

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM**  
**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN 2**

# **TÌM HIỂU**

## **CÁC CHATBOT FRAMEWORK**

*Người hướng dẫn:* **ThS. VŨ ĐÌNH HỒNG**

*Người thực hiện:* **NGUYỄN TRÍ QUANG - 51303135**

**NGUYỄN ĐẠI THỊNH – 51303403**

Lớp : **13050303**

Khóa : **17**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2017**

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM**  
**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN 2**

# **TÌM HIỂU**

## **CÁC CHATBOT FRAMEWORK**

*Người hướng dẫn:* **ThS. VŨ ĐÌNH HỒNG**

*Người thực hiện:* **NGUYỄN TRÍ QUANG - 51303135**

**NGUYỄN ĐẠI THỊNH – 51303403**

Lớp : **13050303**

Khóa : **17**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2017**

## LỜI CẢM ƠN

Tại trường Trường ĐH Tôn Đức Thắng, được sự chỉ bảo và giảng dạy nhiệt tình của quý thầy cô, đặc biệt là quý thầy cô khoa **Công Nghệ Thông Tin** đã truyền đạt cho em những kiến thức về lý thuyết và thực hành trong suốt thời gian học ở trường. Cùng với sự nỗ lực của nhóm, tụi em đã hoàn thành bài đồ án.

Từ những kết quả đạt được này, em xin chân thành cảm ơn Quý thầy cô trường Trường ĐH Tôn Đức Thắng, đã truyền đạt cho em những kiến thức bổ ích trong thời gian qua. Đặc biệt, là thầy **Th.S Vũ Đình Hồng** đã tận tình hướng dẫn em hoàn thành tốt báo cáo đồ án 2 này.

Do kiến thức còn hạn hẹp nên không tránh khỏi những thiếu sót trong cách hiểu, lỗi trình bày. Tụi Em rất mong nhận được sự đóng góp ý kiến của quý thầy cô để báo cáo tốt nghiệp đạt được kết quả tốt hơn.

## **ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của chúng tôi và được sự hướng dẫn của **Th.S Vũ Đình Hồng**. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong bài thuyết trình còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào chúng tôi xin hoàn toàn chịu trách nhiệm về nội dung bài thuyết trình của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày tháng năm*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Nguyễn Trí Quang*

*Nguyễn Đại Thịnh*

## **PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN**

### **Phần xác nhận của GV hướng dẫn**

---

---

---

---

---

---

---

Tp. Hồ Chí Minh, ngày      tháng      năm  
(kí và ghi họ tên)

### **Phần đánh giá của GV chấm bài**

---

---

---

---

---

---

---

Tp. Hồ Chí Minh, ngày      tháng      năm  
(kí và ghi họ tên)

# MỤC LỤC

<b>LỜI CẢM ƠN.....</b>	<b>3</b>
<b>PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN .....</b>	<b>5</b>
<b>CHƯƠNG 1: TỔNG QUAN VỀ CHATBOT .....</b>	<b>1</b>
<b>1.1 Khái niệm: .....</b>	<b>1</b>
1.1.1 Khái niệm về framework:.....	1
1.1.2 Khái niệm về chatbot:.....	1
1.1.3 Giới thiệu về bot framework:.....	2
1.1.4 Cấu trúc của bot framework: .....	3
1.1.5 Bot Framework nhắm tới đối tượng nào? .....	4
1.1.6 Giới thiệu Microsoft Cognitive Services: .....	5
1.1.6.1 Vision api: .....	6
1.1.6.2 Speech api: .....	6
1.1.6.3 Language api: .....	7
1.1.6.4 Knowledge api: .....	8
1.1.6.5 Search api: .....	9
<b>CHƯƠNG 2: BẮT ĐẦU XÂY DỰNG CHATBOT ĐƠN GIẢN .....</b>	<b>11</b>
<b>2.1 Những thứ cần chuẩn bị:.....</b>	<b>11</b>
<b>2.2 Các bước thực hiện:.....</b>	<b>11</b>
Bước 1: Cài đặt Git,Node js và tạo tài khoản.....	11
Bước 2: Tạo project .....	12
Bước 3: Tạo chatbot.....	15
Bước 4: Nhúng bot vào ứng dụng.....	16
Bước 5: Tạo server .....	17
Bước 6: Luyện bot.....	20
a) Tạo 1 application.....	20
b) Dạy ngôn ngữ.....	23
c) Train bot .....	25
d) Test chatbot .....	29
<b>CHƯƠNG 3: GIỚI THIỆU CHATBOT DHA .....</b>	<b>35</b>
<b>Tài liệu tham khảo:.....</b>	<b>51</b>
<b>Bảng đánh giá mức độ hoàn thành của sinh viên: .....</b>	<b>51</b>
<b>Phụ lục .....</b>	<b>52</b>



