

챗봇과 마이크로소프트 봇 프레임워크

- 김영욱 (한국마이크로소프트 부장)
- 이강호 (국립한국복지대학교 컴퓨터정보보안과 교수)

I. 챗봇의 개념

2018년 초기부터 챗봇(Chat Bot)에 관한 관심이 뜨거워졌다. 챗봇이라는 개념이 없었던 개념이 아니지만 지금은 챗봇을 개발 하기 위한 여러가지 기술이 무르익었다. 하지만 본격적으로 들어가기 먼저 이야기 할 것은 챗봇은 아이폰엔에 등장하는 자비스나 터미네이터와 같은 완전한 인공지능으로 생각하면 안 된다.

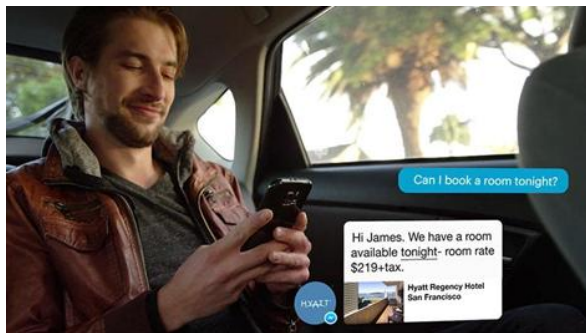


Fig. 1. 챗봇을 이용해서 호텔을 예약하는 모습[1]

챗봇은 애플의 Siri, Microsoft의 코타나와 같은 개인비서와는 달리 특정한 용도나 업무용으로 최적화 된다. 챗봇은 용도를 제한했기 때문에 빠르게 현실적인 서비스를 제공할 수 있게 되었고 빠르게 확산되고 있다고 볼 수 있다. 챗봇은 이미 다양한 용도로 이미 개발되고 있거나 혹은 이미 서비스를 제공하고 있는데 그 중에 몇 가지 시나리오를 살펴 보면 아래와 같다.

- 주문: 가장 흔한 것은 피자나 자장면과 같은 주문관련 기능들을 구현하는 것이다. 비교적 시나리오가 단순하고 도입 결과가 분명하기 때문에 가장 빠르게 챗봇이 도입되고 있는 분야이다.

- 예약: 호텔이나 미용, 교통편과 같이 예약 분야도 챗봇이 가장 흔하게 언급되는 분야이다. 항공권 최저가를 검색해주는 스카이스캐너(SkyScanner)와 같은 분야에서는 벌써 챗봇의 사용이 활발히 이루어지고 있다[2].

- 고객지원: FAQ와 같이 반복적으로 이루어지는 문의를 처리하는 봇으로 가시적인 효과가 바로 나타나는 분야이다. 회사마다 편차는 있지만 대부분의 고객문의는 80%이상이 단순 문의가 많기 때문에 챗봇을 활용해서 단순 문의만 해결해도 상당한 효과를 볼 수 있다.

- 교육: 대화형으로 질문에 응답하거나 학습 진도를 챙겨주는 LMS(Learning Management System), 콘텐츠를 관리하는 LCMS(Learning Content Management System)등과 같은 기존 시스템과 연동되어 개인별로 친절하게 학습 안내를 해준다. 또 퀴즈를 통해서 학습 성취도를 체크하거나 평가할 수도 있다.

- 물류: 단순히 배송중인 물건의 위치나 상황 등을 안내해 줄 수도 있지만 과거 데이터를 기반으로 도착 예정 시간을 알려주는 봇을 만들 수 있다.

몇 가지 사례를 우선 살펴 보았지만 봇을 활용할 수 있는 시나리오는 이외에도 무궁무진하다.

국내에서도 챗봇을 개발하고 활용하는 사례는 어렵지 않게 찾을 수 있다. 대한항공, 아시아나 항공 그리고 진에어까지 항공사들은 빠르게 챗봇 서비스를 제공하고 있다.

