

[| 기타 질문 | >](#)

## 파일 리스트 불러오기



천익 챗봇 고수 1:1 채팅

2020.03.18. 14:26 조회 90

댓글 2 URL 복사 ⋮

질문할 어플 (새자봇, 젤브봇, 메신저봇, 닥토봇, 기타)

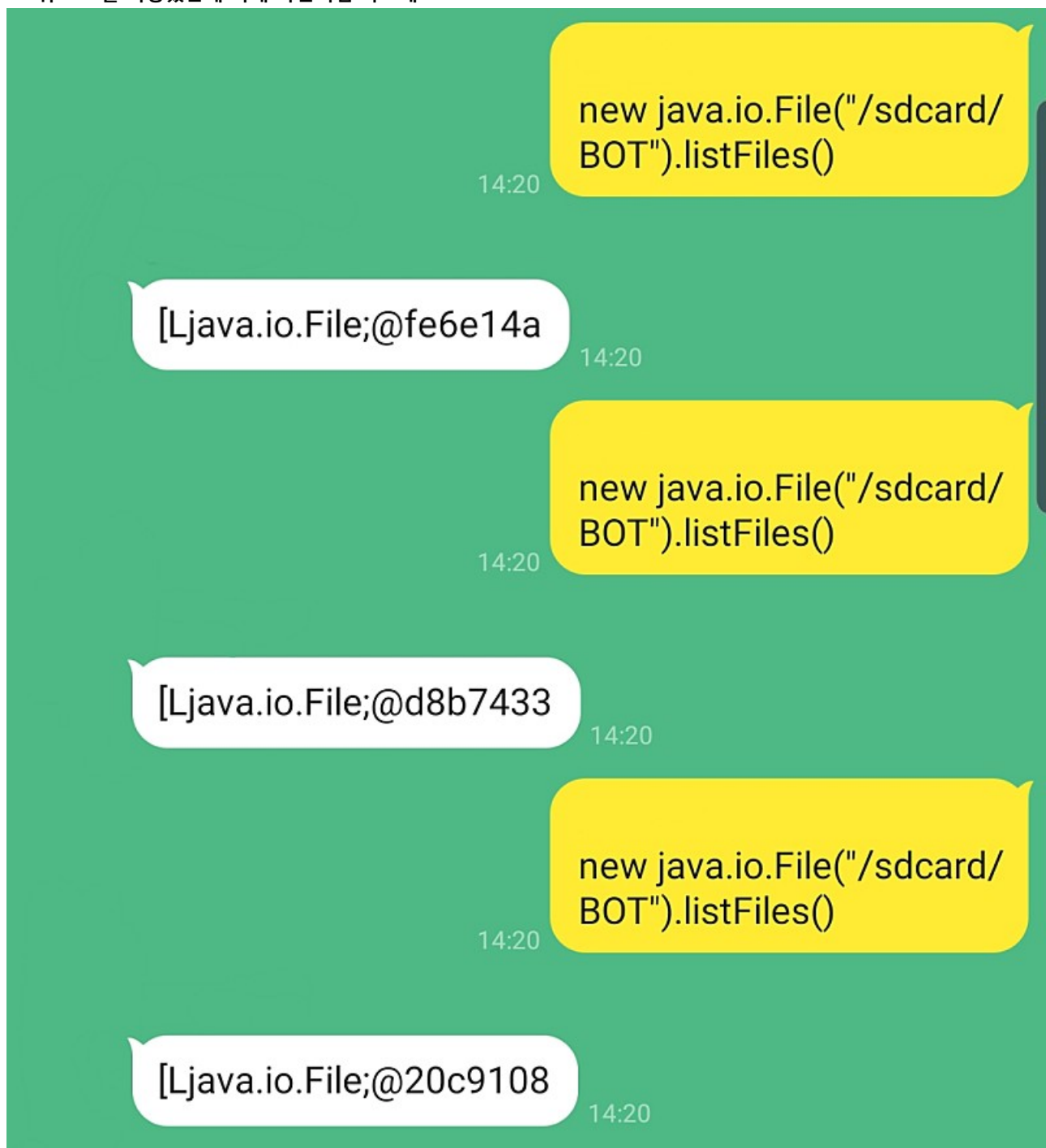
=> 메신저봇(파란봇)

질문할 소스

=> new java.io.File("경로").listFiles()

질문내용

=> 위 소스를 사용했는데 아래 사진처럼 나오네요



new java.io.File("/sdcard/  
BOT").listFiles()

14:20

[Ljava.io.File;@4d82d9

14:20

어떻게하면 되죠?



