가져옵니다. ·디바이스 유형은 "ph"와 "temp"로 구분됩니다
단일 센서 정보	-회원 IMEI -디바이스 유형 -디바이스 고유 번호 -센서의 SEQ	Data 센서 정보 Result Success 혹은 fail	·GET방식 ·회원이 등록한 디바이스 중 파라미터에 넣은 디바이스의 SEQ에 해당하는 센서의 정보를 가져옵니다. ·디바이스 유형은 "ph"와 "temp"로 구분됩니다

디바이스 생성	-회원 IMEI -디바이스 이름 -디바이스 유형	Data 디바이스 고유번호 멤버 고유번호 디바이스 타입 디바이스 이름 디바이스 고유 토큰 Result Success 혹은 fail	·POST방식 ·디바이스를 생성합니다. ·디바이스 유형은 "ph"와 "temp"로 구분됩니다
디바이스 업데이트	-회원 IMEI -디바이스 고유번호 -디바이스 이름 -디바이스 유형	Data [1] Result Success 혹은 fail	·POST방식 ·디바이스의 이름과 유형을 수정합니다. ·디바이스 유형은 "ph"와 "temp"로 구분됩니다
디바이스 상태 변경	-회원 IMEI -디바이스 고유번호 -변경할 상태	Data [1] Result Success 혹은 fail	·POST방식 ·디바이스의 상태를 변경합니다. ·기본적으로 Mantaray 서비스에서는 'P' : 사용중 , 'S' : 정지, 'D' : 사용안함 이렇게 세 가지 파라미터를 사용합니다.
센서 등록	-회원 IMEI -디바이스 고유번호 -센서 이름 -센서 설정값	Data 센서 고유 번호 디바이스 고유 번호 센서 이름 센서 설정값 센서 SEQ Result Success 혹은 fail	·POST방식 ·센서 설정값은 센서의 최소값/최대값을 설정하기 위한 파라미터입니다.
센서 업데이트	-회원 IMEI -디바이스 고유번호 -센서SEQ -센서 이름 -센서 설정값	Data [1] Result Success 혹은 fail	·POST방식 ·센서의 설정값을 수정합니다.
센서 상태 변경	-회원 IMEI -디바이스 고유번호 -센서SEQ -변경할 상태	Data [1] Result Success 혹은 fail	·POST방식 ·센서의 상태를 수정합니다. ·기본적으로 Mantaray 서비스에서는 'P' : 사용중 , 'S' : 정지, 'D' : 사용안함 이렇게 세 가지 파라미터를 사용합니다.
토큰 등록	-디바이스 고유 토큰	Data [1] Result Success 혹은 fail	·POST방식 ·토큰을 등록시켜 디바이스를 활성화합니다. ·등록할 토큰을 헤더에 넣어서 호출합니다.

[로그 관리]

연동 명칭	요청 시 파라미터	응답(JSON)	설명
로그 전체 리스트	-회원 IMEI	Data 활성화 된 디바이스 목록 Result Success 혹은 fail	·GET방식 ·로그가 존재하는 모든 디바이스의 목록을 가져옵니다. ·디바이스 유형에 관계없이 모두 출력됩니다.
로그 디바이스 타입 리스트	-회원 IMEI -디바이스 유형	Data 활성화 된 디바이스 목록 Result Success 혹은 fail	·GET방식 ·로그가 존재하는 디바이스의 목록을 가져옵니다. ·파라미터에 사용된 유형에 따라 출력이 됩니다.
로그 센서 전체 리스트	-회원 IMEI -디바이스 유형 -디바이스 고유 번호 -날짜	Data 디바이스에 귀속된 센서목록의 날짜별 전체 로그 Result Success 혹은 fail	·POST방식 ·디바이스에 귀속된 센서들의 날짜에 따른 전체 로그를 가져옵니다.
로그 센서 단일 리스트	-회원 IMEI -디바이스 유형 -디바이스 고유 번호 -센서SEQ -날짜	Data 디바이스에 귀속된 센서목록의 날짜별 전체 로그 Result Success 혹은 fail	·POST방식 ·단일 센서의 날짜에 따른 전체 로그를 가져옵니다.
이벤트 전체 리스트	-회원 IMEI	Data 이벤트 로그 목록 Result Success 혹은 fail	·GET방식 ·회원에게 귀속된 센서들의 모든 이벤트 로그를 가져옵니다.
이벤트 타입 리스트	-회원 IMEI -디바이스 유형	Data 이벤트 로그 목록 Result Success 혹은 fail	·GET방식 ·회원에게 귀속된 센서 중 디바이스 유형에 맞는 이벤트 로그를 가져옵니다.
이벤트 센서 전체 리스트	-회원 IMEI -디바이스 유형 -디바이스 고유 번호	Data 이벤트 로그 목록 Result Success 혹은 fail	·GET방식 ·단일 디바이스의 전체 이벤트 로그를 가져옵니다.
이벤트 센서 단일 리스트	-회원 IMEI -디바이스 유형 -디바이스 고유 번호 -센서SEQ -날짜	Data 이벤트 로그 목록 Result Success 혹은 fail	·GET방식 ·단일 센서의 날짜별 이벤트 로그를 가져옵니다.
이벤트 센서 단일 리스트 전체	-회원 IMEI -디바이스 유형 -디바이스 고유 번호 -센서SEQ	Data 이벤트 로그 목록 Result Success 혹은 fail	·GET방식 ·단일 센서의 이벤트 로그 전체를 가져옵니다.

로그 입력	Json 형식의 Raw로 디바이스 토큰과 센서값 전달	Data [1] Result Success 혹은 fail	·POST방식 사용 ·로그를 입력합니다. ·Json 형식으로 전달하며 구성은 디바이스 토큰과 센서값입니다.

2. 회원 관리

2.1 아이디 중복 확인

회원가입 이전에 아이디 중복 확인을 통하여 동일한 아이디의 존재 유무를 체크한 후 회원가입을 진행해주셔야 합니다.

URL - http://13.209.241.209/api/member/companyMemberCheck

전송 방식 - POST (application/x-www-form-urlencoded)

파라미터	파라미터 값	설명
id	중복 확인을 할 ID	회원가입 전 아이디 중복 확인을 합니다.

JAVA OK HTTP 예시

```
OkHttpClient client = new OkHttpClient();
MediaType mediaType = MediaType.parse("application/x-www-form-urlencoded");
RequestBody body = RequestBody.create(mediaType, "id=YOUR_ID");
Request request = new Request.Builder()
  .url("http://13.209.241.209/api/member/companyMemberCheck")
  .post(body)
  .addHeader("Content-Type", "application/x-www-form-urlencoded")
  .addHeader("Authorization", "Bearer YOUR_COMPANY_TOKEN")
  .build();

Response response = client.newCall(request).execute();
```

결과 예시

```
{
    "result": "success"
}
```

2.2 회원가입

회원가입을 하기 이전에 아이디 중복 체크를 통해 중복된 아이디가 존재하는지 확인해주신 후 중복체크가 완료되고 회원가입이 이루어 질 수 있게 제작이 되어야합니다.
한 기기에서 회원가입이 한번 이루어 지면 회원의 IMEI가 바뀌지 않는 이상
같은 기기로 다중 회원가입이 불가능합니다.

