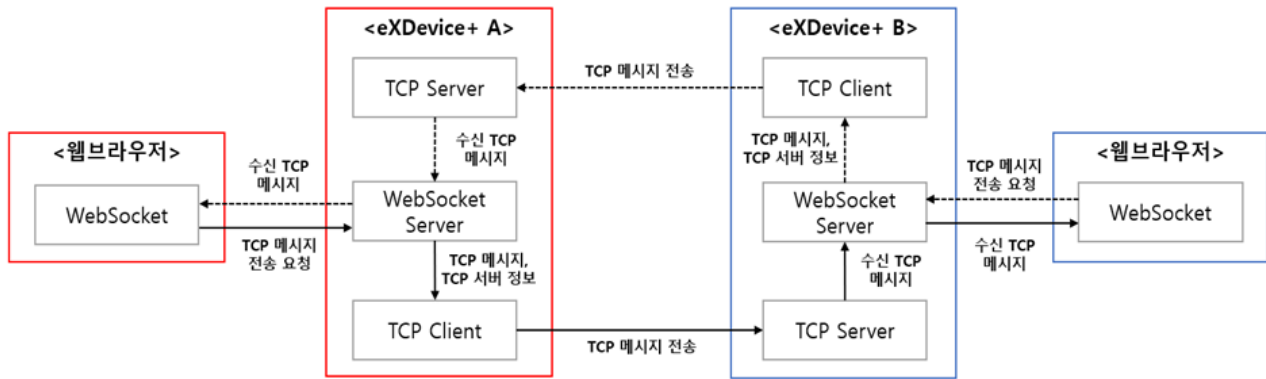


원격 eXDevice+간 TCP Socket 통신 서비스

eXDevice+는 다른 원격 PC에 설치되어 있는 eXDevice+에게 메시지를 송신하기 위한 서비스를 제공합니다. 이때, eXDevice+간 통신은 TCP Socket 통신을 통해 이루어집니다.



<eXDevice+간 TCP Socket 통신 구조>

eXDevice+간 TCP Socket 통신을 위해 다음과 같은 정보가 필요하며, 해당 정보에 대한 관리는 별도의 서버 또는 시스템에서 CRUD가 지원되어야 합니다.

데이터 종류	설명	용도	비고
IP	eXDevice+ TCP 서버 IP	TCP 메시지를 수신 받기 위한 서버 IP	로컬 IP
Port	eXDevice+ TCP 서버 port	TCP 메시지를 수신 받기 위한 서버 port	지정 port
endpoint	메시지를 수신할 client 정보	최종적으로 메시지를 수신 받기 위한 client 식별 정보	지정 endpoint 명

endpoint는 `OpenTcpServer` 서비스를 호출하며 등록하는 정보이며, 원격 WebSocket client가 어떤 WebSocket client에게 메시지를 전송할지 식별하기 위한 정보입니다. 다시 말해, **endpoint**는 일종의 사용자 ID와 유사한 역할을 수행합니다.

서비스 리스트

원격 eXDevice+간 TCP Socket 통신 서비스 리스트는 다음과 같습니다.

No.	서비스명	API명	비고
1	Socket 서버 open	OpenTcpServer	서버 open, endpoint 등록

No.	서비스명	API명	비고
2	Socket 서버 close	CloseTcpServer	서버 close, endpoint 제거
3	Socket 서버 정보 획득	GetTcpServerInfo	
4	endpoint 리스트 획득	GetTcpEndpoints	
5	TCP 메시지 송신	SendTcpMessage	ReceiveTcpMessage로 메시지 수신
6	TCP 메시지 수신	ReceiveTcpMessage	직접 호출 불가

서비스 상태 코드 리스트

본 서비스에서 공통으로 사용되는 서비스 상태코드 리스트에 대해 정의합니다.

이 상태 코드는 응답 JSON 메시지 중 *return* 또는 *returnValue*의 값에 해당합니다.