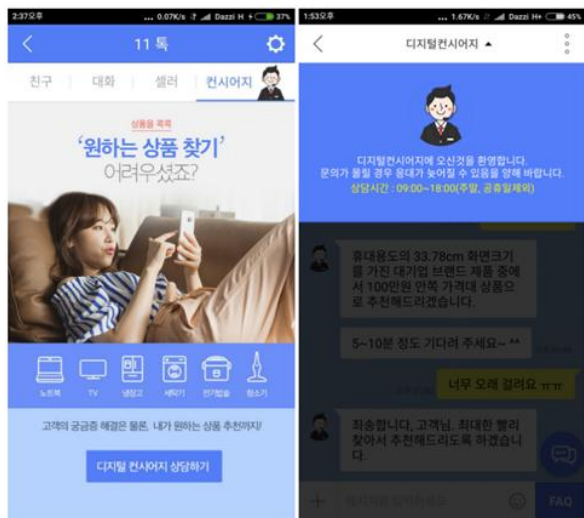


Fig. 2. 11번가의 챗봇의 활용

인터넷 쇼핑몰들도 빠르게 챗봇 서비스를 도입하고 있다. 하지만 한 가지 다른 점은 아무래도 매출과 연결되다 보니 고객과 좀 더 밀착된 서비스를 제공하기 위해서 챗봇 + 상담원을 함께 서비스하는 경우가 많다. 그래서 처음 접속했을 때에는 챗봇이었는데 좀 더 구체적으로 물어보거나 특정 제품의 추천을 요구하면 상담원이 넘겨 받아서 상담을 진행하는 방식이다. 이와 같은 방식으로도 상담원들의 부하를 제법 줄일 수 있을 뿐 아니라 비슷한 질문을 받았을 경우에는 기존의 답변을 그대로 사용할 수 있기 때문에 효과적이다.

II. 챗봇의 개발

봇을 개발할 때에는 기본적으로 웹 개발을 지원하는 모든 언어로 개발이 가능하지만 여러 형태로 제공되는 챗봇 라이브러리를 사용하면 개발의 시행 착오를 상당부분 줄일 수 있다.

챗봇을 만들 수 있는 플랫폼은 Microsoft의 Bot Framework 이외에도 ChatScript, Pandorabots, facebook의 Bots for Messenger, Rebot.me, Imperson 등 여러 플랫폼이 있지만 현재까지 가장 많은 메신저를 지원하는 플랫폼은 Microsoft Bot Framework이다[3].

챗봇이란 채팅을 해주는 봇(Bot)이라고 짧게 설명할 수 있지만 실제로 봇을 개발하다 보면 채팅이라는 인터페이스 뒤에 숨어 있는 표 1의 다양한 기술들이 필요하다. 사용자가 입력한 문장을 이해 할 수 있게 하는 자연어 처리 기술이나 사용자의 문장이나 사진을 분석하는 패턴 인식 기술들 그리

고 정리되어 있는 데이터베이스나 저장공간에서 필요한 정보를 처리해주는 텍스트 마이닝 기술도 필요하다. 물론 모든 기술이 한꺼번에 다 필요하지는 않지만 대개의 경우 적절한 서비스들을 조합해서 개발하게 된다.

Table 1. 챗봇 개발에 필요한 다양한 기술

관련 기술	주요내용
패턴인식 (Pattern Recognition)	기계에 의하여 도형, 문자, 음성 등을 식별하는 것
자연어처리 (natural Language Processing)	인간이 보통 쓰는 언어를 컴퓨터에 인식시켜 처리하는 일 정보검색 질의 응답, 시스템 자동 번역, 통역 등이 포함됨
시맨틱 웹 (Symantic Web)	컴퓨터가 정보 자원의 뜻을 이해하고 논리적 추론까지 할 수 있는 차세대 지능형 웹
텍스트 마이닝 (TextMining)	비정형 텍스트 데이터에서 새롭고 유용한 정보를 찾아내는 과정 또는 기술
상황인식 컴퓨팅 (ContextAware Computing)	가상공간에서 현실의 상황을 정보화하고 이를 활용하여 사용자 중심의 지능화된 서비스를 제공하는 기술.

III. 마이크로소프트 봇 프레임워크 (Bot Framework)

Microsoft는 2016년 3월 31일 매년 개최 해왔던 빌드 컨퍼런스에서 봇 프레임워크(Bot Framework)를 발표 했다. Microsoft가 발표한 봇 프레임워크는 .NET과 Node.js 두 가지 언어를 지원하고 클라우드 기반의 봇 포털을 통해서 대부분의 메신저와의 연결이 가능하게 했다. 또 제공되는 SDK(Software Development Kit)은 Github(<https://github.com/Microsoft/BotBuilder>)를 통해서 오픈소스로 제공되고 있다.