URL - http://13.209.241.209/api/member/insertCompanyMember

전송 방식 - POST (application/x-www-form-urlencoded)

파라미터	파라미터 값	설명
Id Imei pw	회원가입을 할 멤버 ID 기기의 고유 IMEI 비밀번호	·회원가입 전 아이디 중복 확인이 필요합니다. ·아이디 혹은 IMEI가 동일한 계정이 있을 시 회원가입이 이루어지지 않습니다

JAVA OK HTTP 예시

```
OkHttpClient client = new OkHttpClient();
MediaType mediaType = MediaType.parse("application/x-www-form-urlencoded");
RequestBody body = RequestBody.create(mediaType, "id=YOUR_ID&
imei=YOUR_IMEI&pw=YOUR_PASSWORD");
Request request = new Request.Builder()
  .url("http://13.209.241.209/api/member/insertCompanyMember")
  .post(body)
  .addHeader("Content-Type", "application/x-www-form-urlencoded")
  .addHeader("Authorization", "Bearer YOUR_COMPANY_TOKEN")
  .build();

Response response = client.newCall(request).execute();
```

결과 예시


```
{
    "data": {
        "UPDATE_DE": {
            "val": "now()"
        },
        "MBER_IDX": 0,
        "COMPANY_IDX": 1,
        "MBER_ID": "ID",
        "MBER_PW": "$2a$10$tYsgRoJZKtN13UvhW1ZoQ.v/B9GM6l637g7GIyeH3hyQBi4SiA",
        "MBER_IMEI": "IMEI",
        "REG_DE": "2019-10-10T20:44:07.582Z"
    },
    "result": "success"
}
```


2.3 로그인

로그인 API 호출 시 User에 대한 정보가 결과로 출력이 됩니다.
모바일 앱 개발 시 로그인 결과로 나오는 IMEI로 모든 호출에 대응해야 가입한 기기가 아닌 다른 기기로 로그인이 되어도 정상적으로 멤버 디바이스에 대한 결과를 얻어올 수 있습니다.
가입된 기기 외에 다른 기기에서 로그인을 원치 않을 경우 MBER_IMEI와
기기의 IMEI를 비교하여 로그인 승인을 처리해주시면 되겠습니다.

URL - http://13.209.241.209/api/member/login
전송 방식 - POST (application/x-www-form-urlencoded)

파라미터	파라미터 값	설명
Id pw	멤버 ID 비밀번호	·로그인 ·개발 시 IMEI 로그인 계정의 IMEI로 사용해야 타 기기에서 로그인 시 정상적으로 API 호출이 이루어질 수 있습니다.

JAVA OK HTTP 예시

```
OkHttpClient client = new OkHttpClient();

MediaType mediaType = MediaType.parse("application/x-www-form-urlencoded");

RequestBody body = RequestBody.create(mediaType, "id=YOUR_ID&pw=YOUR_PASSWORD");

Request request = new Request.Builder()

  .url("http://13.209.241.209/api/member/companyMemberLogin")

  .post(body)

  .addHeader("Content-Type", "application/x-www-form-urlencoded")

  .addHeader("Authorization", "Bearer YOUR_COMPANY_TOKEN")

  .build();


Response response = client.newCall(request).execute();
```

결과 예시

```
{
    "user": {
        "MBER_IDX": 1,
        "COMPANY_IDX": 1,
        "MBER_ID": " ID ",
        "MBER_PW": "$2a$10$5o5SGVvCOG4JSI7PaUJulOKJ18Ty9N7427nH93kTGl6zow",
        "MBER_IMEI": " IMEI ",
        "MBER_TOKEN": null,
        "MBER_STATUS": "P",
        "REG_DE": "2019-10-10T04:28:29.000Z",
        "UPDATE_DE": "2019-10-10T04:25:52.000Z"
    },
    "result": "success"
}
```

2.4 회원 IMEI 업데이트

회원 IMEI 업데이트로 가입된 회원의 IMEI를 변경할 수 있습니다.
기기 변경 시 사용하실 수 있습니다.

URL - http://13.209.241.209/api/member/companyMemberUpdate
전송 방식 - POST (application/x-www-form-urlencoded)

파라미터	파라미터 값	설명
memberIdx imei	멤버 고유 번호 수정할 IMEI	회원의 IMEI를 수정합니다.

JAVA OK HTTP 예시

```
OkHttpClient client = new OkHttpClient();

MediaType mediaType = MediaType.parse("application/x-www-form-urlencoded");
RequestBody body = RequestBody.create(mediaType,
"memberIdx=YOUR_MEMBERIDX&imei=YOUR_IMEI");
Request request = new Request.Builder()
  .url("http://13.209.241.209/api/member/companyMemberUpdate")
  .post(body)
  .addHeader("Content-Type", "application/x-www-form-urlencoded")
  .addHeader("Authorization", "Bearer YOUR_COMPANY_TOKEN ")
.build();

Response response = client.newCall(request).execute();
```

결과 예시

```
{
    "data": [1],
    "result": "success"
}
```

2.5 회원 상태 변경

회원 계정의 상태를 변경합니다.

상태는 'P' : 사용중 'S' : 정지, 'D' : 사용안함 이렇게 세 가지가 파라미터로 구분합니다.

URL - http://13.209.241.209 /api/member/companyMemberStatus

전송 방식 - POST (application/x-www-form-urlencoded)

파라미터	파라미터 값	설명
memberIdx status	멤버 고유 번호 수정할 상태 문자	회원의 계정 상태를 수정합니다.

JAVA OK HTTP 예시

```
OkHttpClient client = new OkHttpClient();

MediaType mediaType = MediaType.parse("application/x-www-form-urlencoded");
RequestBody body = RequestBody.create(mediaType,
"memberIdx= YOUR_MEMBERIDX &status=STATUS");
Request request = new Request.Builder()
  .url("http://13.209.241.209/api/member/companyMemberStatus")
  .post(body)
  .addHeader("Content-Type", "application/x-www-form-urlencoded")
  .addHeader("Authorization", "Bearer YOUR_COMPANY_TOKEN ")
.build();

Response response = client.newCall(request).execute();
```

결과 예시

```
{
    "data": [1],
    "result": "success"
}
```

3. 디바이스 관리

3.1 디바이스 전체 리스트

회원 계정에 귀속된 모든 디바이스의 목록을 가져옵니다.
디바이스 목록에는 디바이스 정보와 디바이스 내에 생성된 센서 목록이 출력됩니다.

