

```

const fs = require('fs'),
ytdl = require('@distube/ytdl-core'),
fse = require("fs-extra"),
moment = require("moment-timezone"),
Youtube = require('youtube-search-api');

module.exports.config = {
  name: "sing",
  version: "1.0.3",
  hasPermission: 0,
  credits: "D-Jukie fix by TKDEV patch lỗi by ChatGPT",
  description: "Nghe nhạc của Youtube ngay trên Messenger",
  commandCategory: "Tiện ích",
  usages: "[tên bài hát]",
  cooldowns: 3,
  usePrefix: true
};

module.exports.run = async function({ api, event, args }) {
  if (!args[0])
    return api.sendMessage("☒ Vui lòng nhập tên bài hát!", event.threadID,
event.messageID);
  try {
    const data = (await Youtube.GetListByKeyword(args.join(" "), false,
6)).items.filter(i => i.type === "video");
    if (!data.length)
      return api.sendMessage("☒ Không tìm thấy bài nào phù hợp!",
event.threadID, event.messageID);
  }

```

```

const msg = data.map((v, i) =>
  `|> ${i + 1}. ${v.title}\n|> ☒ ${v.channelTitle}\n|> ☒☒
  ${v.length.simpleText}\n—————`
).join('\n');
const link = data.map(v => v.id);
return api.sendMessage(
  `☒ Kết quả:\n${msg}\n\n☒ Reply STT để bot gửi nhạc cho bạn!`,
  event.threadID,
  (err, info) => global.client.handleReply.push({
    type: 'reply',
    name: module.exports.config.name,
    author: event.senderID,
    messageID: info.messageID,
    link
  }),
  event.messageID
);
} catch (e) {
  console.error(e);
  return api.sendMessage("☒ Lỗi khi tìm kiếm bài hát!", event.threadID,
event.messageID);
}
};

```

```

module.exports.handleReply = async function({ api, event, handleReply }) {
  const { threadID, messageID, body, senderID } = event;
  const id = handleReply.link[parseInt(body) - 1];
  if (!id)

```

```
return api.sendMessage("☒ Số bạn chọn không hợp lệ", threadID,
messageID);

const path = `${__dirname}/cache/sing-${senderID}.mp3`;

try {
  ytdl.cache.update = () => {}; // fix lỗi cache decipher
  const info = await ytdl.getInfo(`https://www.youtube.com/watch?v=${id}`);
  const v = info.videoDetails;

  const audioFormats = ytdl.filterFormats(info.formats, 'audioonly');
  const format = audioFormats.find(f =>
    f.mimeType?.includes('audio/mp4') && f.audioBitrate <= 128
  ) || audioFormats[0]; // fallback

  if (!format?.url)
    return api.sendMessage("☒ Không thể tìm thấy định dạng nào phù hợp!",
threadID, messageID);

  const stream = ytdl.downloadFromInfo(info, {
    format,
    highWaterMark: 1 << 25
  }).pipe(fs.createWriteStream(path));

  stream.on('finish', () => {
    const size = fs.statSync(path).size;
    if (size > 26214400)
      return api.sendMessage("☒ File quá lớn không thể gửi lên Messenger!",
```

```
threadID, () => fse.unlinkSync(path), messageID);
```

```
    api.unsendMessage(handleReply.messageID);  
    api.sendMessage({  
      body: `=== [ YouTube Music ] ===`
```

```
☒ Tên bài hát: ${v.title}
```

```
☒☒ Thời lượng: ${convertHMS(v.lengthSeconds)} |
```

```
☒ Tác giả: ${v.author.name}
```

```
☒ Ngày đăng: ${v.uploadDate}
```

```
☒☒ Lượt xem: ${v.viewCount}
```

```
☒ ${moment.tz("Asia/Ho_Chi_Minh").format("HH:mm:ss | DD/MM/YYYY")}
```

```
☒☒`,
```

```
    attachment: fs.createReadStream(path)
```

```
  }, threadID, () => fse.unlinkSync(path), messageID);
```

```
});
```

```
stream.on('error', e => {
```

```
  console.error(e);
```

```
  return api.sendMessage("☒ Lỗi khi tải bài hát!", threadID, messageID);
```

```
});
```

```
} catch (e) {
```

```
  console.error(e);
```

```
  return api.sendMessage("☒ Đã xảy ra lỗi khi tải bài hát!", threadID,  
messageID);
```

```
}
```

```
};
```

```
function convertHMS(s) {  
  const h = Math.floor(s / 3600),  
        m = Math.floor((s % 3600) / 60),  
        sec = s % 60;  
  return [h, m, sec].map(v => v < 10 ? "0" + v : v)  
    .filter((v, i) => v !== "00" || i > 0)  
    .join(":");  
}
```