리턴코드	리턴 코드 내용	대상 서비스	비고
-1	정의되지 않은 예외	공통	Log 파일에서 내용 확인 필요
0	정상	공통	-
1	TCP 서버 open실패	OpenTcpServer	-
2	동일한 requestKey가 이미 등록됨	OpenTcpServer	-
3	동일한 endpoint가 이미 등록됨	OpenTcpServer	-
4	endpoint 등록 실패	OpenTcpServer	-
5	TCP 서버가 open 되어 있지 않음	CloseTcpServer, SendTcpMessage, GetTcpServerInfo, GetTcpEndpoint	-
6	endpoint가 등록되어 있지 않음	CloseTcpServer, GetTcpEndpoint	-
7	endpoint 제거 실패	CloseTcpServer	-
8	TCP 서버 종료 실패	CloseTcpServer	-
9	TCP 메시지 전송 실패	SendTcpMessage	-
10	원격 eXDevice+ TCP 서버 연결 실패	SendTcpMessage	-
11	원격 eXDevice+의 endpoint가 연결 해제됨	SendTcpMessage	-
12	원격 eXDevice+로 부터 TCP 수신 응답 획득 실패	SendTcpMessage	-
13	원격 eXDevice+가 TCP 메시지 처리 중 오류 발생	SendTcpMessage	-

리턴코드	리턴 코드 내용	대상 서비스	비고
14	원격 eXDevice+에 요청한 endpoint가 등록되어 있지 않음	SendTcpMessage	-
15	원격 eXDevice+가 endpoint에게 메시지 전송 중 연결 해제됨	SendTcpMessage	-
16	지정한 port 외 다른 port로 TCP 서버 open 됨	OpenTcpServer	-
17	지정한 port가 다른 프로그램에서 사용 중	OpenTcpServer	-

1. Socket 서버 open

API명

- OpenTcpServer

정의

- 원격 eXDevice+로부터 TCP 메시지를 수신하기 위해 TCP 서버를 open하고, 메시지를 수신할 WebSocket client의 식별 정보를 등록합니다.
- TCP 서버가 다른 WebSocket client에 의해 open 되어 있는 경우, 별도의 TCP 서버를 open 하지 않고 WebSocket client의 식별 정보를 추가 등록합니다.

설명

- WebSocket client는 자신에게 송신 된 메시지를 식별하기 위해 서비스 호출 시 전달한 *requestKey*를 사용하며, TCP 메시지는 WebSocket 이벤트 중 *onmessage* 이벤트를 통해 수신합니다.
- TCP 서버 open에 성공 한 경우 관리 서버 등에 IP, port, endpoint 정보를 등록해 사용할 수 있습니다.

✔ TCP 메시지를 수신 받는 WebSocket client의 프로토콜은 `ReceiveTcpMessage` 를 참고하시길 바랍니다.

⚠ eXDevice+는 TCP 서버 open 시 해당 PC의 IPv4 주소와, 입력한 port 넘버를 기준으로 서버를

open 합니다. 이때, eXDevice+는 하나의 TCP 서버만 open 할 수 있습니다.

호출 예시

```
{
  "service": "OpenTcpServer",
  "requestKey": "Random Request Key",
  "param": {
    "endpoint": "administrator"
  }
}
```

- requestKey
 - TCP 메시지 수신 시 TCP 메시지를 식별하기 위한 Key.
 - `CloseTcpServer`를 호출 하기 전 까지 관리되어야 하는 정보.
- endpoint
 - 원격 eXDevice+에서 특정 WebSocket client를 식별하기 위한 식별 정보.
- port
 - TCP 서버를 open 하기 위한 port 넘버.
 - IP는 해당 PC의 IPv4 주소.

응답 예시

```
{
  "service": "OpenTcpServer",
  "requestKey": "호출시 사용한 Request Key",
  "statusCode": "0000",
  "return": 0
}
```

- return
 - 서비스 상태 코드.

2. Socket 서버 close

API명

- CloseTcpServer

정의

- 등록된 endpoint를 제거하고 TCP 서버를 Close 합니다.
- 만약 서비스 호출 후 등록된 endpoint가 남아 있는 경우 요청한 endpoint만 제거하고 서버는 open 상태를 유지하며, 모든 endpoint가 제거되는 경우 eXDevice+는 TCP 서버를 Close합니다.

설명

- WebSocket client가 endpoint를 제거하지 않고 eXDevice+와 WebSocket 연결이 해제되는 경우, eXDevice+는 해당 WebSocket client가 등록한 모든 endpoint를 제거합니다.

⚠ endpoint는 자신이 등록한 endpoint만을 제거할 수 있습니다.

호출 예시

```
{
  "service": "CloseTcpServer",
  "requestKey": "Random Request Key",
  "param": {
    "endpoint": "제거할 endpoint"
  }
}
```

- endpoint
 - OpenTcpServer 로 등록한 endpoint.

