

[| 일반 소스 공유 | >](#)

## 연속 출석 소스 입니다~



진탄 챗봇 고수 1:1 채팅

2019.12.17. 00:16 조회 379

댓글 14 URL 복사

### 공유할 소스 용도

=> 연속 출석 체크용

### 소스 코드

```
=> const scriptName="test.js";
```

```
var attendance = {};  
var attendanceMsgList = ["출석", "ㄸ", "ㄸㄸ", "cc", "cx"];  
var commandString = "/";  
var commandStringLength = commandString.length;
```

```
FS = FileStream;  
var path = "/sdcard/katalkbot/Database/attendance.json";
```

```
function response(room, msg, sender, isGroupChat, replier, ImageDB, packageName, threadId)  
{  
  if(false == IsBotCommand(msg))  
  {  
    return;  
  }  
}
```

```
//출석조회
```

```
if(true == IsAttendanceListCommand(msg))  
{  
  replier.reply(PrintAttendance(room, sender));  
}
```

```
// 출석
```

```
else if(true == IsAttendanceCommand(msg))  
{  
  if(!FS.read(path))  
  {  
    //replier.reply("FS.read()");  
    FS.write(path, "{}");  
  }  
}
```

```
attendance = JSON.parse(FS.read(path));
```

```
if(undefined == attendance[room])  
{  
  //replier.reply("attendance[room] undefined");  
  attendance[room] = {};  
}
```

```
if(undefined == attendance[room][sender])  
{
```

```

//replier.reply("attendance[room][sender] undefined");
attendance[room][sender] = { LastAttendanceDay : '2019-12-09', ContinuedAttendanceDayCount : 1,
TotalAttendanceDayCount : 0 };
}

var LastAttendanceDayString = GetDayString(attendance[room][sender].LastAttendanceDay);
//replier.reply("last : " + LastAttendanceDayString);
var todayString = GetDayString(new Date());
//replier.reply("today : " + todayString);
var diffDay = GetDiffDay(LastAttendanceDayString, todayString);
//replier.reply("diffDay : " + diffDay);

// 출석 일 수가 다르다
if(diffDay >= 1)
{
attendance[room][sender].LastAttendanceDay = todayString;
attendance[room][sender].TotalAttendanceDayCount++;

// 출석일 비교 값이 1이면 연속 출석
if(1 == diffDay)
{
//replier.reply("diffDay == 1");
attendance[room][sender].ContinuedAttendanceDayCount++;
}
else
{
//replier.reply("diffDay != 1");
attendance[room][sender].ContinuedAttendanceDayCount = 0;
}

replier.reply(PrintAttendance(room, sender));
FS.write(path, JSON.stringify(attendance));
}
else
{
replier.reply(sender + "님은 오늘 출석 하셨습니다.");
}
}
}

function PrintAttendance(room, sender)
{
if(undefined == attendance[room])
{
return (sender + "님은 출석 기록이 없습니다.");
}
else if(attendance[room][sender])
{
return (sender + "님은 출석 기록이 없습니다.");
}
else
{
return (sender + "님 출석현황\n" +
"최근 출석 일 : " + attendance[room][sender].LastAttendanceDay + "\n" +
"연속 출석 일 수 : " + attendance[room][sender].ContinuedAttendanceDayCount + "\n");
}
}