호출 결과에 대한 값들의 설명은 아래 표를 참고해주세요.

[디바이스 정보]

이름	설명
DEVICE_INFO_IDX	디바이스 고유 번호
MBER_IDX	멤버 고유 번호
DEVICE_INFO_NAME	디바이스 이름
DEVICE_INFO_TOKEN	디바이스 토큰
DEVICE_INFO_TOKEN_REGIST	디바이스 활성화 유무
DEVICE_INFO_STATUS	디바이스 상태
DEVICE_INFO_SENSOR_TYPE	디바이스 유형
REG_DE	생성 날짜
UPDATE_DE	마지막 업데이트 날짜
SENSOR	센서 목록

[센서 정보]

이름	설명
SENSOR_INFO_IDX	센서 고유 번호
DEVICE_INFO_IDX	디바이스 고유 번호
SENSOR_INFO_NAME	센서 이름
SENSOR_INFO_CONTENT	센서 설정값
SENSOR_INFO_STATUS	센서 상태
SENSOR_INFO_SEQ	센서 SEQ
REG_DE	생성 날짜
UPDATE_DE	마지막 업데이트 날짜

URL - http://13.209.241.209 /api/device/{ imei }/list

전송 방식 - GET

파라미터	파라미터 값	설명
imei	회원 IMEI	·회원이 등록한 모든 디바이스에 대한 목록을 가져옵니다. ·디바이스 목록에는 디바이스 정보와 디바이스에 귀속된 센서의 리스트가 출력이 됩니다.

JAVA OK HTTP 예시

```
OkHttpClient client = new OkHttpClient();


Request request = new Request.Builder()
  .url("http://13.209.241.209/api/device/ YOUR_IMEI/list")
  .get()
  .addHeader("Authorization", "Bearer YOUR_COMPANY_TOKEN ")
  .build();


Response response = client.newCall(request).execute();
```

결과 예시


```
{
"data": [
        {
            "DEVICE_INFO_IDX": 1,
            "MBER_IDX": 1,
            "DEVICE_INFO_NAME": "ABC",
            "DEVICE_INFO_TOKEN": "15f3f53dc6091eb774d1ad7fagrwfvsdfa530a30",
            "DEVICE_INFO_TOKEN_REGIST": "Y",
            "DEVICE_INFO_STATUS": "S",
            "DEVICE_INFO_SENSOR_TYPE": 1,
            "REG_DE": "2019-10-10T04:37:05.000Z",
            "UPDATE_DE": "2018-10-10T05:48:08.000Z",
            "SENSOR": [
```

```
                {
                    "SENSOR_INFO_IDX": 10,
                    "DEVICE_INFO_IDX": 1,
                    "SENSOR_INFO_NAME": "sensor1",
                    "SENSOR_INFO_CONTENT": "temp:10/50,ph:10/50",
                    "SENSOR_INFO_STATUS": "P",
                    "SENSOR_INFO_SEQ": 0,
                    "REG_DE": "2019-11-08T04:37:56.000Z",
                    "UPDATE_DE": "2019-11-14T05:48:08.000Z"
                },
                {
                    "SENSOR_INFO_IDX": 11,
                    "DEVICE_INFO_IDX": 1,
                    "SENSOR_INFO_NAME": "sensor2",
                    "SENSOR_INFO_CONTENT": "temp:5/10,ph:5/30",
                    "SENSOR_INFO_STATUS": "D",
                    "SENSOR_INFO_SEQ": 1,
                    "REG_DE": "2019-10-10T10:58:26.000Z",
                    "UPDATE_DE": "2019-10-10T05:48:08.000Z"
                }
            ]
        }... ],
    "result": "success"
}
```


3.2 디바이스 타입별 리스트

회원에게 귀속된 디바이스 중 파라미터로 넣은 type에 해당하는 디바이스들에 대한 목록을 출력합니다.

URL - http://13.209.241.209 /api/device/**{ imei }**/**{ type }**/list

전송 방식 - GET

파라미터	파라미터 값	설명
Imei type	회원 IMEI 디바이스 유형	·회원이 등록한 디바이스 중 파라미터로 넣은 유형에 대한 목록을 가져옵니다. ·디바이스 유형은 "ph"와 "temp"로 구분됩니다

JAVA OK HTTP 예시

```
OkHttpClient client = new OkHttpClient();

Request request = new Request.Builder()
  .url("http://13.209.241.209/api/log/device/YOUR_IMEI/TYPE/list")
  .get()
  .addHeader("Authorization", "Bearer YOUR_COMPANY_TOKEN ")
  .build();
Response response = client.newCall(request).execute();
```

결과 예시


```
{
    "data": [
        {
"DEVICE_INFO_IDX": 1,
            "MBER_IDX": 1,
            "DEVICE_INFO_NAME": "device1",
            "DEVICE_INFO_TOKEN": "0bee6ac197855899d397esdgfsdfg126c535ffe1",
            "DEVICE_INFO_TOKEN_REGIST": "N",
            "DEVICE_INFO_STATUS": "D",
            "DEVICE_INFO_SENSOR_TYPE": 2,
            "REG_DE": "2019-11-10T18:51:01.000Z",
            "UPDATE_DE": "2019-10-10T22:44:59.000Z",
            "SENSOR": []
        },
        {
            "DEVICE_INFO_IDX": 2,
            "MBER_IDX": 1,
```

```
            "DEVICE_INFO_NAME": "device2",
            "DEVICE_INFO_TOKEN": "a2dbee315623d95214d0cefa5f798ad7ce923656",
            "DEVICE_INFO_TOKEN_REGIST": "Y",
            "DEVICE_INFO_STATUS": "P",
            "DEVICE_INFO_SENSOR_TYPE": 2,
            "REG_DE": "2019-10-10T18:54:03.000Z",
            "UPDATE_DE": "2019-10-10T22:41:28.000Z",
            "SENSOR": [
                {
                    "SENSOR_INFO_IDX": 4,
                    "DEVICE_INFO_IDX": 2,
                    "SENSOR_INFO_NAME": "sensor1",
                    "SENSOR_INFO_CONTENT": "temp:10/20,humi:10/90",
                    "SENSOR_INFO_STATUS": "D",
                    "SENSOR_INFO_SEQ": 0,
                    "REG_DE": "2019-10-10T18:58:26.000Z",
                    "UPDATE_DE": "2019-10-10T22:41:20.000Z"
                }
            ]
        },
        {
            "DEVICE_INFO_IDX": 3,
            "MBER_IDX": 1,
            "DEVICE_INFO_NAME": "device3",
            "DEVICE_INFO_TOKEN": "79de2fsdfge5093f657f795faba0cc27f8560",
            "DEVICE_INFO_TOKEN_REGIST": "N",
            "DEVICE_INFO_STATUS": "S",
            "DEVICE_INFO_SENSOR_TYPE": 2,
            "REG_DE": "2019-10-10T22:34:56.000Z",
            "UPDATE_DE": "2019-10-10T22:38:05.000Z",
            "SENSOR": []
        }
...
    ],
    "result": "success"
```

}

3.3 센서 전체 리스트

회원에게 귀속된 디바이스 중 device 파라미터에 넣은 디바이스 고유 번호에 해당하는 디바이스의 정보와 디바이스에 포함된 센서 목록을 가져옵니다.