천익님의 게시글 더보기 >

❤ 좋아요 0 💬 댓글 2

🔗 공유 | 신고

댓글 등록순 최신순 ↻

댓글알림 ☐



새름  
join("₩n")

2020.03.18. 14:27 답글쓰기



천익 작성자  
아...ㅋㅋ 감사합니당

2020.03.18. 14:30 답글쓰기

Hibot

댓글을 남겨보세요



등록

✎ 글쓰기

답글

목록

▲ TOP

'| 기타 질문 |' 게시판 글

이 게시판 새글 구독하기 ☐

자바,커피 [2]

르르르르

2020.03.18.

sleep기능 질문 [3]	애플망고	2020.03.18.
파일 리스트 불러오기 ☹️ [2]	천익	2020.03.18.
카카오링크 param 연결 질문 ☹️ [1]	비닐봉지	2020.03.18.
초보 질문! [9]	양01	2020.03.17.

1 2 3

전체보기

## 이 카페 인기글

안녕하세요!

트위터봇  
♡0 💬6

가르치기 리로드..

Milk2  
♡0 💬9

간편 자동응답과 한글코딩을 메인으로 제공  
하는 카톡봇앱 개발 예정

성빈  
♡1 💬16

자바스크립트

도유니  
♡0 💬10

[illegible]

### 틱택토 (Player vs Bot)

```

25 if (part instanceof MenuItem)
26     replace.replace(1, 0, replace);
27     return;
28 }
29 }
30
31 insert(part) = void
32 replace.replace(part instanceof MenuItem ? 1 : 0, 1, part)
33 }
34 }

```

혹시 여기서 변수를 저장하는 부분이 어딘지

1 2 3 4 5

반가워요.