# CHƯƠNG 1: TỔNG QUAN VỀ CHATBOT

## 1.1 Khái niệm:

### 1.1.1 Khái niệm về framework:

- Framework là một thư viện các lớp đã được xây dựng hoàn chỉnh, bộ khung để phát triển các Phần mềm ứng dụng. Có thể ví framework như tập các “Vật liệu” ở từng lĩnh vực cho người lập trình viên, thay vì họ phải mất nhiều thời gian để tự thiết kế trước khi dùng. Do vậy, người lập trình viên chỉ cần tìm hiểu và khai thác các vật liệu này rồi thực hiện ( tức lập trình ) để gắn kết chúng lại với nhau, tạo ra sản phẩm.

### 1.1.2 Khái niệm về chatbot:

- Nói một cách đơn giản, nó là một phần của phần mềm mà bạn sẽ chat với nó, để làm điều gì đó hoặc chỉ để giải trí.
- Hãy nghĩ về nó như một sự thay thế cho tất cả các ứng dụng mà bạn đã tải xuống. Thay vì phải mở ứng dụng thời tiết để xem nhiệt độ, bạn có thể hỏi chatbot này và nó sẽ nói cho bạn thời tiết như thế nào.
- Cuối cùng, một chatbot có thể trở thành trợ lý ảo của riêng bạn để chăm sóc tất cả mọi thứ, từ việc gọi một chiếc Uber cho bạn cho đến thiết lập một cuộc hẹn. Hoặc Facebook Messenger hay một nền tảng nào khác sẽ cho phép một loạt các chatbot khác nói chuyện với bạn về bất cứ điều gì có liên quan. Ví dụ, một chatbot từ Southwest Airlines có thể cho bạn biết chuyến bay của bạn bị hoãn, một chatbot khác từ FedEx có thể cho bạn biết kiện hàng của bạn đang trên đường chuyển đi, và nhiều điều khác nữa.
- Thực ra, những chatbot này đã xuất hiện từ những năm 1960. Nếu bạn sử dụng AIM hay nền tảng chat khác, có thể bạn đã từng trò chuyện với chatbot như Smarter Child. Nhưng lúc đó chúng thực sự chỉ có mục đích giải trí, chứ không có tác dụng gì với doanh nghiệp. Và giờ những công ty như Facebook hay Microsoft đang làm chúng ta nghĩ khác đi về chúng.



- Dù chatbot có thể đang bị thổi phồng về tiềm năng, nhưng nó thực sự có thể thay đổi một cách căn bản cách chúng ta sử dụng điện thoại và máy tính.

### **1.1.3 Giới thiệu về bot framework:**

- Tại hội nghị Build 2016 của mình diễn ra từ ngày 30/3 đến hết ngày 1/4 vừa qua ở San Francisco, Microsoft đã đưa ra 7 thông báo lớn, quan trọng và mang tầm chiến lược nhất của mình. Và 2 trong 7 thông báo đó có liên quan đến Chatbot:
  - Công bố dự án Bot Framework.
  - Skype thế hệ mới sẽ có khả năng giao tiếp với Bot.
- Qua đó, các bạn cũng có thể thấy được phần nào tầm quan trọng của chatbot đối với tương lai của ngành công nghệ. Không chỉ Microsoft mà các ông lớn khác trong ngành như Apple (Siri), Amazon (Alexa), Google (Google Now), Facebook (Messenger Bots) đều đang tham gia vào cuộc đua tranh nhau vị trí số 1 về Chatbot.
- Tìm hiểu về Chatbot chưa bao giờ là chậm vì cả thế giới đều đang tìm hiểu và phát triển nó ngày một tốt hơn. Vậy tại sao chúng ta không đánh vào những thứ mới mẻ này thay vì những dịch vụ đã thâm căn cố đế khó có thể lay chuyển của những tập đoàn lớn mạnh? Liệu Chatbot có phải là một cơ hội để FPT nói riêng và Việt Nam nói chung vươn mạnh ra thị trường quốc tế.
- Vậy còn chần chờ gì nữa, chúng ta sẽ bắt tay luôn vào việc tìm hiểu Chatbot cũng như những dịch vụ của Microsoft sẽ giúp chúng ta phát triển Chatbot một cách dễ dàng hơn. Ở bài viết này, mình xin phép giới thiệu đi sâu về Microsoft Bot Framework.