URL - http://13.209.241.209/api/device/**{imei}**/**{type}**/**{device}**/list

전송 방식 - GET

파라미터	파라미터 값	설명
Imei Type device	회원 IMEI 디바이스 유형 디바이스 고유 번호	·회원이 등록한 디바이스 중 파라미터에 넣은 디바이스 고유 번호에 해당하는 디바이스의 정보와 디바이스에 귀속된 센서 목록을 가져옵니다. ·디바이스 유형은 "ph"와 "temp"로 구분됩니다

JAVA OK HTTP 예시

```
OkHttpClient client = new OkHttpClient();

MediaType mediaType = MediaType.parse("application/x-www-form-urlencoded");
Request request = new Request.Builder()
  .url("http://13.209.241.209/api/log/device/{imei}/{type}/{device}/list")
  .get()
  .addHeader("Authorization", "Bearer YOUR_COMPANY_TOKEN ")
  .build();

Response response = client.newCall(request).execute();
```

결과 예시


```
{
    "data": [
        {
            "DEVICE_INFO_IDX": 1,
            "MBER_IDX": 2,
            "DEVICE_INFO_NAME": "device1",
            "DEVICE_INFO_TOKEN": "03c5be240fb02db48fdgr5f113fd123a493b9e60",
            "DEVICE_INFO_TOKEN_REGIST": "Y",
            "DEVICE_INFO_STATUS": "D",
            "DEVICE_INFO_SENSOR_TYPE": 2,
            "REG_DE": "2019-10-10T08:13:28.000Z",
            "UPDATE_DE": "2019-10-10T22:45:09.000Z",
            "SENSOR": [
                {
                    "SENSOR_INFO_IDX": 16,
                    "DEVICE_INFO_IDX": 1,
                    "SENSOR_INFO_NAME": "sensor4",
                    "SENSOR_INFO_CONTENT": "temp:10/50,humi:10/70",
                    "SENSOR_INFO_STATUS": "D",
                    "SENSOR_INFO_SEQ": 0,
                    "REG_DE": "2019-10-10T08:13:46.000Z",
                    "UPDATE_DE": "2019-10-10T11:31:19.000Z"
                }
...
            ]
        }
    ],
    "result": "success"
}
```

3.4 단일 센서 정보

회원이 등록한 디바이스 중 파라미터에 넣은 디바이스의 SEQ에 해당하는
단일 센서의 정보를 가져옵니다.

URL - http://13.209.241.209/api/device/{imei}/{type}/{device}/{seq}/list

전송 방식 - GET

파라미터	파라미터 값	설명
Imei Type Device seq	회원 IMEI 디바이스 유형 디바이스 고유 번호 센서SEQ	·회원이 등록한 디바이스 중 파라미터에 넣은 디바이스의 SEQ에 해당하는 센서의 정보를 가져옵니다. ·디바이스 유형은 "ph"와 "temp"로 구분됩니다

JAVA OK HTTP 예시

```
OkHttpClient client = new OkHttpClient();

Request request = new Request.Builder()
   .url("http://13.209.241.209/api/device/{imei}/{type}/{device}/{seq}/list")
   .get()
   .addHeader("Authorization", "Bearer YOUR_COMPANY_TOKEN ")
.build();

Response response = client.newCall(request).execute();
```

결과 예시

```
{
"data": [
                        {
```

```
                "SENSOR_INFO_IDX": 15,
                "DEVICE_INFO_IDX": 1,
                "SENSOR_INFO_NAME": "sensor1",
                "SENSOR_INFO_CONTENT": "temp:5/50,ph:20/40",
                "SENSOR_INFO_STATUS": "D",
                "SENSOR_INFO_SEQ": 0,
                "REG_DE": "2019-10-10T04:37:56.000Z",
                "UPDATE_DE": "2019-10-10T05:48:08.000Z"
            },
],
"result": "success"
}
```


3.5 디바이스 생성

디바이스를 등록합니다.

URL - http://13.209.241.209/api/device/{imei}/insert

전송 방식 - POST (application/x-www-form-urlencoded)

파라미터	파라미터 값	설명
Imei **deviceName** **type**	회원 IMEI 디바이스 이름 디바이스 유형	·디바이스를 등록합니다. ·디바이스 유형은 "ph"와 "temp"로 구분됩니다

JAVA OK HTTP 예시

```
OkHttpClient client = new OkHttpClient();

MediaType mediaType = MediaType.parse("application/x-www-form-urlencoded");
RequestBody body
= RequestBody.create(mediaType, "deviceName=DEVICE_NAME&type=TYPE");
Request request = new Request.Builder()
```

```
  .url("http://13.209.241.209/api/device/{imei}/insert")
  .post(body)
   .addHeader("Content-Type", "application/x-www-form-urlencoded")
  .addHeader("Authorization", "Bearer YOUR_COMPANY_TOKEN ")
  .build();

Response response = client.newCall(request).execute();
```

결과 예시


```
{
    "data": {
        "REG_DE": {
            "val": "now()"
        },
        "UPDATE_DE": {
            "val": "now()"
        },
        "DEVICE_INFO_IDX": 1,
        "MBER_IDX": 1,
        "DEVICE_INFO_SENSOR_TYPE": 2,
        "DEVICE_INFO_NAME": "Device1",
        "DEVICE_INFO_TOKEN": "a2dbee309030895214d0ce8ad7ce9afegrty3656"
    },
    "result": "success"
}
```


3.6 디바이스 업데이트

디바이스의 이름, 유형을 변경합니다.