응답 예시

```
{
  "service": "CloseTcpServer",
  "requestKey": "호출시 사용한 Request Key",
  "statusCode": "0000",
  "return": 0
}
```

- return
 - 서비스 상태 코드.

3. Socket 서버 정보 획득

API명

- GetTcpServerInfo

정의

- 현재 open된 TCP 서버의 IP, port 정보를 획득합니다.

호출 예시

```
{
  "service": "GetTcpServerInfo",
  "requestKey": "Random Request Key",
  "param": {}
}
```

응답 예시

```
{
  "service": "GetTcpServerInfo",
  "requestKey": "호출시 사용한 Request Key",
  "statusCode": "0000",
  "return": {
    "returnValue": 0,
    "ip": "123.123.123.123",
  }
}
```

```
"port":13441
}
```

- returnValue
 - 서비스 상태 코드.
- ip
 - open된 TCP 서버 IP.
- port
 - open된 TCP 서버 port.

4. endpoint 리스트 획득

API명

- GetTcpEndpoints

정의

- 서비스를 호출한 WebSocket client가 등록한 모든 endpoint 리스트를 획득합니다.

⚠ 다른 WebSocket client가 등록한 endpoint의 정보는 조회할 수 없습니다.

호출 예시

```
{
  "service": "GetTcpEndpoints",
  "requestKey": "Random Request Key...",
  "param": {}
}
```

응답 예시

```
{
  "service": "GetTcpEndpoints",
  "requestKey": "호출시 사용한 Request Key",
  "statusCode": "0000",
  "return": {
    "returnValue": 0,
    "endpointInfo": [
      {"requestKey": "9e1f-7026-863f", "endpoint": "endpoint1"},
      {"requestKey": "797a-250c-7c94", "endpoint": "endpoint2"}
    ]
  }
}
```

- returnValue
 - 서비스 상태 코드.
- endpointInfo: WebSocket client가 등록한 모든 requestKey 및 endpoint 정보 배열.
 - requestKey
 - WebSocket client가 TCP 메시지를 식별하기 위해 등록한 requestKey.
 - endpoint
 - WebSocket client가 원격 eXDevice+가 자신을 식별할 수 있도록 등록한 식별 정보.

5. TCP 메시지 송신

API명

- SendTcpMessage

정의

- 원격 eXDevice+에게 TCP 메시지를 송신합니다.

설명

- TCP 메시지의 최종 종착지는 원격 eXDevice+에 endpoint를 등록한 WebSocket client입니다.
- 본 서비스는 송신하는 메시지의 형태에 따라 다음과 같이 구분됩니다.

송신 타입	설명	비고
TCP 메시지 송신	TCP 메시지를 송신하되, WebSocket client의 응답을 받지 않습니다.	
응답이 필요한 TCP 메시지 송신	TCP 메시지를 송신하고, WebSocket client의 응답을 받습니다.	

i 응답 필요 유무는 서비스 호출 시 송신자의 endpoint 정보 포함 유무에 따라 나뉩니다.

호출 예시

1. TCP 메시지 송신

응답이 필요하지 않은 TCP 메시지를 송신할 때, 송신자는 수신자가 자신을 식별할 수 있는 이름(*name*) 정보만 기입합니다.

```
{
  "service": "SendTcpMessage",
  "requestKey": "Random Request Key",
  "param": {
    "sender": {
      "name": "수신자가 식별할 수 있는 이름"
    },
    "receiver": {
      "ip": "123.123.123.123",
      "port": 13441,
      "endpoint": "수신자가 eXDevice+에 등록한 endpoint"
    },
    "message": "송신하고자 하는 메시지"
  }
}
```

- sender: TCP 메시지 송신자 정보.
 - name
 - 수신자가 식별할 수 있는 이름.

- receiver: TCP 메시지 수신자 정보.
 - ip
 - 원격 eXDevice+ TCP 서버 IP.
 - port
 - 원격 eXDevice+ TCP 서버 port.
 - endpoint
 - 메시지를 수신한 원격 eXDevice+가 WebSocket client에게 TCP 메시지를 전달하기 위한 데이터.
 - 수신자가 등록한 endpoint 정보.
- message
 - 송신할 메시지.