Microsoft Bot Framework

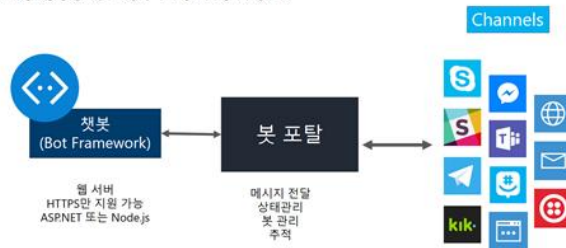


Fig. 3. Microsoft Bot Framework 구성

Microsoft의 봇 프레임워크는 인기 있는 메신저들을 대부분 지원하고 있다. 여기서는 메신저들을 대화를 위한 채널이라고 이야기 한다. Microsoft가 공식적으로 지원하는 채널로는 글을 쓰고 있는 현재로는 Skype, Web, Email(Office 365), Facebook, GroupMe, Kik, Slack, Telegram, Twilio(SMS), Direct Line App Integration 등 글로벌 하게 많이 쓰이고 있는 대부분의 메신저를 지원하고 있다. 지금도 계속 채널로 협의중인 메신저가 있어서 향후 더 많은 메신저들이 채널에 포함 될 것으로 보인다.

채널에 포함되어 있는 메신저들은 봇 서비스(Bot Service)에 연결되어 있다. 봇 포탈은 클라우드 상에서 제공되고 있는 서비스로 항상 채널들과 연결을 유지하고 메시지를 중계하고 상태를 관리하는 등 다양한 기능을 제공한다. 이렇게 봇 포탈을 통해서 메신저들을 연결하는 방식은 연결 방식이 서로 상이한 메신저들을 연결하는데 있어서 상당한 이점을 제공해 준다. 어떤 메신저를 사용하든 간에 봇 프레임워크에서는 연결 방식이나 개발방법에 있어서 거의 동일한 방식으로 개발할 수 있게 해준다.

C#이나 Node.js로 개발한 봇은 웹서버에 배포해야 하는데 웹서버를 나중에 봇 포탈에 등록하는 과정을 거치게 된다. 이때 웹서버는 반드시 HTTPS를 지원해야 한다. 이는 개인 정보보호 문제가 민감한 대화를 다루는 서비스이기 때문에 기본적으로 보안을 고려할 수 밖에 없기 때문이다.

만약 .NET이나 Node.js가 아니거나 혹은 메신저가 아니라 별도의 디바이스나 앱에서 봇을 사용하고 싶다면 제공되는 Direct Line App Integration를 이용해서 연결 할 수 있다.

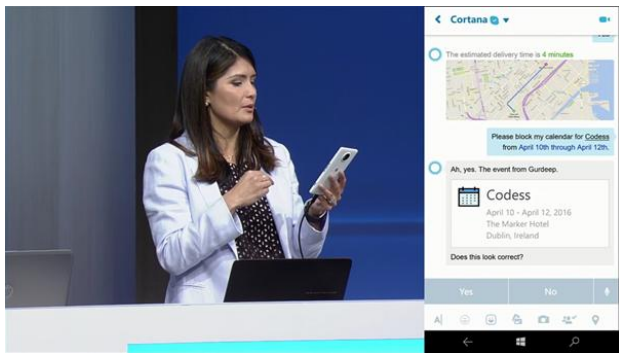


Fig. 4. Build Conference 2017에서 봇 프레임워크를 발표하는 모습

IV. 마이크로소프트 봇 프레임워크 한계와 해결책

Microsoft의 봇 프레임워크는 스카이프나 페이스북 메신저, 슬랙과 텔레그램 등 최근 인기 있는 대부분의 메신저는 모두 지원하고 있지만 그렇다고 해도 모든 메신저를 다 지원하지는 못한다.