URL - http://13.209.241.209/api/device/{imei}/{device}/update

전송 방식 - POST (application/x-www-form-urlencoded)

파라미터	파라미터 값	설명
Imei Device deviceName type	회원 IMEI 디바이스 고유 번호 변경 할 이름 디바이스 유형	·디바이스의 이름, 유형을 변경합니다. ·디바이스 유형은 "ph"와 "temp"로 구분됩니다

JAVA OK HTTP 예시

```
OkHttpClient client = new OkHttpClient();

MediaType mediaType = MediaType.parse("application/x-www-form-urlencoded");
RequestBody body = RequestBody.create(mediaType, "deviceName=DEVICE_NAME&type=TYPE");
Request request = new Request.Builder()
  .url("http://13.209.241.209/api/device/{imei}/{device}/update")
  .post(body)
  .addHeader("Content-Type", "application/x-www-form-urlencoded")
  .addHeader("Authorization", "Bearer YOUR_COMPANY_TOKEN ")
.build();

Response response = client.newCall(request).execute();
```

결과 예시

```
{
    "data": [0],
    "result": "success"
}
```

3.7 디바이스 상태 변경

디바이스의 상태를 변경합니다.

URL - http://13.209.241.209/api/device/{imei}/{device}/use

전송 방식 - POST (application/x-www-form-urlencoded)

파라미터	파라미터 값	설명
Imei Device status	회원 IMEI 디바이스 고유 번호 변경할 상태	·디바이스의 상태를 변경합니다. ·기본적으로 Mantaray 서비스에서는 'P' : 사용중 , 'S' : 정지, 'D' : 사용안함 이렇게 세 가지 파라미터를 사용합니다.

JAVA OK HTTP 예시

```
OkHttpClient client = new OkHttpClient();


MediaType mediaType = MediaType.parse("application/x-www-form-urlencoded");
RequestBody body = RequestBody.create(mediaType, "status=STATUS");
Request request = new Request.Builder()
  .url("http://13.209.241.209/api/device/{imei}/{device}/use")
  .post(body)
  .addHeader("Content-Type", "application/x-www-form-urlencoded")
  .addHeader("Authorization", "Bearer YOUR_COMPANY_TOKEN ")
  .build();


Response response = client.newCall(request).execute();
```

결과 예시

```
{
    "data": [0],
    "result": "success"
}
```

3.8 센서 등록

센서를 등록합니다.

센서의 설정값은 센서의 최소값/최대값을 설정하기 위한 파라미터입니다.

설정값은 temp:**{최소값}**/**{최대값}**,humi:**{최소값}**/**{최대값}**"의 형태로 파라미터를 입력합니다,

URL - http://13.209.241.209/api/device/{imei}/{device}//insert

전송 방식 - POST (application/x-www-form-urlencoded)

파라미터	파라미터 값	설명
Imei Device sensorName content	회원 IMEI 디바이스 고유 번호 센서 이름 센서 설정값	·센서를 등록합니다. ·센서 설정값은 센서의 최소값/최대값을 설정하기 위한 파라미터입니다.

JAVA OK HTTP 예시

```
OkHttpClient client = new OkHttpClient();

MediaType mediaType = MediaType.parse("application/x-www-form-urlencoded");
RequestBody body =
RequestBody.create(mediaType, "sensorName=SENSOR_NAME&content=SENSOR_SETTING_VALUE");
Request request = new Request.Builder()
  .url("http://13.209.241.209/api/device/{imei}/{device}/insert")
  .post(body)
  .addHeader("Content-Type", "application/x-www-form-urlencoded")
  .addHeader("Authorization", "Bearer YOUR_COMPANY_TOKEN ")
  .build();

Response response = client.newCall(request).execute();
```

결과 예시


```
{
    "data": {
        "REG_DE": {
            "val": "now()"
        },
        "UPDATE_DE": {
            "val": "now()"
        },
        "SENSOR_INFO_IDX": 1,
        "DEVICE_INFO_IDX": "1",
        "SENSOR_INFO_NAME": " SENSOR_NAME ",
        "SENSOR_INFO_CONTENT": " SENSOR_SETTING_VALUE ",
        "SENSOR_INFO_SEQ": 0
    },
    "result": "success"
}
```


3.9 센서 업데이트

센서의 이름과 설정값을 수정합니다.

설정값은 temp:**{최소값}**/**{최대값}**,humi:**{최소값}**/**{최대값}**"의 형태로 파라미터를 입력합니다,

URL - http://13.209.241.209/api/device/{imei}/{device}/{seq}/update

전송 방식 - POST (application/x-www-form-urlencoded)

파라미터	파라미터 값	설명
Imei Device Seq sensorName content	회원 IMEI 디바이스 고유 번호 센서seq 센서 이름 센서 설정값	·센서를 수정합니다. ·센서 설정값은 센서의 최소값/최대값을 설정하기 위한 파라미터입니다.

JAVA OK HTTP 예시

```
OkHttpClient client = new OkHttpClient();

MediaType mediaType = MediaType.parse("application/x-www-form-urlencoded");
RequestBody body
= RequestBody.create(mediaType, "sensorName=SENSOR_NAME&content=SENSOR_SETTING_VALUE");
Request request = new Request.Builder()
  .url("http://13.209.241.209/api/device/{imei}/{device}/{seq}/update")
  .post(body)
  .addHeader("Content-Type", "application/x-www-form-urlencoded")
  .addHeader("Authorization", "Bearer YOUR_COMPANY_TOKEN ")
.build();

Response response = client.newCall(request).execute();
```

결과 예시

```
{
    "data": [1],
    "result": "success"
}
```

3.10 센서 상태 변경

센서의 상태를 변경합니다.

URL - http://13.209.241.209/api/device/{imei}/{device}/{seq}/use

전송 방식 - POST (application/x-www-form-urlencoded)

파라미터	파라미터 값	설명
Imei Device seq status	회원 IMEI 디바이스 고유 번호 센서seq 센서 상태	·센서의 상태를 수정합니다. ·기본적으로 Mantaray 서비스에서는 'P' : 사용중 , 'S' : 정지, 'D' : 사용안함 이렇게 세 가지 파라미터를 사용합니다.

JAVA OK HTTP 예시

```
OkHttpClient client = new OkHttpClient();

MediaType mediaType = MediaType.parse("application/x-www-form-urlencoded");
RequestBody body = RequestBody.create(mediaType, "status=STATUS");
Request request = new Request.Builder()
  .url("http://13.209.241.209/api/device/{imei}/{device}/{seq}/use")
  .post(body)
  .addHeader("Content-Type", "application/x-www-form-urlencoded")
  .addHeader("Authorization", "Bearer YOUR_COMPANY_TOKEN ")
.build();

Response response = client.newCall(request).execute();
```

결과 예시

```
{
    "data": [1],
    "result": "success"
}
```

3.11 토큰 등록

토큰을 등록시켜디바이스를 활성화합니다.

URL - http://13.209.241.209/api/device

전송 방식 - POST (application/x-www-form-urlencoded)

파라미터	파라미터 값	설명
Token(header)	디바이스 토큰	·토큰을 등록시켜 디바이스를 활성화합니다. ·등록할 토큰을 헤더에 넣어서 호출합니다.