```

```

        연속 출석 일 수 : " + attendance[room][sender].ContinuedAttendanceDayCount + "wn +
        "총 출석 일 수 : " + attendance[room][sender].TotalAttendanceDayCount);
    }
}

function GetDayString(day)
{
    var inputDay = day instanceof Date ? day : new Date(day);
    // 2019-12-14
    var today = inputDay.getFullYear() + "-" + ('0' + (inputDay.getMonth() + 1)).slice(-2) + "-" + ('0' +
inputDay.getDate()).slice(-2);
    return today;
}

function GetDiffDay(_date1, _date2)
{
    var date1 = _date1 instanceof Date ? _date1 : new Date(_date1);
    var date2 = _date2 instanceof Date ? _date2 : new Date(_date2);

    date1 = new Date(date1.getFullYear(), date1.getMonth() + 1, date1.getDate());
    date2 = new Date(date2.getFullYear(), date2.getMonth() + 1, date2.getDate());

    var diff = Math.abs(date2.getTime() - date1.getTime());
    diff = Math.ceil(diff / (1000 * 3600 * 24));

    return diff;
}

function IsBotCommand(msg)
{
    var subMsg = msg.substring(0, commandStringLength);
    if(subMsg == commandString)
    {
        return true;
    }

    return false;
}

function IsAttendanceCommand(msg)
{
    var commandMsg = msg.substring(commandStringLength, msg.length);
    for(var i = 0; i < attendanceMsgList.length; i++)
    {
        if(true == commandMsg.includes(attendanceMsgList[i]))
        {
            return true;
        }
    }

    return false;
}

function IsAttendanceListCommand(msg)
{

```

```

var commandMsg = msg.substring(commandStringLength, msg.length);
if(-1 != commandMsg.indexOf("출석조회"))
{
    return true;
}

return false;
}

1일차const scriptName="test.js";

var attendance = {};
var attendanceMsgList = ["출석", "ㄸ ㅅ", "ㄸ ㄸ", "cc", "cx"];
var commandString = "/";
var commandStringLength = commandString.length;

FS = FileStream;
var path = "/sdcard/katalkbot/Database/attendance.json";

function response(room, msg, sender, isGroupChat, replier, ImageDB, packageName, threadId)
{
    if(false == IsBotCommand(msg))
    {
        return;
    }

    //출석조회
    if(true == IsAttendanceListCommand(msg))
    {
        replier.reply(PrintAttendance(room, sender));
    }
    // 출석
    else if(true == IsAttendanceCommand(msg))
    {
        if(!FS.read(path))
        {
            //replier.reply("FS.read()");
            FS.write(path, "{}");
        }

        attendance = JSON.parse(FS.read(path));

        if(undefined == attendance[room])
        {
            //replier.reply("attendance[room] undefined");
            attendance[room] = {};
        }

        if(undefined == attendance[room][sender])
        {
            //replier.reply("attendance[room][sender] undefined");
            attendance[room][sender] = { LastAttendanceDay : '2019-12-09', ContinuedAttendanceDayCount : 1,
TotalAttendanceDayCount : 0 };
        }
    }
}

```

```

var LastAttendanceDayString = GetDayString(attendance[room][sender].LastAttendanceDay);
//replier.reply("last : " + LastAttendanceDayString);
var todayString = GetDayString(new Date());
//replier.reply("today : " + todayString);
var diffDay = GetDiffDay(LastAttendanceDayString, todayString);
//replier.reply("diffDay : " + diffDay);

// 출석 일 수가 다르다
if(diffDay >= 1)
{
    attendance[room][sender].LastAttendanceDay = todayString;
    attendance[room][sender].TotalAttendanceDayCount++;

    // 출석일 비교 값이 1이면 연속 출석
    if(1 == diffDay)
    {
        //replier.reply("diffDay == 1");
        attendance[room][sender].ContinuedAttendanceDayCount++;
    }
    else
    {
        //replier.reply("diffDay != 1");
        attendance[room][sender].ContinuedAttendanceDayCount = 0;
    }

    replier.reply(PrintAttendance(room, sender));
    FS.write(path, JSON.stringify(attendance));
}
else
{
    replier.reply(sender + "님은 오늘 출석 하셨습니다.");
}
}

function PrintAttendance(room, sender)
{
    if(undefined == attendance[room])
    {
        return (sender + "님은 출석 기록이 없습니다.");
    }
    else if(undefined == attendance[room][sender])
    {
        return (sender + "님은 출석 기록이 없습니다.");
    }
    else
    {
        return (sender + "님 출석현황\n" +
            "최근 출석 일 : " + attendance[room][sender].LastAttendanceDay + "\n" +
            "연속 출석 일 수 : " + attendance[room][sender].ContinuedAttendanceDayCount + "\n" +
            "총 출석 일 수 : " + attendance[room][sender].TotalAttendanceDayCount);
    }
}