#### 1.1.4 Cấu trúc của bot framework:

- Bot Framework là 1 nền tảng gồm các công cụ và tài nguyên nhằm giúp cho các nhà phát triển có thể tự xây dựng Chatbot của riêng mình. Bot Framework bao gồm 3 thành phần chính sau:
  - **Bot Builder SDK:** Bộ công cụ được chia sẻ trên GitHub dưới dạng mã nguồn mở theo giấy phép MIT này sẽ cung cấp tất cả mọi thứ bạn cần để xây dựng Chatbot bằng Node.js, .NET hay REST API.
  - Sử dụng SDK này Chatbot có thể tương tác với người dùng không chỉ bằng văn bản, mà còn cả gửi/nhận file đính kèm, sử dụng hình ảnh để tương tác... Thậm chí bây giờ SDK còn hỗ trợ cả Skype Call.
  - **Bot Framework Developer Portal:** Cho phép bạn kết nối Chatbot của mình tới Skype, Slack, Facebook Messenger, Kik, Office 265 mail và những kênh giao tiếp thông dụng khác. Đơn giản chỉ việc đăng ký bot, chọn kênh giao tiếp và publish bot lên Bot Directory.
  - **Bot Directory:** Đây là một thư mục cộng đồng chứa tất cả những bot đã được kiểm duyệt đăng ký thông qua Developer Portal. Người dùng có thể trải nghiệm, khám phá những bot có sẵn hoặc thêm bot mà mình phát triển vào Bot Directory.
- Kênh giao tiếp Web Chat và Skype đã được cấu hình mặc định khi bạn tạo Chatbot. Cho mỗi kênh giao tiếp mới (Facebook Messenger, Slack...), bạn cần phải tự cấu hình.

### **1.1.5 Bot Framework nhắm tới đối tượng nào?**

- Đối tượng mà Bot Framework nhắm tới là bất kỳ ai muốn tạo nên một Chatbot cho riêng mình. Bạn không nhất thiết phải giỏi thuật toán, phải biết về xử lý ngôn ngữ tự nhiên mới viết được một Chatbot thông minh.
- Khi tự mình viết Chatbot, người phát triển sẽ phải đối mặt với những vấn đề sau: Chatbot cần phải hiểu được ngôn ngữ và có kỹ năng đàm thoại; chúng cần phải hiệu quả, linh hoạt và có khả năng mở rộng; và chúng phải kết nối được với người dùng – lý tưởng nhất là ở bất kỳ ngữ cảnh và ngôn ngữ nào. Bot Framework cung cấp những công cụ hỗ trợ người phát triển giải quyết các vấn đề này, do đó chúng ta chỉ cần tập trung vào phát triển nghiệp vụ của Chatbot.
- Khi sử dụng Bot Framework, bạn sẽ có những lợi thế sau:
  - Viết một Chatbot đồng thời có thể chạy trên những nền tảng khác nhau: Skype, Facebook Messenger, Slack...
  - Được cung cấp Embedded Web Chat Control (bạn có thể nhúng Chatbot này vào luôn trang web của mình mà không cần tốn công code).
  - Direct Line API: REST API giúp bạn có thể thêm Chatbot của mình vào bất kỳ ứng dụng nào, không giới hạn về ngôn ngữ và nền tảng.
  - Cuối cùng là phần quan trọng nhất: Làm cho Chatbot của bạn trở nên thông minh hơn bằng Cognitive Services như LUIS cho việc xử lý ngôn ngữ tự nhiên, Translator giúp tự động phiên dịch qua hơn 30 ngôn ngữ khác nhau...