JAVA OK HTTP 예시

```
OkHttpClient client = new OkHttpClient();

Request request = new Request.Builder()
  .url("http://13.209.241.209/api/device")
  .post(null)
  .addHeader("Content-Type", "application/x-www-form-urlencoded")
  .addHeader("token", DEVICE_TOKEN)
  .build();

Response response = client.newCall(request).execute();
```

결과 예시

```
{
    "data": [1],
    "result": "success"
}
```

4. 로그 관리

4.1 로그 전체 리스트

로그가 존재하는 모든 디바이스의 목록을 가져옵니다.

"LOG_CONTENT"는 가장 마지막에 업데이트 된 센서값입니다.

URL - http://13.209.241.209/api/log/device/{imei}/list

전송 방식 - GET

파라미터	파라미터 값	설명
imei	회원 imei	·로그가 존재하는 모든 디바이스의 목록을 가져옵니다. ·디바이스 유형에 관계없이 모두 출력됩니다

JAVA OK HTTP 예시

```
OkHttpClient client = new OkHttpClient();


Request request = new Request.Builder()
  .url("http://13.209.241.209/api/log/device/{imei}/list")
  .get()
  .addHeader("Content-Type", "application/x-www-form-urlencoded")
  .addHeader("Authorization", "Bearer YOUR_COMPANY_TOKEN ")
  .build();


Response response = client.newCall(request).execute();
```

결과 예시


```
{
    "data": [
        {
            "DEVICE_LOG_IDX": 1000,
            "DEVICE_INFO_IDX": 1,
            "SENSOR_INFO_IDX": 2,
            "LOG_CONTENT": "temp:-25.15,ph:5.37",
            "REG_DE": "2019-11-08T16:16:58.000Z",
            "MBER_IDX": 1,
            "DEVICE_INFO_NAME": "device1",
            "SENSOR_INFO_NAME": "sensor1",
            "SENSOR_INFO_STATUS": "D",
            "SENSOR_INFO_SEQ": 0
        },
        {
            "DEVICE_LOG_IDX": 1100,
            "DEVICE_INFO_IDX": 100,
            "SENSOR_INFO_IDX": 2,
            "LOG_CONTENT": "temp:27.15,ph:3.13",
            "REG_DE": "2019-11-08T17:10:45.000Z",
            "MBER_IDX": 2,
```

```
                "DEVICE_INFO_NAME": "device2",
                "SENSOR_INFO_NAME": "sensor2",
                "SENSOR_INFO_STATUS": "D",
                "SENSOR_INFO_SEQ": 0
            }
...
        ],
        "result": "success"
}
```


4.2 로그 디바이스 타입 리스트

지정한 타입에 맞으면서 로그가 있는 센서들의 목록을 가져옵니다.

URL - http://13.209.241.209/api/log/device/{imei}/{type}/list

전송 방식 - GET

파라미터	파라미터 값	설명
Imei type	회원 imei 디바이스 유형	·로그가 존재하는 디바이스의 목록을 가져옵니다. ·파라미터에 사용된 유형에 따라 디바이스 목록이 출력됩니다.

JAVA OK HTTP 예시

```
OkHttpClient client = new OkHttpClient();

Request request = new Request.Builder()
  .url("http://13.209.241.209/api/log/device/{imei}/[type]/list")
  .get()
  .addHeader("Authorization", "Bearer YOUR_COMPANY_TOKEN ")
.build();

Response response = client.newCall(request).execute();
```

결과 예시


```
{
    "data": [
        {
            "DEVICE_LOG_IDX":1101,
            "DEVICE_INFO_IDX": 2,
            "SENSOR_INFO_IDX": 10,
            "LOG_CONTENT": "temp:13,humi:10",
            "REG_DE": "2019-10-14T19:17:08.000Z",
            "MBER_IDX": 1
            "DEVICE_INFO_NAME": "device3",
            "DEVICE_INFO_SENSOR_TYPE": 2,
            "SENSOR_INFO_NAME": "sensor3",
            "SENSOR_INFO_STATUS": "D",
            "SENSOR_INFO_SEQ": 0
        },
        {
            "DEVICE_LOG_IDX": 2000,
            "DEVICE_INFO_IDX": 3,
            "SENSOR_INFO_IDX": 20,
            "LOG_CONTENT": "temp:24,humi:22",
            "REG_DE": "2019-10-10T19:42:45.000Z",
            "MBER_IDX": 1,
            "DEVICE_INFO_NAME": "device4",
            "DEVICE_INFO_SENSOR_TYPE": 2,
            "SENSOR_INFO_NAME": "sensor5",
            "SENSOR_INFO_STATUS": "P",
            "SENSOR_INFO_SEQ": 0
        }
...
    ],
    "result": "success"
}
```

4.3 로그 센서 전체 리스트

단일 디바이스의 센서 중 로그가 존재하는 센서들의 목록을 가져옵니다.

URL - http://13.209.241.209/api/log/device/{imei}/{type}/{device}/list

전송 방식 - GET

파라미터	파라미터 값	설명
Imei Type device	회원 imei 디바이스 유형 디바이스 고유 번호	.디바이스 고유 번호에 해당하는 디바이스의 센서 중 로그가 존재하는 센서들의 목록을 가져옵니다.

JAVA OK HTTP 예시

```
OkHttpClient client = new OkHttpClient();

MediaType mediaType = MediaType.parse("application/x-www-form-urlencoded");
Request request = new Request.Builder()
  .url("http://13.209.241.209/api/log/device/{imei}/{type}/{device}/list")
  .get()
  .addHeader("Authorization", "Bearer YOUR_COMPANY_TOKEN ")
.build();

Response response = client.newCall(request).execute();
```

결과 예시

```
{
    "data": [
        {
            "DEVICE_LOG_IDX": 2010,
            "DEVICE_INFO_IDX": 101,
            "SENSOR_INFO_IDX": 3,
            "LOG_CONTENT": "temp:13,humi:10",
```

```
            "REG_DE": "2019-10-10T19:17:08.000Z",
            "MBER_IDX": 1,
            "DEVICE_INFO_NAME": "device1",
            "DEVICE_INFO_SENSOR_TYPE": 2,
            "SENSOR_INFO_NAME": "sensor1",
            "SENSOR_INFO_STATUS": "D",
            "SENSOR_INFO_SEQ": 0
        }    ...],
    "result": "success"
}
```


4.4 로그 센서 단일 리스트

단일 센서의 날짜에 따른 로그를 가져옵니다.
날짜 형식은 YYYY-MM-DD의 형식으로 파라미터를 입력합니다.

URL - http://13.209.241.209/api/log/device/{imei}/{type}/{device}/{seq}/list
전송 방식 - POST (application/x-www-form-urlencoded)

파라미터	파라미터 값	설명
Imei Type Device Seq Date	회원 imei 디바이스 유형 디바이스 고유 번호 센서SEQ 날짜	·단일 센서의 날짜에 따른 전체 로그를 가져옵니다.