천방지축하연  
♡0 💬4

```

41:         }
42:         register napi_t* napi = (napi_t*)calloc(1, sizeof(napi_t));
43:         napi->name = name;
44:         napi->data = data;
45:         register napi_t* napi2 = (napi_t*)calloc(1, sizeof(napi_t));
46:         napi2->name = name;
47:         napi2->data = data;
48:         if (NAPI_SUCCESS != napi2->init()) {
49:             return napi;
50:         }
51:         napi2->name = name;
52:         napi2->data = data;
53:         if (NAPI_SUCCESS != napi2->init()) {
54:             return napi;
55:         }
56:         return napi2;
57:     }
58:     ~Napi() {
59:         if (NAPI_SUCCESS != napi->close()) {
60:             return;
61:         }
62:         if (NAPI_SUCCESS != napi2->close()) {
63:             return;
64:         }
65:     }
66: };
67:
68: void Napi::Init() {
69:     napi_t* napi = (napi_t*)calloc(1, sizeof(napi_t));
70:     napi->name = "Napi";
71:     napi->data = (void*)0;
72:     napi->init();
73:     napi_t* napi2 = (napi_t*)calloc(1, sizeof(napi_t));
74:     napi2->name = "Napi2";
75:     napi2->data = (void*)0;
76:     napi2->init();
77: }
78:
79: void Napi::Close() {
80:     napi_t* napi = (napi_t*)calloc(1, sizeof(napi_t));
81:     napi->name = "Napi";
82:     napi->data = (void*)0;
83:     napi->close();
84:     napi_t* napi2 = (napi_t*)calloc(1, sizeof(napi_t));
85:     napi2->name = "Napi2";
86:     napi2->data = (void*)0;
87:     napi2->close();
88: }
89:
90: void Napi::Close2() {
91:     napi_t* napi = (napi_t*)calloc(1, sizeof(napi_t));
92:     napi->name = "Napi";
93:     napi->data = (void*)0;
94:     napi->close();
95:     napi_t* napi2 = (napi_t*)calloc(1, sizeof(napi_t));
96:     napi2->name = "Napi2";
97:     napi2->data = (void*)0;
98:     napi2->close();
99: }
100:
101: void Napi::Close3() {
102:     napi_t* napi = (napi_t*)calloc(1, sizeof(napi_t));
103:     napi->name = "Napi";
104:     napi->data = (void*)0;
105:     napi->close();
106:     napi_t* napi2 = (napi_t*)calloc(1, sizeof(napi_t));
107:     napi2->name = "Napi2";
108:     napi2->data = (void*)0;
109:     napi2->close();
110: }
111:
112: void Napi::Close4() {
113:     napi_t* napi = (napi_t*)calloc(1, sizeof(napi_t));
114:     napi->name = "Napi";
115:     napi->data = (void*)0;
116:     napi->close();
117:     napi_t* napi2 = (napi_t*)calloc(1, sizeof(napi_t));
118:     napi2->name = "Napi2";
119:     napi2->data = (void*)0;
120:     napi2->close();
121: }
122:
123: void Napi::Close5() {
124:     napi_t* napi = (napi_t*)calloc(1, sizeof(napi_t));
125:     napi->name = "Napi";
126:     napi->data = (void*)0;
127:     napi->close();
128:     napi_t* napi2 = (napi_t*)calloc(1, sizeof(napi_t));
129:     napi2->name = "Napi2";
130:     napi2->data = (void*)0;
131:     napi2->close();
132: }
133:
134: void Napi::Close6() {
135:     napi_t* napi = (napi_t*)calloc(1, sizeof(napi_t));
136:     napi->name = "Napi";
137:     napi->data = (void*)0;
138:     napi->close();
139:     napi_t* napi2 = (napi_t*)calloc(1, sizeof(napi_t));
140:     napi2->name = "Napi2";
141:     napi2->data = (void*)0;
142:     napi2->close();
143: }
144:
145: void Napi::Close7() {
146:     napi_t* napi = (napi_t*)calloc(1, sizeof(napi_t));
147:     napi->name = "Napi";
148:     napi->data = (void*)0;
149:     napi->close();
150:     napi_t* napi2 = (napi_t*)calloc(1, sizeof(napi_t));
151:     napi2->name = "Napi2";
152:     napi2->data = (void*)0;
153:     napi2->close();
154: }
155:
156: void Napi::Close8() {
157:     napi_t* napi = (napi_t*)calloc(1, sizeof(napi_t));
158:     napi->name = "Napi";
159:     napi->data = (void*)0;
160:     napi->close();
161:     napi_t* napi2 = (napi_t*)calloc(1, sizeof(napi_t));
162:     napi2->name = "Napi2";
163:     napi2->data = (void*)0;
164:     napi2->close();
165: }
166:
167: void Napi::Close9() {
168:     napi_t* napi = (napi_t*)calloc(1, sizeof(napi_t));
169:     napi->name = "Napi";
170:     napi->data = (void*)0;
171:     napi->close();
172:     napi_t* napi2 = (napi_t*)calloc(1, sizeof(napi_t));
173:     napi2->name = "Napi2";
174:     napi2->data = (void*)0;
175:     napi2->close();
176: }
177:
178: void Napi::Close10() {
179:     napi_t* napi = (napi_t*)calloc(1, sizeof(napi_t));
180:     napi->name = "Napi";