```

```

function GetDayString(day)
{
    var inputDay = day instanceof Date ? day : new Date(day);
    // 2019-12-14
    var today = inputDay.getFullYear() + "-" + ('0' + (inputDay.getMonth() + 1)).slice(-2) + "-" + ('0' +
inputDay.getDate()).slice(-2);
    return today;
}

```

```

function GetDiffDay(_date1, _date2)
{
    var date1 = _date1 instanceof Date ? _date1 : new Date(_date1);
    var date2 = _date2 instanceof Date ? _date2 : new Date(_date2);

    date1 = new Date(date1.getFullYear(), date1.getMonth() + 1, date1.getDate());
    date2 = new Date(date2.getFullYear(), date2.getMonth() + 1, date2.getDate());

    var diff = Math.abs(date2.getTime() - date1.getTime());
    diff = Math.ceil(diff / (1000 * 3600 * 24));

    return diff;
}

```

```

function IsBotCommand(msg)
{
    var subMsg = msg.substring(0, commandStringLength);
    if(subMsg == commandString)
    {
        return true;
    }

    return false;
}

```

```

function IsAttendanceCommand(msg)
{
    var commandMsg = msg.substring(commandStringLength, msg.length);
    for(var i = 0; i < attendanceMsgList.length; i++)
    {
        if(true == commandMsg.includes(attendanceMsgList[i]))
        {
            return true;
        }
    }

    return false;
}

```

```

function IsAttendanceListCommand(msg)
{
    var commandMsg = msg.substring(commandStringLength, msg.length);
    if(-1 != commandMsg.indexOf("출석조회"))
    {
        return true;
    }
,

```

```
}
```

```
return false;
```

```
}
```

### 최초 출석

오후 11:38 --출석

FS.read()

attendance[room] undefined

attendance[room][sender] undefined

last : 2019-12-09

today : 2019-12-10

diffDay : 1

diffDay == 1

님 출석현황  
최근 출석 일 : 2019-12-10  
연속 출석 일 수 : 1  
총 출석 일 수 : 1

오후 11:38

### 2일차 연속 출석

오후 11:38 --출석

last : 2019-12-10

today : 2019-12-11

diffDay : 1

diffDay == 1

님 출석현황  
최근 출석 일 : 2019-12-11  
연속 출석 일 수 : 2  
총 출석 일 수 : 2

오후 11:38

### 3일차 연속 출석

오후 11:39 --출석

last : 2019-12-11

today : 2019-12-12

diffDay : 1

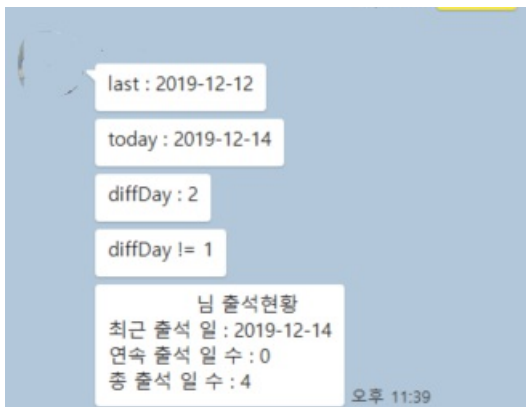
diffDay == 1

님 출석현황  
최근 출석 일 : 2019-12-12  
연속 출석 일 수 : 3  
총 출석 일 수 : 3

오후 11:39

### 5일차 연속 출석 굿김

오후 11:39 --출석



( 말머리 등록 해주세요! )



