

DM 유저 상태 목록 조회

≡ 설명	DM 방의 유저 상태 목록 조회 API
≡ REST API	/api/community/dm/{dm_id}/{user_id}
⌵ 메소드	GET
☑ 설계 완료	✓

request

Header - traceId

response

```
{
  "resultCode": 200,
  "resultMessage": "OK",
  "resultData": {
    "1": {
      "userId": 1,
      "userName": "example1",
      "profile": "example.com/recamp/~~~",
      "state": "offline"
    },
    "2": {
      "userId": 2,
      "userName": "example1",
      "profile": "example.com/recamp/~~~",
      "state": "offline"
    }
  }
}
```

Exception

```
{  
  "exception": "example",  
  "code": 10000,  
  "message": "example"  
}
```

유저 상태 목록 조회

≡ 설명	해당 길드에 있는 유저들의 상태 정보 조회
≡ REST API	/api/community/guild/{guild_id}/{user_id}
Ⓣ 메소드	GET
☑ 설계 완료	✓

request

Header - traceId

response

```
{
  "resultCode": 200,
  "resultMessage": "OK",
  "resultData": {
    "guildStates": { //길드 On/Off 유저 상태 정보
                      //userId -> value
      "2": {
        "userId": 2,
        "userName": "example1",
        "profile": "example.com/recamp/~~~",
        "state": "offline"
      }
    },
    //channelId -> info
    "voiceChannelStates": {
      "1": [
        {
          "userReadId": 10000032,
          "guildId": 24,
          "userId": 2,
          "profile": "example.com/recamp/~~~",
          "nickname": "example1"
        }
      ]
    }
  }
}
```

```
    },  
    {...}  
  ]  
}  
}
```

Exception

```
{  
  "exception": "example",  
  "code": 10000,  
  "message": "example"  
}
```

유저 프로필 정보 변경

≡ 설명	유저의 프로필 정보 변경시 API
≡ REST API	/api/community/change/user
⌵ 메소드	PATCH
☑ 설계 완료	✓

request

```
Header - traceId
{
  "userId" : 1,
  "profile" : "example.com/example/~~~~"
}
```

response

```
"example"
```

Exception

```
{
  "exception": "example",
  "code": 10000,
  "message": "example"
}
```

유저 닉네임 정보 변경

≡ 설명	유저의 닉네임 정보 변경시 API
≡ REST API	/api/community/change/user/nickname
Ⓣ 메소드	PATCH
☑ 설계 완료	✓

request

```
Header - traceId
{
  "userId" : 1,
  "nickname" : "example"
}
```

response

```
"example"
```

Exception

```
{
  "exception": "example",
  "code": 10000,
  "message": "example"
}
```

유저 정보 저장

≡ 설명	회원가입 시 커뮤니티 서버로 유저 정보 저장 API
≡ REST API	/api/community/registration/user
⌵ 메소드	POST
☑ 설계 완료	✓

request

```
Header - traceId,  
body  
{  
  "userId" : 1,  
  "email" : "example@example.com",  
  "name" : "example",  
  "profile" : "example.com/recamp/~~~"  
}
```

response

```
{  
  "resultCode": 200,  
  "resultMessage": "OK"  
}
```

Exception

```
{  
  "exception": "example",  
  "code": 10000,  
}
```

```
"message": "example"  
}
```


길드 생성

≡ 설명	길드 생성 API
≡ REST API	/api/community/registration/guild
⌵ 메소드	POST
☑ 설계 완료	✓

request

```
Header - traceId,  
body  
{  
    "managerId" : 1,  
    "name" : "example",  
},  
form-data profile
```

response

```
{  
    "resultCode": 200,  
    "resultMessage": "OK"  
}
```

Exception

```
{  
    "exception": "example",  
    "code": 10000,  
    "message": "example"  
}
```


길드 초대링크 생성

≡ 설명	길드에 초대 링크를 생성 API
≡ REST API	/api/community/invitation/guild
⌵ 메소드	POST
☑ 설계 완료	✓

request

```
Header - traceId,  
body  
{  
  "userId" : 1,  
  "guildId" : 2,  
}
```

response

```
{  
  "resultCode": 200,  
  "resultMessage": "OK",  
  "resultData": "0c3eda9837aa4055911dfb87280e9c95.1.1"  
}
```

Exception

```
{  
  "exception": "example",  
  "code": 10000,  
  "message": "example"  
}
```