JAVA OK HTTP 예시

```
OkHttpClient client = new OkHttpClient();

MediaType mediaType = MediaType.parse("application/x-www-form-urlencoded");
```

```
RequestBody body = RequestBody.create(mediaType, "date=2019-11-11");
Request request = new Request.Builder()
  .url("http://13.209.241.209/api/log/device/{imei}/{type}/{device}/{seq}/list")
  .post(body)
  .addHeader("Content-Type", "application/x-www-form-urlencoded")
  .addHeader("Authorization", "Bearer YOUR_COMPANY_TOKEN ")
.build();

Response response = client.newCall(request).execute();
```

결과 예시


```
{
    "data": [
        {
            "DEVICE_LOG_IDX": 9487,
            "DEVICE_INFO_IDX": 101,
            "SENSOR_INFO_IDX": 72,
            "LOG_CONTENT": "temp:13,humi:10",
            "REG_DE": "2019-11-14T19:17:08.000Z",
            "MBER_IDX": 38,
            "DEVICE_INFO_NAME": "tfugg",
            "DEVICE_INFO_SENSOR_TYPE": 2,
            "SENSOR_INFO_NAME": "duufhej",
            "SENSOR_INFO_STATUS": "D",
            "SENSOR_INFO_SEQ": 0
        }    ....],
    "result": "success"
}
```

4.5 이벤트 전체 리스트

회원에게 귀속된 센서들의 모든 이벤트 로그를 가져옵니다.

URL - http://13.209.241.209/api/log/event/{imei}/list

전송 방식 - GET

파라미터	파라미터 값	설명
Imei	회원 imei	·회원에게 귀속된 센서들의 모든 이벤트 로그를 가져옵니다.

JAVA OK HTTP 예시

```
OkHttpClient client = new OkHttpClient();

Request request = new Request.Builder()
   .url("http://13.209.241.209/api/log/event/{imei}/list")
    .get()
   .addHeader("Authorization", "Bearer YOUR_COMPANY_TOKEN ")
.build();

Response response = client.newCall(request).execute();
```

결과 예시


```
{
    "data": [
        {
            "EVENT_LOG_IDX": 20,
            "DEVICE_LOG_IDX": 800,
            "LOG_CONTENT": "온도:설정값 이하,ph:설정값 이하",
            "REG_DE": "2019-10-10T07:13:51.000Z",
            "MBER_IDX": 1,
            "DEVICE_INFO_IDX": 2,
```

```
            "SENSOR_INFO_IDX": 5,
            "SENSOR_INFO_CONTENT": "temp:10/50,ph:20/40",
            "SENSOR_INFO_SEQ": 0,
            "MBER_IMEI": "1234567891011",
            "SENSOR_TYPE": "ph"
        },
        {
            "EVENT_LOG_IDX": 21,
            "DEVICE_LOG_IDX": 801,
            "LOG_CONTENT": "온도:설정값  이하,ph:설정값  이하",
            "REG_DE": "2019-10-10T07:14:02.000Z",
            "MBER_IDX": 1,
            "DEVICE_INFO_IDX": 3,
            "SENSOR_INFO_IDX": 10,
            "SENSOR_INFO_CONTENT": "temp:10/50,ph:10/40",
            "SENSOR_INFO_SEQ": 0,
            "MBER_IMEI": "1234567891011",
            "SENSOR_TYPE": "ph"
        }
],
"result": "success"
}
```


4.6 이벤트 타입 리스트

회원에게 귀속된 센서들의 로그중 디바이스 유형에 맞는 이벤트 로그를 전부 가져옵니다.

URL - http://13.209.241.209/api/log/event/{imei}/{type}/list

전송 방식 - GET

파라미터	파라미터 값	설명
Imei type	회원 imei 디바이스 유형	·회원에게 귀속된 센서 중 디바이스 유형에 맞는 이벤트 로그를 가져옵니다.

JAVA OK HTTP 예시

```
OkHttpClient client = new OkHttpClient();


Request request = new Request.Builder()
   .url("http://13.209.241.209/api/log/event/{imei}/{type}/list")
    .get()
   .addHeader("Authorization", "Bearer YOUR_COMPANY_TOKEN ")
.build();


Response response = client.newCall(request).execute();
```

결과 예시


```
{
    "data": [
        {
            "EVENT_LOG_IDX": 20,
            "DEVICE_LOG_IDX": 80,
            "LOG_CONTENT": "온도:설정값 이하,ph:설정값 이하",
            "REG_DE": "2019-11-08T07:13:51.000Z",
            "MBER_IDX": 1,
            "DEVICE_INFO_IDX": 2,
            "SENSOR_INFO_IDX": 7,
            "SENSOR_INFO_CONTENT": "temp:5/50,ph:20/40",
            "SENSOR_INFO_SEQ": 0,
            "MBER_IMEI": "1234567891011",
            "SENSOR_TYPE": "ph"
        },
        {
            "EVENT_LOG_IDX": 21,
            "DEVICE_LOG_IDX": 809,
            "LOG_CONTENT": "온도:설정값 이하,ph:설정값 이하",
            "REG_DE": "2019-10-08T07:14:02.000Z",
            "MBER_IDX": 1,
```

```
            "DEVICE_INFO_IDX": 3,
            "SENSOR_INFO_IDX": 10,
            "SENSOR_INFO_CONTENT": "temp:5/50,ph:20/40",
            "SENSOR_INFO_SEQ": 0,
            "MBER_IMEI": "1234567891011",
            "SENSOR_TYPE": "ph"
        }
...
],
"result": "success"
}
```


4.7 이벤트 센서 전체 리스트

단일 디바이스의 이벤트 로그를 전부 가져옵니다.

URL - http://13.209.241.209/api/log/event/{imei}/{type}/{device}/list

전송 방식 - GET

파라미터	파라미터 값	설명
Imei Type device	회원 imei 디바이스 유형 디바이스 고유 번호	·단일 디바이스의 전체 이벤트 로그를 가져옵니다.