또 국내에서는 가장 많은 사람들이 사용하고 있는 카카오톡을 직접적으로 지원하지 못하고 있는 점도 한계점으로 볼 수 있다. 하지만 카카오톡의 경우 페이스북 메신저나 스카이프 등 다른 메신저들과는 달리 관련된 Open API를 제공하지 않고 있어서 사실상 연결이 쉽지 않은 상태이다. 다만 카카오톡은 특정 용도로 사용되는 옐로우 아이디(Yellow ID)만 연동 가능하게 열어둔 상태이기 때문에 만약에 카카오톡 연결이 꼭 필요하다면 옐로우 아이디를 먼저 개설하는 작업이 먼저 필요하다. 옐로우 아이디는 기업이나 특정 단체의 대표 카카오톡 아이디를 만드는 것으로 카카오톡을 이용해서 빠르게 고객과 소통할 수 있는 장점이 있다. 옐로우 아이디에 관한 정보는 웹사이트(<http://yellowid.kakao.com/>)에서 볼 수 있다.

Table 2. 메신저 별 사용자 수









메신저	 WhatsApp	 Facebook	 QQ Mobile
가입자	1000	900	853
메신저	 WeChat	 Viber	 LINE
가입자	697	249	215
메신저	 Snap Chat	 Kik	 Kakaotalk
가입자	200	100	48



Fig. 5. 카카이가 제공하는 YellowID[4]

V. 개발 환경과 Hello World 봇 제작

모든 개발은 메모장으로도 가능하지만 가장 적절한 도구를 사용하는 것은 생산성 측면에서 불가피한 선택이 된다. 봇 프레임워크를 다루기 위해서는 기본적으로 Visual Studio 2017을 사용하는 것은 필수라고 할 수 있겠다.

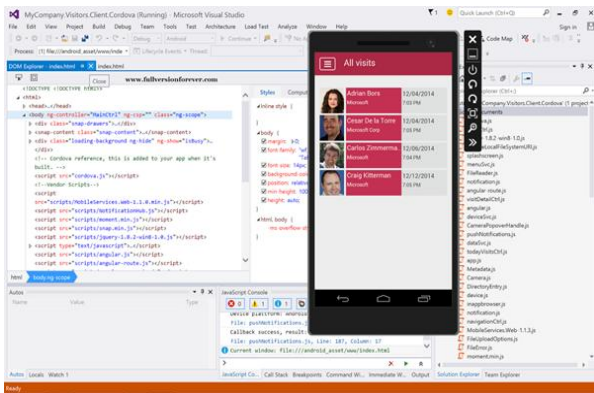


Fig. 6. Visual Studio 2017

Visual Studio 2017은 봇 프레임워크 개발 이외에도 iOS, Android 개발을 위한 Xamarin과 Windows 앱, 웹 개발 등 거의 모든 영역을 개발할 수 있는 좋은 개발도구이다. 아래는 봇 프레임워크를 개발하기 위해서 필요한 개발 도구 들이다.

- Visual Studio 2015 또는 상위버전
2017 이상의 버전을 권장한다. Visual Studio는 정식으로 구입한 라이선스를 가지고 있다면 해당 버전을 사용하면 되고

그렇지 않은 경우라면 Visual Studio Community 버전을 다운로드 받아 사용하면 된다. Visual Studio Community는 상용버전인 Professional 버전과 동일한 기능을 제공하고 있으며 학생이나 비영리 기관 스타트업 등에 대상에 따라서 무료로 사용할 수 있다. 물론 현업에 사용할 때에는 라이선스 제약이 있으니 라이선스를 미리 확인하고 사용해야 한다. 자세한 내용과 다운로드는 <http://visualstudio.com> 에서 할 수 있다. 가끔 적이면 봇 프레임워크와 같이 새로운 기술을 사용할 경우에는 영문판 Visual Studio를 사용하는 것이 경험상 유리하다.

- 봇 프로젝트 템플릿

Visual Studio 2017의 경우 봇 프레임워크를 사용할 수 있는 템플릿이 기본으로 포함되어 있지 않다. 그래서 추가적으로 봇 프레임워크 템플릿을 다운로드 받아서 설치해 주어야 한다. 다운로드 주소는 바뀔 수 있지만 봇 프레임워크 사이트에 있는 링크를 따라가서 템플릿을 다운로드 받을 수 있다[5].

다운로드 받은 파일은 Bot Application.zip이라는 하나의 압축 파일로 되어 있는데 이 파일을 Documents\Visual Studio 2017\Templates\Project Templates\ Visual C# 아래에 복사하고 Visual Studio 를 다시 실행시키면 봇 애플리케이션을 선택할 수 있다.