### 1.1.6 Giới thiệu Microsoft Cognitive Services:

- **Microsoft Cognitive Services** bao gồm một bộ các API ứng dụng trí tuệ nhân tạo thông minh, cho phép lập trình viên ở mọi cấp độ từ những bạn sinh viên viết ứng dụng đầu tiên của mình hay những lập trình viên chuyên nghiệp làm việc cho những công ty, tổ chức lớn đều có thể tạo ra được những ứng thông minh hơn một cách dễ dàng.
- Các API của Cognitive Services được viết dưới dạng REST API do vậy lập trình viên có thể tích hợp các API này trên nhiều nền tảng khác nhau như iOS, Android, hay Windows, chỉ cần có kết nối Internet.



HÌNH 1

- Tính đến thời điểm viết bài viết này, Microsoft Cognitive Services bao gồm 21 API được chia thành 5 nhóm: **Vision, Speech, Language, Knowledge** và **Search**.

#### 1.1.6.1 Vision api:

- Là nhóm API liên quan đến xử lý hình ảnh, trong nhóm Vision API này hiện Microsoft cung cấp 4 API sau:
  - **Computer Vision API:** API này cho phép trích xuất những thông tin có giá trị từ bức ảnh của bạn với khả năng xác định được kiểu đối tượng trong ảnh (là bánh mỳ, con chó, con mèo hay cây cối, ...) hay nếu là người thì API này cũng xác định được giới tính của nhân vật trong ảnh. Ngoài ra, API này cũng hỗ trợ nhận diện được những nhận vật nổi tiếng hay trích xuất chữ có trong bức hình của bạn.
  - **Face API:** Cái tên nói lên tất cả, đây là API cho phép phát hiện khuôn mặt có trong bức hình của bạn. Ngoài ra, API này cũng trả về các thuộc tính của khuôn mặt như tuổi, giới tính, độ rạn ngòai của nụ cười hay thậm chí là chiều dài của tóc, ... Ngoài phát hiện khuôn mặt, Face API còn cho phép so sánh 2 khuôn mặt có phải là của cùng một người hay không.
  - **Emotion API:** API này cho phép xác định tâm trạng của người có trong bức hình xem họ đang vui, đang buồn hay đang giận dữ.
  - **Video API:** API này là một tập hợp các thuật toán xử lý video tân tiến của Microsoft. Với Video API, các nhà phát triển có thể tích hợp các tính năng chỉnh sửa video bao gồm chống rung, phát hiện khuôn mặt người, phát hiện chuyển động hay tạo video thumbnail.

#### 1.1.6.2 Speech api:

- Là nhóm API liên quan đến xử lý âm thanh với công nghệ xử lý đến từ Bing. Trong nhóm Speech API này hiện Microsoft cung cấp 3 API:

- **Bing Speech API:** API này cho phép trích xuất một tập tin âm thanh sang dạng chữ, chuyển đổi định dạng chữ sang âm thanh (tức là đọc chữ) hay đoán ý của một câu nói.
- **Custom Recognition Intelligent Service (CRIS):** CRIS cho phép bạn có thể tùy biến language model và acoustic model sao cho phù hợp với ứng dụng hoặc người dùng của bạn.
- **Speaker Recognition API:** Với những thuật toán nhận dạng giọng nói tân tiến của Microsoft, API này cho phép nhận dạng giọng nói của người nói trong một tập tin âm thanh. API này bao gồm 2 thành phần: speaker verification và speaker identification tạm dịch tương ứng là xác nhận người nói và xác định người nói. **Speaker Verification** cho phép xác nhận và xác thực người dùng bằng giọng nói của họ. Lập trình viên chỉ cần cho người dùng đọc một đoạn văn bản có sẵn để lưu lại dữ liệu giọng nói của họ (enrollment) rồi ở mỗi lần cần xác thực, người dùng chỉ cần đọc lại chính xác đoạn văn bản đã được dùng để lấy dữ liệu giọng nói của mình (bước enrollment) để xác thực. **Speaker Identification** có thể xác định được người đang nói trong một tập tin âm thanh dựa trên một tập dữ liệu các người nói tiềm năng. Tính năng này cũng có thể được sử dụng để xác thực người dùng bằng giọng nói. Tuy nhiên thay vì phải đọc chính xác một đoạn văn bản cố định như Speaker Verification thì khi sử dụng tính năng Speaker Identification này, người dùng có thể đọc một đoạn văn bản bất kỳ, API sẽ phân tích và đối chiếu với tập dữ liệu giọng nói của người dùng để so sánh và xác thực.