길드 가입

≡ 설명	길드에 초대 링크를 통해 가입하는 API
≡ REST API	/api/community/join/guild/{invitation_value}/{user_id}
⌵ 메소드	GET
☑ 설계 완료	✓

request

Header - traceId

response

```
{
  "resultCode": 200,
  "resultMessage": "OK"
}
```

Exception

```
{
  "exception": "example",
  "code": 10000,
  "message": "example"
}
```

길드 조회

≡ 설명	유저가 로그인했을 때 유저의 길드 목록 조회 API
≡ REST API	/api/community/check/guild/{user_id}
Ⓣ 메소드	GET
☑ 설계 완료	✓

request

Header - traceId

response

```
{
  "resultCode": 200,
  "resultMessage": "OK",
  "resultData": {
    "1": {
      "guildReadId": 1,
      "guildId": 1,
      "userId": 1,
      "name": "example",
      "profile": "https://storage.googleapis.com/rememb
    },
    "2": {
      "guildReadId": 2,
      "guildId": 2,
      "userId": 1,
      "name": "example",
      "profile": "https://storage.googleapis.com/rememb
    },
    "3": {
      "guildReadId": 3,
      "guildId": 3,
      "userId": 1,
```

```

        "name": "example",
        "profile": "https://storage.googleapis.com/rememb
    },
    "4": {
        "guildReadId": 4,
        "guildId": 4,
        "userId": 1,
        "name": "example",
        "profile": "https://storage.googleapis.com/rememb
    },
    "5": {
        "guildReadId": 5,
        "guildId": 5,
        "userId": 1,
        "name": "스위스",
        "profile": "https://storage.googleapis.com/rememb
    },
    "6": {
        "guildReadId": 6,
        "guildId": 6,
        "userId": 1,
        "name": "로그 서버",
        "profile": "https://storage.googleapis.com/rememb
    },
    "7": {
        "guildReadId": 7,
        "guildId": 7,
        "userId": 1,
        "name": "해시태그",
        "profile": "https://storage.googleapis.com/rememb
    },
    "8": {
        "guildReadId": 8,
        "guildId": 8,
        "userId": 1,
        "name": "비둘기",
        "profile": "https://storage.googleapis.com/rememb
    },

```

```

    "9": {
      "guildReadId": 9,
      "guildId": 9,
      "userId": 1,
      "name": "노트북",
      "profile": "https://storage.googleapis.com/rememb
    },
    "10": {
      "guildReadId": 10,
      "guildId": 10,
      "userId": 1,
      "name": "hashtag",
      "profile": "https://storage.googleapis.com/rememb
    }
  }
}

```

Exception

```

{
  "exception": "example",
  "code": 10000,
  "message": "example"
}

```


길드 삭제

≡ 설명	길드를 삭제하는 API
≡ REST API	/api/community/delete/guild
⌵ 메소드	DELETE
☑ 설계 완료	✓

request

```
Header - traceId
{
  "guildId" : 1,
  "managerId" : 1
}
```

response

```
{
  "resultCode": 200,
  "resultMessage": "OK"
}
```

Exception

```
{
  "exception": "example",
  "code": 10000,
  "message": "example"
}
```

길드 유저 이름 변경

≡ 설명	길드에서의 유저 이름 변경 API
≡ REST API	/api/community/change/guild/username
⌵ 메소드	PATCH
☑ 설계 완료	✓

request

```
Header - traceId
{
  "userId" : 1,
  "guildId" : 1,
  "nickname" : "호날두"
}
```

response

```
"example"
```

Exception

```
{
  "exception": "example",
  "code": 10000,
  "message": "example"
}
```

카테고리 생성

≡ 설명	길드에서 카테고리 생성하는 API
≡ REST API	/api/community/registration/category
⌵ 메소드	POST
☑ 설계 완료	✓

request

```
Header - traceId,  
body  
{  
  "name" : "example",  
  "userId" : 1,  
  "guildId" : 1  
}
```

response

```
{  
  "resultCode": 200,  
  "resultMessage": "OK"  
}
```

Exception

```
{  
  "exception": "example",  
  "code": 10000,  
  "message": "example"  
}
```


카테고리 삭제

≡ 설명	카테고리를 삭제하는 API
≡ REST API	/api/community/delete/category
⌵ 메소드	DELETE
☑ 설계 완료	✓

request

```
Header - traceId
{
  "guildId" : 1,
  "categoryId" : 1
  "userId" : 1
}
```

response

```
{
  "resultCode": 200,
  "resultMessage": "OK"
}
```

Exception

```
{
  "exception": "example",
  "code": 10000,
  "message": "example"
}
```