JAVA OK HTTP 예시

```
OkHttpClient client = new OkHttpClient();

Request request = new Request.Builder()
  .url("http://13.209.241.209/api/log/event/{imei}/{type}/{device}/list")
  .get()
  .addHeader("Authorization", "Bearer YOUR_COMPANY_TOKEN ")
.build();
```

```
Response response = client.newCall(request).execute();
```

결과 예시


```
{
    "data": [
        {
            "EVENT_LOG_IDX": 20,
            "DEVICE_LOG_IDX": 80,
            "LOG_CONTENT": "온도:설정값 이하,ph:설정값 이하",
            "REG_DE": "2019-10-10T07:13:51.000Z",
            "MBER_IDX": 1,
            "DEVICE_INFO_IDX": 2,
            "SENSOR_INFO_IDX": 3,
            "SENSOR_INFO_CONTENT": "temp:5/50,ph:20/40",
            "SENSOR_INFO_SEQ": 0,
            "MBER_IMEI": "1234567891011",
            "SENSOR_TYPE": "ph"
        },
        {
            "EVENT_LOG_IDX": 25,
            "DEVICE_LOG_IDX": 81,
            "LOG_CONTENT": "온도:설정값 이하,ph:설정값 이하",
            "REG_DE": "2019-10-10T07:14:02.000Z",
            "MBER_IDX": 1,
            "DEVICE_INFO_IDX": 2,
            "SENSOR_INFO_IDX": 3,
            "SENSOR_INFO_CONTENT": "temp:5/50,ph:20/40",
            "SENSOR_INFO_SEQ": 0,
            "MBER_IMEI": "1234567891011",
            "SENSOR_TYPE": "ph"
        }
...
],
"result": "success"
}
```

4.8 이벤트 센서 단일 리스트

단일 센서의 날짜별 이벤트 로그를 가져옵니다.

URL - http://13.209.241.209/api/log/event/{imei}/{type}/{device}/{seq}/list

전송 방식 - POST (application/x-www-form-urlencoded)

파라미터	파라미터 값	설명
Imei Type Device Seq date	회원 imei 디바이스 유형 디바이스 고유 번호 센서SEQ 날짜	·단일 센서의 날짜별 이벤트 로그를 가져옵니다.

JAVA OK HTTP 예시

```
OkHttpClient client = new OkHttpClient();
RequestBody body = RequestBody.create(mediaType, "date=2019-11-11");
Request request = new Request.Builder()
  .url("http://13.209.241.209/api/log/event/{imei}/{type}/{device}/{seq}/list")
  .post(body)
  .addHeader("Authorization", "Bearer YOUR_COMPANY_TOKEN ")
.build();

Response response = client.newCall(request).execute();
```

결과 예시


```
{
    "data": [
        {
            "EVENT_LOG_IDX": 20,
            "DEVICE_LOG_IDX": 80,
            "LOG_CONTENT": "온도:설정값 이하,ph:설정값 이하",
```

```
            "REG_DE": "2019-11-08T07:13:51.000Z",
            "MBER_IDX": 1,
            "DEVICE_INFO_IDX": 2,
            "SENSOR_INFO_IDX": 3,
            "SENSOR_INFO_CONTENT": "temp:5/50,ph:20/40",
            "SENSOR_INFO_SEQ": 0,
            "MBER_IMEI": "1234567891011",
            "SENSOR_TYPE": "ph"
        },
        {
            "EVENT_LOG_IDX": 29,
            "DEVICE_LOG_IDX": 86,
            "LOG_CONTENT": "온도:설정값  이하,ph:설정값  이하",
            "REG_DE": "2019-11-08T07:14:02.000Z",
            "MBER_IDX": 1,
            "DEVICE_INFO_IDX": 2,
            "SENSOR_INFO_IDX": 3,
            "SENSOR_INFO_CONTENT": "temp:5/50,ph:20/40",
            "SENSOR_INFO_SEQ": 0,
            "MBER_IMEI": "1234567891011",
            "SENSOR_TYPE": "ph"
        }
...
],
"result": "success"
}
```


4.9 이벤트 센서 단일 전체 리스트

단일 센서의 이벤트 로그를 전부 가져옵니다.

URL - http://13.209.241.209/api/log/event/{imei}/{type}/{device}/{seq}/alllist

전송 방식 - GET

파라미터	파라미터 값	설명
Imei Type Device Seq	회원 imei 디바이스 유형 디바이스 고유 번호 센서SEQ	.단일 센서의 이벤트 로그를 전부 가져옵니다.

JAVA OK HTTP 예시

```
OkHttpClient client = new OkHttpClient();


Request request = new Request.Builder()
  .url("http://13.209.241.209/api/log/event/{imei}/{type}/{device}/{seq}/alllist")
    .get()
  .addHeader("Authorization", "Bearer YOUR_COMPANY_TOKEN ")
.build();


Response response = client.newCall(request).execute();
```

결과 예시


```
{
    "data": [
        {
            "EVENT_LOG_IDX": 20,
            "DEVICE_LOG_IDX": 808,
            "LOG_CONTENT": "온도:설정값 이하,ph:설정값 이하",
            "REG_DE": "2019-11-08T07:13:51.000Z",
            "MBER_IDX": 1,
            "DEVICE_INFO_IDX": 2,
            "SENSOR_INFO_IDX": 3,
            "SENSOR_INFO_CONTENT": "temp:5/50,ph:20/40",
            "SENSOR_INFO_SEQ": 0,
            "MBER_IMEI": "1234567891011",
            "SENSOR_TYPE": "ph"
```

```
        },
        {
            "EVENT_LOG_IDX": 29,
            "DEVICE_LOG_IDX": 809,
            "LOG_CONTENT": "온도:설정값 이하,ph:설정값 이하",
            "REG_DE": "2019-11-08T07:14:02.000Z",
            "MBER_IDX": 1,
            "DEVICE_INFO_IDX": 2,
            "SENSOR_INFO_IDX": 3,
            "SENSOR_INFO_CONTENT": "temp:5/50,ph:20/40",
            "SENSOR_INFO_SEQ": 0,
            "MBER_IMEI": "1234567891011",
            "SENSOR_TYPE": "ph"
        }
...
],
"result": "success"
}
```


4.10 로그 입력

센서 로그를 생성합니다

Data는 Json형태로 생성하여 호출해줘야합니다.

URL - http://13.209.241.209/api/log/insert

전송 방식 - POST (application/x-www-form-urlencoded)

파라미터	파라미터 값	설명
Json 형식 Device Token Valuse	디바이스 토큰 센서값	·로그를 입력합니다. Json 형식으로 전달하며 구성은 디바이스 토큰과 센서값입니다.

데이터 형식

```
{
    "device": DEVICE_TOKEN,
    "data": [
        {
            "seq": 0,
            "value": "temp:0,humi:0"
        },
        {
            "seq": 1,
            "value": "temp:2,humi:0"
        }
    ]
}
```

JAVA OK HTTP 예시

```
OkHttpClient client = new OkHttpClient();

MediaType mediaType = MediaType.parse("application/json");
RequestBody body = RequestBody.create(mediaType, DATA);
Request request = new Request.Builder()
  .url("http://13.209.241.209/api/log/insert")
  .post(body)
  .addHeader("Content-Type", "application/json")
.build();

Response response = client.newCall(request).execute();
```

결과 예시

```
{
    "data": [1],
    "result": "success"
}
```