181:     napi->data = (void*)0;
182:     napi->close();
183:     napi_t* napi2 = (napi_t*)calloc(1, sizeof(napi_t));
184:     napi2->name = "Napi2";
185:     napi2->data = (void*)0;
186:     napi2->close();
187: }
188:
189: void Napi::Close11() {
190:     napi_t* napi = (napi_t*)calloc(1, sizeof(napi_t));
191:     napi->name = "Napi";
192:     napi->data = (void*)0;
193:     napi->close();
194:     napi_t* napi2 = (napi_t*)calloc(1, sizeof(napi_t));
195:     napi2->name = "Napi2";
196:     napi2->data = (void*)0;
197:     napi2->close();
198: }
199:
200: void Napi::Close12() {
201:     napi_t* napi = (napi_t*)calloc(1, sizeof(napi_t));
202:     napi->name = "Napi";
203:     napi->data = (void*)0;
204:     napi->close();
205:     napi_t* napi2 = (napi_t*)calloc(1, sizeof(napi_t));
206:     napi2->name = "Napi2";
207:     napi2->data = (void*)0;
208:     napi2->close();
209: }
210:
211: void Napi::Close13() {
212:     napi_t* napi = (napi_t*)calloc(1, sizeof(napi_t));
213:     napi->name = "Napi";
214:     napi->data = (void*)0;
215:     napi->close();
216:     napi_t* napi2 = (napi_t*)calloc(1, sizeof(napi_t));
217:     napi2->name = "Napi2";
218:     napi2->data = (void*)0;
219:     napi2->close();
220: }
221:
222: void Napi::Close14() {
223:     napi_t* napi = (napi_t*)calloc(1, sizeof(napi_t));
224:     napi->name = "Napi";
225:     napi->data = (void*)0;
226:     napi->close();
227:     napi_t* napi2 = (napi_t*)calloc(1, sizeof(napi_t));
228:     napi2->name = "Napi2";
229:     napi2->data = (void*)0;
230:     napi2->close();
231: }
232:
233: void Napi::Close15() {
234:     napi_t* napi = (napi_t*)calloc(1, sizeof(napi_t));
235:     napi->name = "Napi";
236:     napi->data = (void*)0;
237:     napi->close();
238:     napi_t* napi2 = (napi_t*)calloc(1, sizeof(napi_t));
239:     napi2->name = "Napi2";
240:     napi2->data = (void*)0;
241:     napi2->close();
242: }
243:
244: void Napi::Close16() {
245:     napi_t* napi = (napi_t*)calloc(1, sizeof(napi_t));
246:     napi->name = "Napi";
247:     napi->data = (void*)0;
248:     napi->close();
249:     napi_t* napi2 = (napi_t*)calloc(1, sizeof(napi_t));
250:     napi2->name = "Napi2";
251:     napi2->data = (void*)0;
252:     napi2->close();
253: }
254:
255: void Napi::Close17() {
256:     napi_t* napi = (napi_t*)calloc(1, sizeof(napi_t));
257:     napi->name = "Napi";
258:     napi->data = (void*)0;
259:     napi->close();
260:     napi_t* napi2 = (napi_t*)calloc(1, sizeof(napi_t));
261:     napi2->name = "Napi2";
262:     napi2->data = (void*)0;
263:     napi2->close();
264: }
265:
266: void Napi::Close18() {
267:     napi_t* napi = (napi_t*)calloc(1, sizeof(napi_t));
268:     napi->name = "Napi";
269:     napi->data = (void*)0;
270:     napi->close();
271:     napi_t* napi2 = (napi_t*)calloc(1, sizeof(napi_t));
272:     napi2->name = "Napi2";
273:     napi2->data = (void*)0;
274:     napi2->close();
275: }
276:
277: void Napi::Close19() {
278:     napi_t* napi = (napi_t*)calloc(1, sizeof(napi_t));
279:     napi->name = "Napi";
280:     napi->data = (void*)0;
281:     napi->close();
282:     napi_t* napi2 = (napi_t*)calloc(1, sizeof(napi_t));
283:     napi2->name = "Napi2";
284:     napi2->data = (void*)0;
285:     napi2->close();
286: }
287:
288: void Napi::Close20() {
289:     napi_t* napi = (napi_t*)calloc(1, sizeof(napi_t));
290:     napi->name = "Napi";
291:     napi->data = (void*)0;
292:     napi->close();
293:     napi_t* napi2 = (napi_t*)calloc(1, sizeof(napi_t));
294:     napi2->name = "Napi2";
295:     napi2->data = (void*)0;
296:     napi2->close();
297: }
298:
299: void Napi::Close21() {
300:     napi_t* napi = (napi_t*)calloc(1, sizeof(napi_t));
301:     napi->name = "Napi";
302:     napi->data = (void*)0;
303:     napi->close();
304:     napi_t* napi2 = (napi_t*)calloc(1, sizeof(napi_t));
305:     napi2->name = "Napi2";
306:     napi2->data = (void*)0;
307:     napi2->close();
308: }
309:
310: void Napi::Close22() {
311:     napi_t* napi = (napi_t*)calloc(1, sizeof(napi_t));
312:     napi->name = "Napi";
313:     napi->data = (void*)0;
314:     napi->close();
315:    
```

### 틱택토 (Player vs Player)

연이 | | | | | | | | | |  
여름  
[학습] 모니카는 여기에 잘 있음  
년이 | | | | | | | | | |

[카카오톡 봇] 양산형 자동학습 봇 소스 공유