카테고리 조회

≡ 설명	카테고리를 조회하는 API
≡ REST API	/api/community/check/category/{guild_id}/{user_id}
⌵ 메소드	GET
☑ 설계 완료	✓

request

Header - traceId

response

```
{
  "resultCode": 200,
  "resultMessage": "OK",
  "resultData": [
    {
      "categoryReadId": 2,
      "guildId": 1,
      "name": "harmony"
    }
  ]
}
```

Exception

```
{
  "exception": "example",
  "code": 10000,
  "message": "example"
}
```

카테고리 수정

≡ 설명	카테고리 이름 수정하는 API
≡ REST API	/api/community/change/category
⌵ 메소드	PATCH
☑ 설계 완료	✓

request

```
Header - traceId
{
  "guildId":1,
  "userId":1,
  "categoryId":1,
  "name":"example"
}
```

response

```
{
  "resultCode": 200,
  "resultMessage": "OK"
}
```

Exception

```
{
  "exception": "example",
  "code": 10000,
  "message": "example"
}
```

채널 생성

≡ 설명	길드에서 카테고리 안의 채널을 생성하는 API
≡ REST API	/api/community/registration/category/channel
⌵ 메소드	POST
☑ 설계 완료	☑

request

```
Header - traceId,  
body  
{  
  "guildId" : 1,  
  "name" : "example",  
  "userId" : 1,  
  "categoryId" : 1,  
  "type" : "TEXT"  
}  
type : TEXT, VOICE, FORUM, ANNOUNCEMENT, STAGE
```

response

```
{  
  "resultCode": 200,  
  "resultMessage": "OK"  
}
```

Exception

```
{  
  "exception": "example",  
  "code": 10000,
```



```
"message": "example"  
}
```

채널 생성

≡ 설명	길드에서 길드 안의 채널을 생성하는 API
≡ REST API	/api/community/registration/guild/channel
⌵ 메소드	POST
☑ 설계 완료	✓

request

```
Header - traceId,  
body  
{  
  "name" : "example",  
  "userId" : 1,  
  "guildId" : 1,  
  "type" : "TEXT"  
}
```

response

```
{  
  "resultCode": 200,  
  "resultMessage": "OK"  
}
```

Exception

```
{  
  "exception": "example",  
  "code": 10000,  
}
```

```
"message": "example"  
}
```

채널 조회

≡ 설명	유저가 길드를 클릭했을 때 길드의 채널 목록 조회 API
≡ REST API	/api/community/check/guild/channel/{guild_id}/{user_id}
⌵ 메소드	GET
☑ 설계 완료	✓

request

Header - traceId

response

```
{
  "resultCode": 200,
  "resultMessage": "OK",
  "resultData": {
    "1": {
      "channelReadId": 1,
      "categoryId": 4,
      "guildId": 1,
      "name": "0Chord의 잡담교실",
      "type": "FORUM"
    },
    "2": {
      "channelReadId": 2,
      "categoryId": 1,
      "guildId": 1,
      "name": "0Chord의 잡담교실",
      "type": "FORUM"
    },
    "3": {
      "channelReadId": 3,
      "categoryId": 1,
      "guildId": 1,

```

```

        "name": "우리들의 하모니",
        "type": "VOICE"
    },
    "4": {
        "channelReadId": 4,
        "categoryId": 2,
        "guildId": 1,
        "name": "BTS 진",
        "type": "TEXT"
    },
    "5": {
        "channelReadId": 5,
        "categoryId": 1,
        "guildId": 1,
        "name": "최성민의 토크쇼",
        "type": "VOICE"
    }
}

```

Exception

```

{
    "exception": "example",
    "code": 10000,
    "message": "example"
}

```

채널 삭제

≡ 설명	채널을 삭제하는 API
≡ REST API	/api/community/delete/channel
Ⓣ 메소드	DELETE
☑ 설계 완료	✓

request

```
Header - traceId
{
  "channelId" : 1,
  "guildId" : 1
  "userId" : 1
}
```

response

```
{
  "resultCode": 200,
  "resultMessage": "OK"
}
```

Exception

```
{
  "exception": "example",
  "code": 10000,
  "message": "example"
}
```