#### 1.1.6.3 Language api:

- Là nhóm API liên quan đến xử lý ngôn ngữ. Nhóm Language API này hiện được Microsoft cung cấp 5 API:
  - **Bing Spell Check API:** API này cho phép phát hiện và sửa các lỗi chính tả có trong một đoạn văn bản mà bạn cung cấp. API còn có khả năng phát hiện từ lóng, sửa lỗi tên riêng hay sửa các từ đồng âm...

- **Web Language Model API:** API này giúp hỗ trợ xử lý ngôn ngữ tự nhiên, với khả năng chèn khoảng cách vào 1 đoạn văn bản được viết liền nhau như hashtag hay đường dẫn.
- **Linguistic Analysis API:** The Linguistic Analysis API giúp bạn hiểu sâu hơn văn bản của mình. API này sẽ giúp phân tích cú pháp của ngôn ngữ tự nhiên để dễ dàng xác định được các thực thể (danh từ) hay các hành động (động từ) có trong văn bản. Việc xử lý văn bản này có thể hữu ích cho các công việc phân tích như phân tích tâm lý.
- **Language Understanding Intelligent Service (LUIS):** LUIS cho phép lập trình viên xây dựng các model hiểu được ngôn ngữ tự nhiên cũng như hiểu được các câu lệnh riêng được thiết kế riêng cho ứng dụng của bạn. Ví dụ: Bạn có thể nói “bật đèn trong phòng ngủ”, gửi câu lệnh đó tới LUIS model, và thay vì trả lại chính xác các câu từ có trong câu lệnh trên, LUIS sẽ trả về dữ liệu chứa: thông tin hành động là “bật” vị trí là “phòng ngủ” và đối tượng hướng đến là “bóng đèn”, từ đó ứng dụng có thể dễ dàng xử lý được câu lệnh của bạn.
- **Text Analytics API:** API này giúp xác định các ẩn ý, từ khóa, chủ đề hay ngôn ngữ được sử dụng có trong một đoạn văn bản.

#### 1.1.6.4 Knowledge api:

- Là nhóm API liên quan đến tri thức. Nhóm Knowledge API này hiện bao gồm 4 API:
  - **Academic Knowledge API:** API này cho phép lập trình viên xây dựng những giải pháp tìm kiếm tài liệu học thuật với tính năng Interpret, trả về kết quả gợi ý cho từ khóa mà người dùng nhập vào dựa vào nguồn dữ liệu phong phú từ hệ thống Microsoft Academic Graph (MAG).

- **Knowledge Exploration Service API:** API này cho phép lập trình viên xây dựng những giải pháp tìm kiếm sử dụng ngôn ngữ tự nhiên bằng cách dịch ngôn ngữ tự nhiên mà người dùng nhập vào sang các biểu thức truy vấn có cấu trúc mà máy tính có thể dễ dàng hiểu và xử lý được.
- **Entity Linking Intelligence Service API:** Với một đoạn văn bản, Entity Linking Intelligence Service sẽ nhận dạng và xác định từng thực thể (entity) có trong đoạn văn dựa vào ngữ cảnh của đoạn văn đó và sẽ liên kết những entity này tới Wikipedia. Lấy ví dụ rằng bạn có một đoạn văn bản trong đó chứa từ cloud, từ cloud này có thể hiểu sang thành “Cloud Computing” (điện toán đám mây) hay “Cloud” (đám mây trên trời), dựa vào ngữ cảnh mà API này sẽ xác định được rằng từ cloud có ý nghĩa như thế nào.
- **Recommendations API:** API này cho phép xây dựng các giải pháp khuyến nghị cho người dùng. Chẳng hạn như bạn xây dựng một ứng dụng bán hàng, sử dụng API này cho phép bạn dễ dàng xây dựng ra các tính năng khuyến nghị mua hàng như “Các sản phẩm được bán chạy”, “Các sản phẩm được mua cùng” hay “Những sản phẩm hàng đầu trong mặt hàng Đồ gia dụng” chẳng hạn, từ đó sẽ khuyến khích người dùng mua nhiều hơn.

