


| 일반 소스 공유 | >

간단한 모듈형 명령어 verifier

 **odask** 챗봇 마스터 🤖 1:1 채팅
2020.03.26. 20:22 조회 122

 댓글 5

 URL 복사



공유할 소스 용도

=> (ex: 가위바위보 게임)

소스 코드

=>

```
//유저 권한 등급: 소스 맨 위에 넣으세요
const PERMISSION = ['admin', 'manager', 'moderator', 'member', 'guest']

function commandChk(localCommand, permission, print) {
  const pindex = PERMISSION.indexOf(permission)
  const ret = []
  function chk() {
    if (!this.enabled) return false
    if (pindex <= PERMISSION.indexOf(this.permission) ? true : false) return false
    return true
  }
  function vrf() {
    if (typeof this == 'function') return this()
  }
  function checker(lcmd) {
    this.forEach((e) => {
      lcmd[e.name] = lcmd[e.name] || e.setting
      if (e.cate == 'group') {
        print == 1 && ret.push('\n분류: ' + e.name + ' (' + e.setting.permission + '/' + (e.setting.enabled ? 'enabled' : 'disabled') + ')\n| ' + e.description)
        if (chk.call(lcmd[e.name])) {
          print == 2 && ret.push('\n분류: ' + e.name + ' (' + e.setting.permission + ')\n| ' + e.description)
          checker.call(e.main, lcmd[e.name]);
        }
      } else if (e.cate == 'command') {
        print == 1 && ret.push('    >' + e.name + ' (' + e.setting.permission + '/' + (e.setting.enabled ? 'enabled' : 'disabled') + ')\n    | ' + e.description)
        if (chk.call(lcmd[e.name]) && vrf.call(e.main.verify)) {
          print == 2 && ret.push('    >' + e.name + ' (' + e.setting.permission + ')\n    | ' + e.description)
          e.main.execute()
        }
      }
    })
  }
  checker.call(this, localCommand)
  print && replier.reply(ret.join('\n'))
}

//setting keys must not be used as group/command name!
const commandList = [{
  cate: 'group',
  name: 'testA',
  description: 'for test purpose',
  setting: {
    enabled: true,
    permission: 'guest'
  },
  main: [
```

```

main: [{
  cate: 'command',
  name: 'test1',
  description: 'this is a test command',
  setting: {
    enabled: true,
    permission: 'guest'
  },
  main: {
    verify: function () {
      //if (msg == '!테스트')
      return true
    },
    execute: function () {
      //replier.reply('asdf')
    }
  }
}], {
  cate: 'command',
  name: 'test2',
  description: 'this is a test command',
  setting: {
    enabled: true,
    permission: 'member'
  },
  main: {
    verify: function () {
      //if (msg == '!테스트')
      return true
    },
    execute: function () {
      //replier.reply('asdf')
    }
  }
}]
}, {
  cate: 'group',
  name: 'testB',
  description: 'for test purpose',
  setting: {
    enabled: true,
    permission: 'guest'
  },
  main: [{
    cate: 'command',
    name: 'test1',
    description: 'this is a test command',
    setting: {
      enabled: false,
      permission: 'guest'
    },
    main: {
      verify: function () {
        //if (msg == '!테스트')
        return true
      },
      execute: function () {
        //replier.reply('asdf')
      }
    }
  }
], {
  cate: 'command',
  name: 'test2',
  description: 'this is a test command',
  setting: {
    enabled: true,
    permission: 'guest'
  },
  main: {
    verify: function () {

```

```

        verify: function () {
            //if (msg == '!테스트')
            return true
        },
        execute: function () {
            //replier.reply('asdf')
        }
    }
}
}}

commandChk.call(commandList, localPermissionJSON /*객체*/ , localUserPermission/* ,print*/)

```

일단 초안이라 대충 만들었으니까 알아서 다듬어 쓰시면 될것 같습니다.

우선 준비물은

방 별 명령어 셋업을 담은 객체가 필요하고...

유저를 닉네임이던 해시코드가던 구분해서 해당 유저의 정보를 반환하는 함수가 필요합니다.

둘다 로컬에 저장이 되면 좋겠죠? 그래야 붓을 리로드해도 변경사항이 저장될테니까요.

```

commandChk.call(commandList, localPermissionJSON /*객체*/ , localUserPermission/* ,print*/)

```

이렇게 호출을 하는데,

localPermissionJSON 을 방별로 할당된 비어있는 객체(알아서 채워줍니다)변수로 바꾸시고

localUserPermission 을 유저별로 할당된 명령어 권한 변수로 바꾸시면 됩니다.

('admin', 'manager', 'moderator', 'member', 'guest' 중 하나 - 알아서 커스텀하셔도됨)

방/유저별로 DB를 연결해주는 함수는 알아서 만드셔야겠죠?

명령어 객체는 보시다시피 트리 구조를 가지고 있는데

group이 폴더같은 느낌이고 **command**가 파일같은 느낌입니다.

group 하위에는 **group** 또는 **command**가 있을 수 있어서 계속 확장이 가능합니다.

상위 **group**의 세팅을 변경하면 하위 **group** 또는 **command**에도 전체 적용됩니다.

command의 main 키의 하위에는 **verify**와 **execute**가 있는데,

verify에서 **truthy**한 값이 **return**될 경우, **execute**가 실행됩니다.