게시글 생성

≡ 설명	게시글 작성 API
≡ REST API	/api/community/registration/board
⌵ 메소드	POST
☑ 설계 완료	✓

request

```
Header - traceId
{
  "userId" : 1,
  "guildId" : 1,
  "channelId" : 1,
  "title" : "example",
  "content" : "content"
},
image-list
```

response

```
{
  "resultCode": 200,
  "resultMessage": "OK"
}
```

Exception

```
{
  "exception": "example",
  "code": 10000,
  "message": "example"
}
```

게시글 목록 조회

≡ 설명	채널을 클릭 시 게시글 목록 조회 API
≡ REST API	/api/community/board/{guild_id}/{channel_id}/{cursor}
⌵ 메소드	GET
☑ 설계 완료	✓

request

Header - traceId

response

```
{
  "resultCode": 200,
  "resultMessage": "OK",
  "resultData": []
}
```

Exception

```
{
  "exception": "example",
  "code": 10000,
  "message": "example"
}
```


게시글 수정

≡ 설명	게시글 수정 API
≡ REST API	/api/community/change/board
⌵ 메소드	PATCH
☑ 설계 완료	✓

request

```
Header - traceId
{
  "userId" : 1,
  "boardId" : 1,
  "title" : "example",
  "content" : "content"
}
```

response

```
{
  "resultCode": 200,
  "resultMessage": "OK"
}
```

Exception

```
{
  "exception": "example",
  "code": 10000,
  "message": "example"
}
```

게시글 삭제

≡ 설명	게시글을 삭제하는 API
≡ REST API	/api/community/delete/board
⌵ 메소드	DELETE
☑ 설계 완료	✓

request

```
Header - traceId
{
  "boardId" : 1,
  "userId" : 1
}
```

response

```
{
  "resultCode": 200,
  "resultMessage": "OK"
}
```

Exception

```
{
  "exception": "example",
  "code": 10000,
  "message": "example"
}
```

게시글 조회

≡ 설명	게시글 클릭 시 게시글 조회 API
≡ REST API	/api/community/board/{boardId}
⌵ 메소드	GET
☑ 설계 완료	✓

request

Header - traceId

response

```
{
  "resultCode": 200,
  "resultMessage": "OK",
  "resultData": {
    "boardId": 1,
    "title": "첫번째 게시글 수정테스트입니다",
    "content": "첫번째 게시글 수정테스트입니다",
    "writerName": "example1",
    "userId": 1,
    "modified": true,
    "createdAt": "2024-01-19T17:32:31.723781",
    "commentsResponseDto": {
      "commentResponseDto": [
        {
          "commentId": 1,
          "boardId": 1,
          "writerName": "0Chord",
          "comment": "첫번째 댓글 수정입니다",
          "userId": 1,
          "modified": true,
          "writerProfile": "example.com/~~~",
          "createdAt": "2024-01-19T17:32:58.755235"
```

```

        }
    ]
},
"imagesResponseDto": {
    "image": [
        {
            "image": "https://storage.googleapis.com/"
        },
        {
            "image": "https://storage.googleapis.com/"
        },
        {
            "image": "https://storage.googleapis.com/"
        }
    ]
},
"emojisResponseDto": {
    "emojiResponseDtos": [
        {
            "emojiId": 1,
            "boardId": 1,
            "commentId": null,
            "emojiType": 1,
            "emojiUsers": [
                1,
                2
            ]
        }
    ]
}
}

```

Exception

```

{
    "exception": "example",
    "code": 10000,

```

```
"message": "example"  
}
```

댓글 작성

≡ 설명	게시글 댓글 작성 API
≡ REST API	/api/community/registration/board/comment
⌵ 메소드	POST
☑ 설계 완료	✓

request

```
Header - traceId
{
  "userId" : 1,
  "boardId" : 1,
  "comment" : "content",
  "writerName" : "exmaple",
  "writerProfile" : "example.com/~~~"
}
```

response

```
{
  "resultCode": 200,
  "resultMessage": "OK"
}
```

Exception

```
{
  "exception": "example",
  "code": 10000,
  "message": "example"
}
```

댓글 목록 조회

≡ 설명	댓글 목록 조회 API → 게시글 조회시 같이 출력되게
≡ REST API	/api/community/comment/{channel_id}/{board_id}/{user_id}
Ⓣ 메소드	GET
☑ 설계 완료	<input type="checkbox"/>

request

Header - traceId

response

"example"