#### 1.1.6.5 Search api:

- Search API cũng cấp các giải pháp tìm kiếm sử dụng sức mạnh của Bing.com cũng như từ nhiều đối tác như AOL, Apple, Amazon, Yahoo, .... Nhóm Search API bao gồm 5 API cho các tác vụ tìm kiếm trang web, hình ảnh, video hay tin tức:
  - **Bing Web Search API:** Đây là API chủ lực của gói Search API. Chỉ với một cú pháp lệnh gọi đến API này, lập trình viên có thể lấy được các kết quả trả về cho trang web, hình ảnh, video hay tin tức tương ứng. Nó khá tương tự với việc bạn tìm kiếm trên các công cụ tìm kiếm như Bing.com hay Google.com. Ngoài ra, lập trình viên cũng sẽ nhận được những tính năng mạnh mẽ từ công



cụ tìm kiếm Bing Search như ranking kết quả tìm kiếm, phân loại kết quả tìm kiếm theo vùng...

- **Bing Autosuggest API:** API này cho phép lập trình viên có thể xây dựng tính năng đề xuất các từ khóa tìm kiếm liên quan kể cả khi từ khóa tìm kiếm chưa được điền đầy đủ. Ví dụ nếu người dùng gõ từ khóa tìm kiếm là “Thời tiết tại H”, API sẽ trả về danh sách các từ khóa gợi ý như “Thời tiết tại Hà Nội”, “Thời tiết tại Hồ Chí Minh” hay “Thời tiết tại Hà Giang” chẳng hạn.
- **Bing Image Search API:** API này cho phép lập trình viên có thể tìm kiếm các hình ảnh tương ứng với từ khóa nhập vào. Ngoài trả về đường dẫn của hình ảnh, API này cũng trả về các metadata hữu ích như kích thước ảnh, màu chủ đạo của ảnh...
- **Bing Video Search API:** API này cho phép lập trình viên có thể tìm kiếm các video tương ứng với từ khóa nhập vào. Ngoài trả về đường dẫn của video, API này cũng trả về các metadata hữu ích khác như tên nhà sản xuất, định dạng mã hóa, ảnh thumbnail...
- **Bing News Search API:** API này cho phép lập trình viên có thể tìm kiếm các tin tức, bài báo tương ứng với từ khóa nhập vào. Ngoài ra, API cũng trả về các metadata hữu ích khác như thể loại, thông tin nhà xuất bản, ngày xuất bản...

Có thể thấy với bộ 21 API mà dịch vụ **Microsoft Cognitive Services** cung cấp, lập trình viên có thể thỏa sức xây dựng các ứng dụng tích hợp các tính năng thông minh sử dụng sức mạnh từ machine learning mà dịch vụ này mang lại một cách dễ dàng chỉ bằng việc gọi API và xử lý kết quả trả về với định dạng JSON.