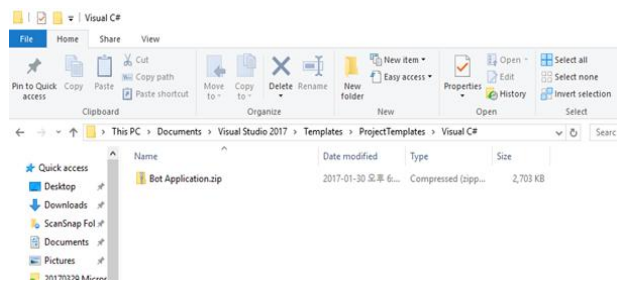


Fig. 7. 봇 템플릿 복사

- 봇 에뮬레이터

봇 에뮬레이터는 봇을 개발하고 테스트 하는 과정에서 각 메신저들을 대신해서 봇을 테스트 할 수 있게 해준다. 봇 에뮬레이터가 없으면 매번 메신저를 설정해서 테스트 해야 하며 또 메신저에서 주고 받는 메시지 이외에도 몇 가지 신호들을 줄 수 있고 또 봇을 개발할 때 필수적이다.

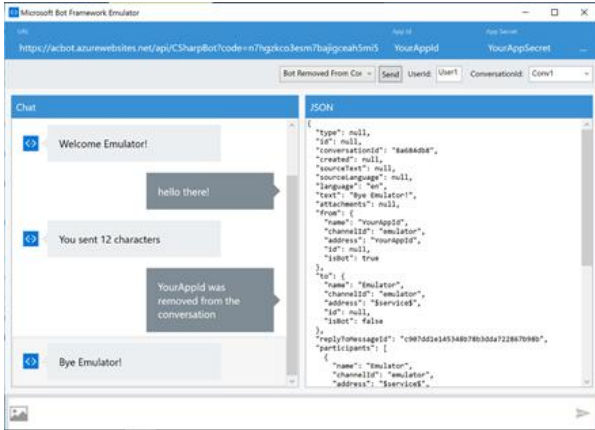


Fig. 8. 봇 에뮬레이터 실행 화면

이제 모든 개발의 시작이라고 하는 ‘Hello World’를 출력해 볼 차례가 되었다. 개발 환경이 잘 구성이 되었으면 첫 번째 샘플은 템플릿에 기본으로 포함되어 있는 코드를 실행하면서 전반적인 구성을 이해한다. 먼저 Visual Studio를 실행하고 File → New → Project를 선택하면 그림 9와 같이 프로젝트를 생성할 수 있는 화면이 나타난다.

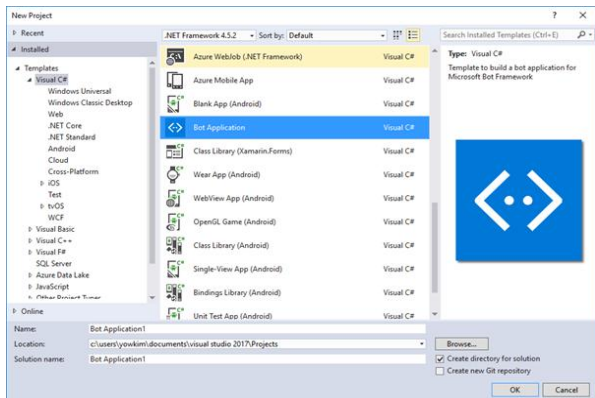


Fig. 9. 프로젝트 생성화면

그림 9에서 먼저 Bot Application을 선택한다. 그리고 프로젝트 이름을 결정한다. 여기서는 기본으로 설정되어 있는 Bot Application1을 사용한다. 그 다음으로는 프로젝트가 저장될 위치를 결정한다. 기본값은 Document\Visual Studio 2017\Projects가 선택되어 있다. 솔루션 이름은 기본적으로 프로젝트 이름과 동일하게 지정되어 있다. 이제 마지막으로 OK 버튼을 누르면 그림 10과 같이 프로젝트가 생성된다.

생성된 프로젝트를 솔루션 탐색기(Solution Explorer)에서 살펴보면 생성된 파일 목록을 볼 수 있다. Web.config 파

일은 프로젝트 설정과 관련된 내용들이 포함되어 있다. default.htm 파일은 아무 역할을 하지 않는다. 챗봇은 기본적으로 웹 프로그래밍 기술로 구성되어 있기 때문에 프로젝트를 실행시켰을 때 기본 페이지를 보여주기 위해서 존재하고 있다. 실제 채팅과 관련된 코드는 Controllers 폴더 하위에 있는 Messages Controller.cs 파일에 있다.