그러니까 **verify**로 명령어 체크를 하고 **execute**에 명령어 본문을 넣으시면 됩니다.

localPermissionJSON 객체는 초기 **init**시 **commandList** 객체의 기본값으로 설정됩니다.

해당 예제 코드에서는:

```

{
    "testA": {
        "enabled": true,
        "permission": "guest",
        "test1": {
            "enabled": true,
            "permission": "guest"
        },
        "test2": {
            "enabled": true,
            "permission": "member"
        }
    },
    "testB": {
        "enabled": true,
        "permission": "guest",
        "test1": {
            "enabled": false,
            "permission": "guest"
        },
        "test2": {
            "enabled": true,
            "permission": "guest"
        }
    }
}
}

```

localPermissionJSON 객체를 직접 수정하면, 기존의 default 세팅을 덮어씹습니다.

객체의 키나 객체 전체를 삭제하면 삭제한 부분은 기본값으로 복원됩니다.

방별 객체가 때문에 방별로 권한 또는 활성화 여부를 다르게 설정할 수 있다는 겁니다.

```
commandChk.call(commandList, localPermissionJSON /*객체*/ , localUserPermission/* ,print*/)

```

마지막 인자인 print는 넣지 않으시거나 1 또는 2를 넣으시면 되는데

넣지 않으시면 그냥 평소대로 처리하고

1을 넣을 경우 모든 명령어의 도움말을 출력하며

2를 넣을 경우 권한과 활성화 여부를 반영한 내가 사용 가능한 명령어의 도움말만 출력합니다.

해당 예제 코드에서는

만약 유저가 guest 권한이라면

1을 넣을 경우:

```
분류: testA (guest/enabled)
| for test purpose
  > test1 (guest/enabled)
    | this is a test command
  > test2 (member/enabled)
    | this is a test command
분류: testB (guest/enabled)
| for test purpose
  > test1 (guest/disabled)
    | this is a test command
  > test2 (guest/enabled)
    | this is a test command

```

2를 넣을 경우:

```
분류: testA (guest)
| for test purpose
  > test1 (guest)
    | this is a test command
분류: testB (guest)
| for test purpose
  > test2 (guest)
    | this is a test command

```

맘에 안드신다면 알아서 적절히 포맷해 사용하세요.



odosk님의 게시글 더보기 >

❤️ 좋아요 1 💬 댓글 5

🔗 공유 | 신고

댓글 등록순 최신순 🔁

댓글알림 ☐



무명

나! 갱장해

2020.03.26. 20:30 답글 쓰기



1급 청정수 MoonBot

써거따

2020.03.27. 00:57 답글 쓰기



페르조나

와, 간단하다...!! 근데 이걸로 뭘할 수 있나용? ㅋㅋㅋ

2020.03.27. 17:06 답글 쓰기



odosk 작성자

명령어 처리요

예를들어 님이 지금 뭐 관리자 권한 eval 명령어를 만든다면
관리자권한을 체크하고 메세지가 /eval로 시작하는지 확인하고 이런 과정을 거치겠죠

저건 그걸 다 모듈화/함수화해서
그냥 permission: 'admin' 하면 권한체크해주고
verifier 함수 안에 if(msg.startsWith('/eval') return true 적어주면 명령어 키워드 체크해주고

명령어 설명은 보통 일일이 손으로 다 적는경우가 대부분인데 그것도 자동화해서 따로 적어줄 필요가 없어지죠

더 개선된 버전도 만들었는데 혹시 진짜 쓸 의향이 있으시면 뭐 연락주시면 보내드리던가 하겠습니다

2020.03.27. 17:18 답글쓰기



페르조나

odosk 호오.. 멋집니다. 언젠가 필요하면 연락드릴게요~

2020.03.27. 17:50 답글쓰기

Hibot

댓글을 남겨보세요



등록

글쓰기

답글

목록

▲ TOP

'| 일반 소스 공유 |' 게시판 글

이 게시판 새글 구독하기 ☐

도움이 될거 같아 올리는 어드민 체크 함수 [4]	하프	2020.03.27.
Kal님 카링 사용한 포켓몬 RPG 추가 컨텐츠 (사냥하기) [25]	Charlottek	2020.03.27.
간단한 모듈형 명령어 verifier [5]	odosk	2020.03.26.
날씨소스 [2]	bass90301	2020.03.26.
저의 카톡봇 두번째 소스 (가위바위보) 🙄 [8]	헤라	2020.03.26.

1 2 3

전체보기

이 카페 인기글

안녕하세요

KHP

♡0 💬6

글자의 개수가 넘어가면 자르기 질문

흑까마귀

♡0 💬5

함수를 입력받아 그래프 그리기

Steve28

♡2 💬4



틱택토 (Player vs Bot)



태양, 달 정보 구현 완료

자동학습 봇 특정 말만 무시하기?

헤히
♡0 💬4

요즘 나타나는 새로운 홍보 수법 중 하나가

Dark Tornado
♡0 💬7



틱택토 (Player vs Player)

5만 4천 2005
가르치기 리포트..

Milk2
♡0 💬9