## CHƯƠNG 2: BẮT ĐẦU XÂY DỰNG CHATBOT ĐƠN GIẢN

### 2.1 Những thứ cần chuẩn bị:

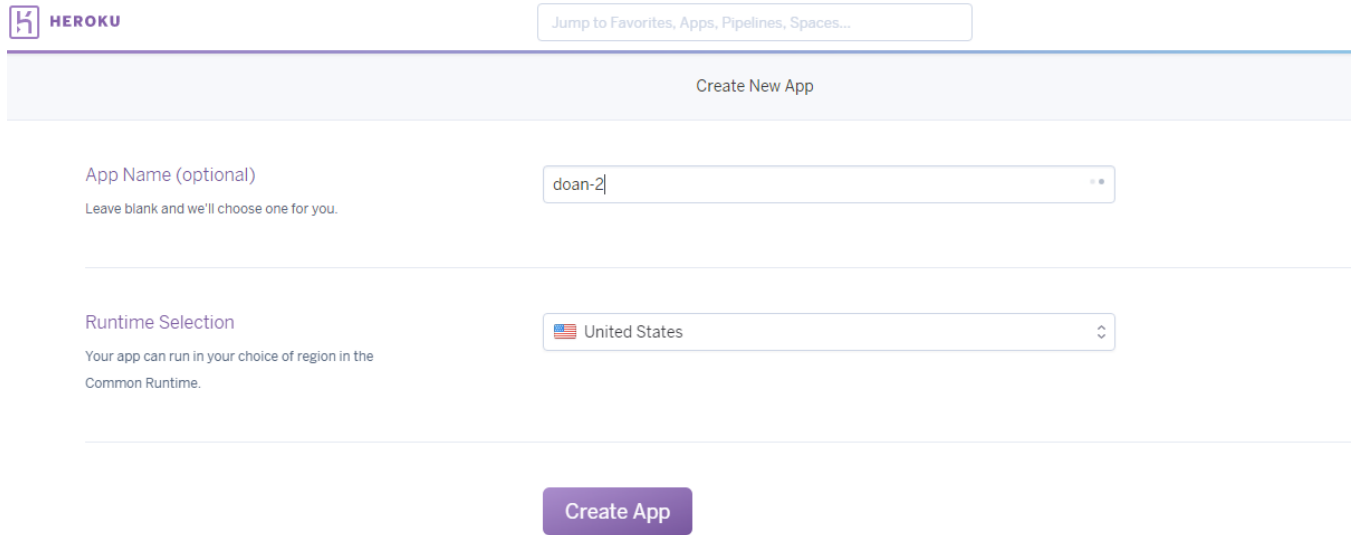
- **Git version control:** Đây là phần mềm quản lí phiên bản , nó cho phép bạn có thể upload project của bạn lên 1 kho chứa, điều tiện lợi của nó là những người làm việc trong 1 nhóm sẽ dễ dàng quản lí, chỉnh sửa các file mình làm chung trong project.
- **Tài khoản Heroku:** Nó là 1 dạng web cung cấp host cho những ai lập trình web. Khi bạn đăng kí tài khoản , nó sẽ tạo cho bạn 1 host để bạn đưa các file của mình lên và hoạt động.
- **Node js:** Là một nền tảng chạy javascript, để tạo 1 con ChatBot bạn cần phải chạy những gì bạn viết ở trên server, và Node js là một công cụ hỗ trợ tuyệt vời cho bạn.
- **Fanpage Facebook, Skype....:** Bạn hoàn toàn có thể nhúng con Bot của mình vào Messenger Facebook hay Skype, để làm được điều đó bạn cần tạo tài khoản của riêng mình.

### 2.2 Các bước thực hiện:

#### Bước 1: Cài đặt Git,Node js và tạo tài khoản

- Chúng ta tải về và cài đặt Git, Node js , các bạn có thể đọc hướng dẫn và cài đặt ở đường dẫn sau:
  - Git: <https://git-scm.com/download/win>
  - Node js: <https://nodejs.org/en/download/>
- Tạo tài khoản heroku trang chủ: <https://www.heroku.com/>

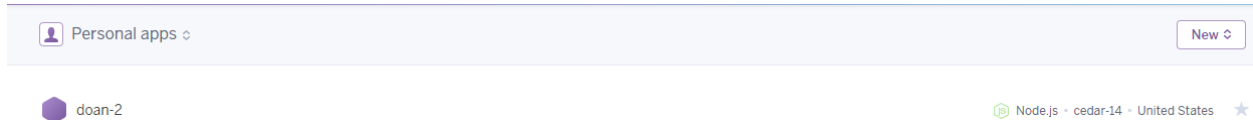
- Sau khi tạo tài khoản, chọn vào create a new app và tạo project cho mình:



The screenshot shows the Heroku 'Create New App' page. At the top left is the Heroku logo. A search bar at the top right contains the text 'Jump to Favorites, Apps, Pipelines, Spaces...'. Below the header is a light blue bar with the text 'Create New App'. The main form has two sections. The first section is 'App Name (optional)' with a subtext 'Leave blank and we'll choose one for you.' and a text input field containing 'doan-2'. The second section is 'Runtime Selection' with a subtext 'Your app can run in your choice of region in the Common Runtime.' and a dropdown menu showing 'United States' with a flag icon. At the bottom of the form is a purple button labeled 'Create App'.