2. 응답이 필요한 TCP 메시지 송신

응답이 필요한 TCP 메시지를 송신할 때 송신자는 `OpenTcpServer` 를 통해 응답 메시지를 수신 받을 TCP 서버를 반드시 open 해야 합니다. 송신자는 TCP 메시지 송신 시 수신자가 송신자인 WebSocket client를 식별할 수 있는 endpoint 정보를 함께 기입합니다.

```
{
  "service": "GetTcpEndpoints",
  "requestKey": "Random Request Key",
  "param": {
    "sender": {
      "name": "수신자가 식별할 수 있는 이름",
      "endpoint": "응답 메시지를 수신 받기 위해 등록한 endpoint"
    },
    "receiver": {
      "ip": "123.123.123.123",
      "port": 13441,
      "endpoint": "수신자가 eXDevice+에 등록한 endpoint"
    },
    "message": "송신하고자 하는 메시지"
  }
}
```

- sender: TCP 메시지 송신자 정보.
 - name
 - 수신자가 식별할 수 있는 이름
 - endpoint
 - 원격 eXDevice+에게 TCP 메시지를 보낼 때 최종 종착지가 될 수신자(WebSocket client) 식별 정보.
- receiver: TCP 메시지를 수신할 수신자 정보.
 - ip
 - 원격 eXDevice+ TCP 서버 IP.
 - port
 - 원격 eXDevice+ TCP 서버 port.
 - endpoint
 - 메시지를 수신한 원격 eXDevice+가 WebSocket client에게 TCP 메시지를 전달하기 위한 데이터.
 - 수신자가 등록한 endpoint 정보.
 - message
 - 송신할 메시지.

응답 예시

```
{
  "service": "SendTcpMessage",
  "requestKey": "호출시 사용한 Request Key",
  "statusCode": "0000",
  "return": 0
}
```

- return

- 서비스 상태 코드.

6. TCP 메시지 수신

API명

- SendTcpMessage

정의

- 다른 원격 eXDevice+가 송신한 TCP 메시지를 수신합니다.

설명

- `SendTcpMessage` 로 다른 원격 eXDevice+가 자신에게 송신한 메시지를 본 서비스를 통해 수신 받습니다.
- 본 서비스로 TCP 메시지를 수신 받으려면 WebSocket 이벤트 중 *onmessage* 이벤트를 통해 수신 합니다.

⚠ 본 서비스는 직접 호출할 수 없습니다.

응답 예시

1. TCP 메시지 수신

일반 TCP 메시지의 경우 송신자의 최소 정보(송신자 식별 정보)만이 메시지에 포함됩니다.

```
{
  "service": "ReceiveTcpMessage",
  "requestKey": "OpenTcpServer 호출 시 전달한 requestKey",
  "sender": {
    "name": "송신자 식별 정보"
  },
}
```

```
"message": "TCP 메시지"
}
```

- requestKey
 - TCP 메시지를 수신한 수신자가 메시지를 수신 받기 위해 *OpenTcpServer* 서비스 호출 시 사용한 *requestKey*.
- sender: TCP 메시지를 송신한 송신자 정보.
 - name
 - TCP 메시지를 송신한 송신자 식별 정보.
- message
 - 송신자가 전달한 실제 메시지.

2. 응답이 필요한 TCP 메시지 수신

응답이 필요한 TCP 메시지는 송신자의 eXDevice+ TCP 서버 정보 및, 메시지를 수신 받기 위한 WebSocket client의 식별 정보가 함께 전달됩니다.

```
{
  "service": "ReceiveTcpMessage",
  "requestKey": "OpenTcpServer 호출 시 전달한 requestKey",
  "sender": {
    "ip": "123.123.123.123",
    "port": 13441,
    "endpoint": "WebSocket Client 식별 정보"
    "name": "송신자 식별 정보"
  },
  "message": "TCP 메시지"
}
```

- requestKey
 - TCP 메시지를 수신한 수신자가 메시지를 수신 받기 위해 *OpenTcpServer* 서비스 호출 시 사용한 *requestKey*.
- sender: TCP 메시지를 송신한 송신자 정보.
 - ip

- TCP 메시지를 송신한 송신자의 원격 eXDevice+ TCP Socket Server IP.
- port
 - TCP 메시지를 송신한 송신자의 원격 eXDevice+ TCP Socket Server port.
- endpoint
 - TCP 메시지를 송신한 송신자의 WebSocket client 식별 정보.
 - 즉, 송신자의 eXDevice+가 어떤 WebSocket client에게 TCP 메시지를 보내야 할지 식별할 수 있는 정보.
- name
 - TCP 메시지를 송신한 송신자 식별 정보.
- message
 - 송신자가 전달한 실제 메시지.