MessagesController.cs 파일을 열어보면 Post() 메소드를 찾아 볼 수 있다. Post()는 사용자들이 챗봇에게 메시지를 보내면 받게 되는 메소드이다.

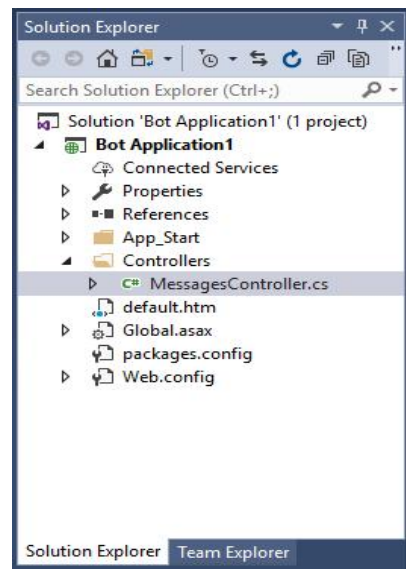


Fig. 10. 봇 프레임워크 템플릿에 의해서 생성된 프로젝트

```

1: public async Task<HttpResponseMessage> Post([FromBody]Activity activity)
2: {
3:     if (activity.Type == ActivityTypes.Message)
4:     {
5:         ConnectorClient connector = new ConnectorClient(new Uri(activity.ServiceUrl));
6:         // calculate something for us to return
7:         int length = (activity.Text ?? string.Empty).Length;
8:
9:         // return our reply to the user
10:        Activity reply = activity.CreateReply($"You sent {activity.Text} which was {length} characters");
11:        await connector.Conversations.ReplyToActivityAsync(reply);
12:    }
13:    else
14:    {
15:        HandleSystemMessage(activity);
16:    }
17:    var response = Request.CreateResponse(HttpStatusCode.OK);
18:    return response;

```

Fig. 11. 기본 샘플 코드

그림 11의 내용을 먼저 설명하자면 1번 라인에 있는 Post() 메서드는 사용자들이 메시지를 보내면 해당 호출되는 메서드로 파라미터로 넘겨받는 activity는 사용자들의 활동을 의미한다. Activity 안에서는 사용자가 보내는 메시지도 포함되어 있는데 activity.Text 속성에 사용자가 보낸 메시지가 들어 있다. 7번 라인은 사용자가 보낸 메시지의 내용이 있을 경우 length에 사용자가 보낸 메시지의 길이를 정수형으로 저장한다. 10번 라인은 사용자가 보낸 메시지의 내용과 길이를 바탕으로 회신을 생성하고 있다. 만약 사용자들이 'Hello'를 입력했다면 회신은 'You send Hello which was 5 characters'라는 회신이 생성된다. 그리고 마지막으로 11번 라인에서 회신을 보내고 있다. 이 샘플을 실행해 보기 위해서는 상단의 실행 버튼을 누르거나 혹은 F5키를 눌러서 실행시켜 보면 된다.

1. 봇 에뮬레이터를 이용한 테스트

기본적으로 챗봇 개발은 웹 프로그래밍이기 때문에 실행을 시키면 웹 페이지가 실행된다. 이때 보여지는 웹 페이지의 내용은 default.htm의 내용이 그냥 출력되기 중요한 내용이 아니다. 그런데 이렇게 웹페이지만 덩그러니 출력되기 때문에 테스트 할 수 있는 방법이 없다.이럴 때 사용하는 것이 그림 12의 봇 에뮬레이터이다.

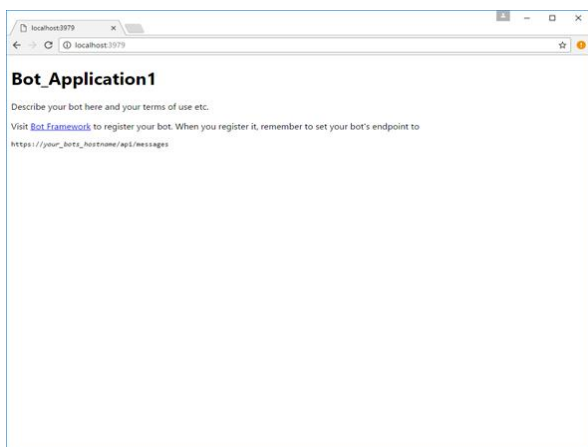


Fig. 12. 기본 샘플의 실행 모습

봇 에뮬레이터는 메신저를 대신해서 챗봇에 연결해서 테스트를 해줄 수 있는 도구로 봇 개발에 있어서 상당한 이점을 제공해 주고 있다.