진탄님의 게시글 더보기 >

❤ 좋아요 1 💬 댓글 14

🔗 공유 | 신고

댓글 등록순 최신순 ↻

댓글알림 ☐



구름

우.와.너.무.어.려.운.걸

2019.12.17. 00:26 답글쓰기



새름

객체사용 및 초보자가 이해하기 어려운 소스 (중급 소스 공유 게시판으로 이동)

2019.12.17. 06:48 답글쓰기



SQLite

고엿네

2019.12.17. 17:10 답글쓰기



jomin398

ㅋㅋㅋ 그냥 작동안하게하셨네 ㅋ

2019.12.18. 13:13 답글쓰기



진탄 작성자

오잉.. 작동 안되나요?

2019.12.18. 13:58 답글쓰기



jomin398

진탄 네... 복붙해봤는데도 여전히 안되서 연구중

2019.12.18. 21:22 답글쓰기



jomin398

작동 이제야되네요... 이 소스에서 날짜 변하는걸 감지하게하는건 뭔가요  
출석 순서 자동으로 메기게 했는데  
그거 몰라서 하루지나면 리셋이 안되네요

2019.12.19. 08:28 답글쓰기



진탄 작성자

날짜 변하는거 감지하는게 날짜 구하는곳이예요~

2019.12.19. 10:28 답글쓰기



jomin398

음....  
그건그렇고 리로딩 직후  
/출석조회  
하면 그냥 오류나오네요....



원래는 try catch 라고 오류 핸들링이 되어있는데 코드에 있건만 .... 봇이 그냥 중지되더라구요.

2019.12.22. 10:51 답글 쓰기



락락

총출석 리스트를 뽑을수 있을까요?  
/목록 << 이렇게치면

# 출석목록입니다

[A]님 - 5회

[B]님 - 3회

이런식으로여

2019.12.30. 18:04 답글 쓰기



jomin398

```
var LastAttendanceDayString = GetDayString(attendance[room][sender].LastAttendanceDay);  
요거 가져오는데에서 에러나더라고요?
```

org.mozilla.javascript.EcmaError: TypeError: Cannot read property "UserName" from undefined (출석봇.js#42)

2020.01.02. 22:40 답글 쓰기



페르조나

.. 작동 안되는 소스인가용..

2020.01.10. 07:18 답글 쓰기



진탄 작성자

작동 되는거 테스트 하고 올렸는데 어떻게 안되죠?

2020.01.10. 21:10 답글 쓰기



페르조나

진탄 메신저로 돌려봤는데 안돼용 ㅠㅠ

2020.01.10. 23:38 답글 쓰기

Hibot

댓글을 남겨보세요



등록

글쓰기

답글

목록

▲ TOP

원하는 기능 주기적으로 실행되는 스케줄러 [3]	땡땡질주	2019.12.17.
네이버 영어사전 [출처] 네이버 영어사전 (카카오톡 봇 커뮤니티)  작성자 Sjjy0916 - 2 [11]	성빈	2019.12.17.
연속 출석 소스 입니다~ ☹️ [14]	진탄	2019.12.17.
소수? ☹️ [6]	줄러	2019.12.16.
문의 소스 ☹️ [2]	김각제	2019.12.16.

123

전체보기

이 카페 인기글

자동학습 봇 특정 말만 무시하기?

헤히  
♡0 💬4

안녕하세요

KHP  
♡0 💬6



틱택토 (카카오톡 링크 적용)

eval 질문

Kiri  
♡0 💬5

에뮬레이터로 카톡 돌리면 정지되나요?

공돌이  
♡0 💬1



네이버에서 지역별 날씨 정보 크롤링하기 /



혹시 여기서 변수를 저장하는 부분이 어딘지 아시는분 계시나요...??



태양, 달 정보 구현 완료



기 나무 곡괭이[노말][으]  
괴락의 문들 중 하나를  
5000분의 1을 얻은 사람

12345