Exception

```
{
  "exception": "example",
  "code": 10000,
  "message": "example"
}
```

댓글 수정

≡ 설명	댓글 수정 API
≡ REST API	/api/community/change/comment
⌵ 메소드	PATCH
☑ 설계 완료	✓

request

```
Header - traceId
{
  "userId" : 1,
  "commentId" : 1,
  "comment" : "첫번째 댓글 수정입니다"
}
```

response

```
{
  "resultCode": 200,
  "resultMessage": "OK"
}
```

Exception

```
{
  "exception": "example",
  "code": 10000,
  "message": "example"
}
```


댓글 삭제

≡ 설명	댓글을 삭제하는 API
≡ REST API	/api/community/delete/comment
⌵ 메소드	DELETE
☑ 설계 완료	✓

request

```
Header - traceId
{
  "commentId" : 1
  "userId" : 1
}
```

response

```
{
  "resultCode": 200,
  "resultMessage": "OK"
}
```

Exception

```
{
  "exception": "example",
  "code": 10000,
  "message": "example"
}
```

이모지 추가

≡ 설명	게시글 이모지 추가 API
≡ REST API	/api/community/registration/board/emoji
Ⓣ 메소드	POST
☑ 설계 완료	✓

request

```
Header - traceId
{
  "boardId" : 1,
  "emojiType" : 1,
  "userId" : 1
}
```

response

```
{
  "resultCode": 200,
  "resultMessage": "OK"
}
```

Exception

```
{
  "exception": "example",
  "code": 10000,
  "message": "example"
}
```

이모지 삭제

≡ 설명	게시글 이모지 삭제 API
≡ REST API	/api/community/delete/emoji
Ⓣ 메소드	DELETE
☑ 설계 완료	✓

request

```
Header - traceId
{
  "emojiId" : 1,
  "userId" : 1
}
```

response

```
{
  "resultCode": 200,
  "resultMessage": "OK"
}
```

Exception

```
{
  "exception": "example",
  "code": 10000,
  "message": "example"
}
```

룸 생성

≡ 설명	DM 룸 생성 API
≡ REST API	/api/community/registration/room
⌵ 메소드	POST
☑ 설계 완료	✓

request

```
Header - traceId
{
  "name" : "example",
  "members" : [1,2,3,4],
  "profile" : "example.com/~~"
}
```

response

```
{
  "resultCode": 200,
  "resultMessage": "OK"
}
```

Exception

```
{
  "exception": "example",
  "code": 10000,
  "message": "example"
}
```

룸 삭제

≡ 설명	DM 룸 삭제 API
≡ REST API	/api/community/delete/room
Ⓣ 메소드	DELETE
☑ 설계 완료	<input type="checkbox"/>

request

```
Header - traceId
{
  "firstUser" : 1,
  "secondUser" : 2
}
```

response

```
{
  "resultCode": 200,
  "resultMessage": "OK"
}
```

Exception

```
{
  "exception": "example",
  "code": 10000,
  "message": "example"
}
```

룸 조회

≡ 설명	DM 유저 룸 조회 API
≡ REST API	/api/community/check/room/{user_id}
⌵ 메소드	GET
☑ 설계 완료	✓

request

Header - traceId

response

```
{
  "resultCode": 200,
  "resultMessage": "OK",
  "resultData": {
    "roomResponseDtos": [
      {
        "roomId": 1,
        "name": "강력한 한방",
        "profile": "https://storage.googleapis.com/ex
      },
      {...}
    ]
  }
}
```

Exception

```
{
  "exception": "example",
  "code": 10000,
```

```
"message": "example"  
}
```

보안을 신경쓰자.. 그렇게 함부로 자리를 비우면 안되죠 0현님.

이번만 한번 봐드립니다. 방금 데이터 다 날라간거예요.

하지만 저는 봐주지 않죠. 날려버리겠습니다.

룸, 길드 한번에 조회

≡ 설명	룸 길드 PK 조회 API
≡ REST API	/api/community/check/room/guild/{user_id}
⌵ 메소드	GET
☑ 설계 완료	✓

request

Header - traceId

response

```
{
  "resultCode": 200,
  "resultMessage": "OK",
  "resultData": {
    "roomIds": [
      1,
      2
    ],
    "guildIds": [
      1,
      2,
      3,
      4
    ]
  }
}
```

Exception

```
{
  "exception": "example",
```



```
"code": 10000,  
"message": "example"  
}
```