봇 에뮬레이터를 실행하고 나면 먼저 연결할 주소를 입력해야 한다. 연결할 주소는 챗봇을 실행 시켰을 때 뜬 웹 브라우저의 주소와 포트 번호를 포함해서 http://localhost:3939/api/messages와 같은 형식으로 입력하고 CONNECT 버튼을 누른다. 이때 Microsoft App ID와 Microsoft App Password는 지금은 비워 놓는다.

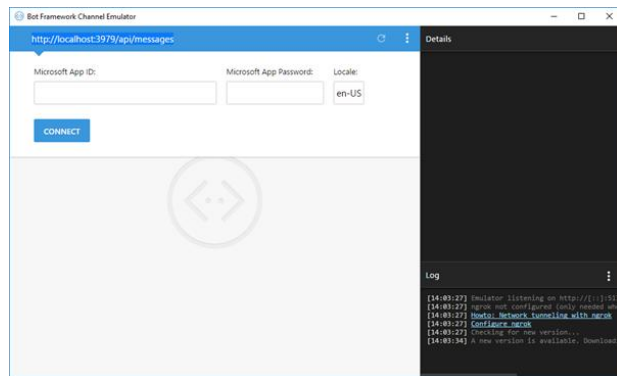


Fig. 13. 봇 에뮬레이터의 실행 모습

별다른 문제가 없다면 오른쪽 로그 창에 200번과 함께 메시지가 뜨면서 접속에 성공했다고 메시지가 나온다. 이제 직접 메시지를 입력해 볼 차례이다. 채팅창에 'Hello'를 입력하고 보내기 버튼을 클릭하면 즉각 회신이 오는 것을 볼 수 있다.

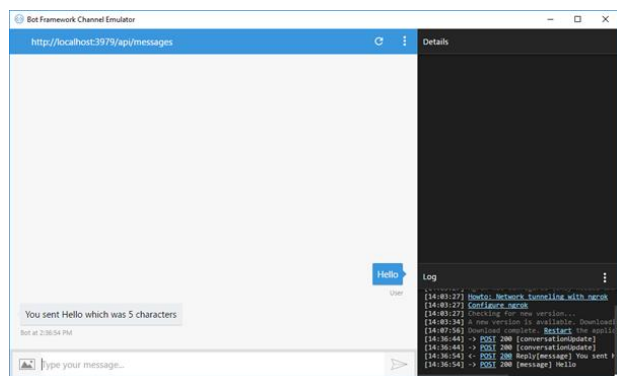


Fig. 14. 봇 에뮬레이터에서 'Hello' 가 실행된 모습.

봇과 관련한 서비스들은 계속해서 발전하고 있다. 자연어를 처리하는 것은 물론이고 각종 인지(Cognitive) 서비스와 연동되기도 하면서 빠르게 발전되고 있다. Microsoft 의 봇 프레임워크도 올 해 5월에 있을 Build 2018 개발자 컨퍼런스에서 좀 더 향상된 새로운 버전이 소개 될 것으로 기대된다.

REFERENCES

- [1] <http://themobileye.com/>
- [2] <https://www.skyscanner.co.kr/>
- [3] <https://dev.botframework.com/>
- [4] <http://yellowid.kakao.com/>
- [5] <https://docs.botframework.com/en-us/download/>

저 자 소 개



김 영 옥
 1996: 동의과학대학교
 전자통신과.
 2016: 세종사이버대학교
 컴퓨터공학과 공학사.
 현 재: 한국마이크로소프트
 DX부서 부장.
 관심분야:
 Chatbot, Cognitive Service,
 Natural Language Process.



이 강 호
 1986: 중앙대학교
 전자공학과 공학석사.
 1991: 중앙대학교
 전자공학과 공학박사
 현 재: 국립한국복지대학교
 컴퓨터정보보안과 교수
 관심분야: 정보보안, Chatbot,