HÌNH 2

- Sau khi tạo:



The screenshot shows the Heroku dashboard. At the top is a navigation bar with 'Personal apps' and a 'New' button. Below the navigation bar, the app 'doan-2' is listed with a purple icon. To the right of the app name, the runtime is specified as 'Node.js - cedar-14 - United States' with a star icon.

HÌNH 3

## Bước 2: Tạo project

- Tiếp theo, bạn vào thư mục mà mình tạo project sử dụng máy tính, click chuột phải chọn Git bash here. Nó sẽ ra giao diện command line, bạn gõ “Git init” để chọn thư mục hiện tại là repository (kho chứa) của Git.

```
Dai Thinh@Sakya MINGW64 /f/DoAn2/doan_2 (master)
$ git init
```

- Cũng ở thư mục đó, bạn nhấn Shift + chuột phải rồi chọn Open command window here. Sau đó login tài khoản Heroku bạn đã tạo.

```
F:\DoAn2\doan_2>heroku login
Enter your Heroku credentials:
Email: daithinh2401@gmail.com
Password: *****
Logged in as daithinh2401@gmail.com
```

- Tạo 1 project node js. Gõ vào **npm init**:

```
MINGW64:/f/DoAn2/doan_2
Dai Thinh@Sakya MINGW64 /f/DoAn2/doan_2 (master)
$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg> --save` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
name: (doan_2)
version: (1.0.0)
description:
git repository:
keywords:
license: (ISC)
About to write to F:\DoAn2\doan_2\package.json:
{
  "name": "doan_2",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Dai Thinh",
  "license": "ISC",
  "dependencies": {
    "botbuilder": "^3.7.0"
  },
  "devDependencies": {},
  "description": ""
}

Is this ok? (yes) yes
```

- Gõ các câu lệnh sau để cài đặt module cho project:
  - Npm install restify
  - Npm install botbuilder
- Sau đó ở thư mục project bạn tạo 1 file “app.js”, file sẽ có các phần như sau:

```
var restify = require('restify');
var builder = require('botbuilder');
```

- Khai báo các module bạn đã install:

```
// Setup Restify Server
var server = restify.createServer();
server.listen(process.env.port || process.env.PORT || 3978, function () {
  console.log('%s listening to %s', server.name, server.url);
});
```

- Cài đặt restify server:

```
// Create chat bot
var connector = new builder.ChatConnector({
  appId: process.env.MICROSOFT_APP_ID,
  appPassword: process.env.MICROSOFT_APP_PASSWORD
});
var bot = new builder.UniversalBot(connector);
server.post('/', connector.listen());
```

### Bước 3: Tạo chatbot

- Sử dụng thư viện của Microsoft bot framework là ChatConnector để tạo một chatbot mới và gán nó vào biến **connector**.
- Chạy bot bằng cách gọi **UniversalBot** truyền vào connector và gán nó cho biến bot.
- Đây được xem là bộ não của chatbot và sẽ điều khiển toàn bộ hoạt động của Chatbot, chẳng hạn như nếu muốn Chatbot trả lời tin nhắn , chỉ việc gọi **bot.dialog** và xử lý công việc cần thiết.
- Để code của ta được chạy trên server , bạn cần tạo thêm 1 file Procfile và thêm đoạn code sau để server biết và chạy file “ app.js” của bạn: **web: node index.js**.
- Sau khi hoàn tất các file , các bạn thực hiện deploy lên server Heroku đã tạo khi nãy, mở Git và gõ theo thứ tự:
  - Git add
  - Git commit –m “ First commit ”
  - Git push heroku master

```
Dai Thinh@Sakya MINGW64 /f/DoAn2/doan_2 (master)
$ git push heroku master
Counting objects: 854, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (776/776), done.
Writing objects: 100% (854/854), 970.50 KiB | 0 bytes/s, done.
Total 854 (delta 118), reused 0 (delta 0)
```

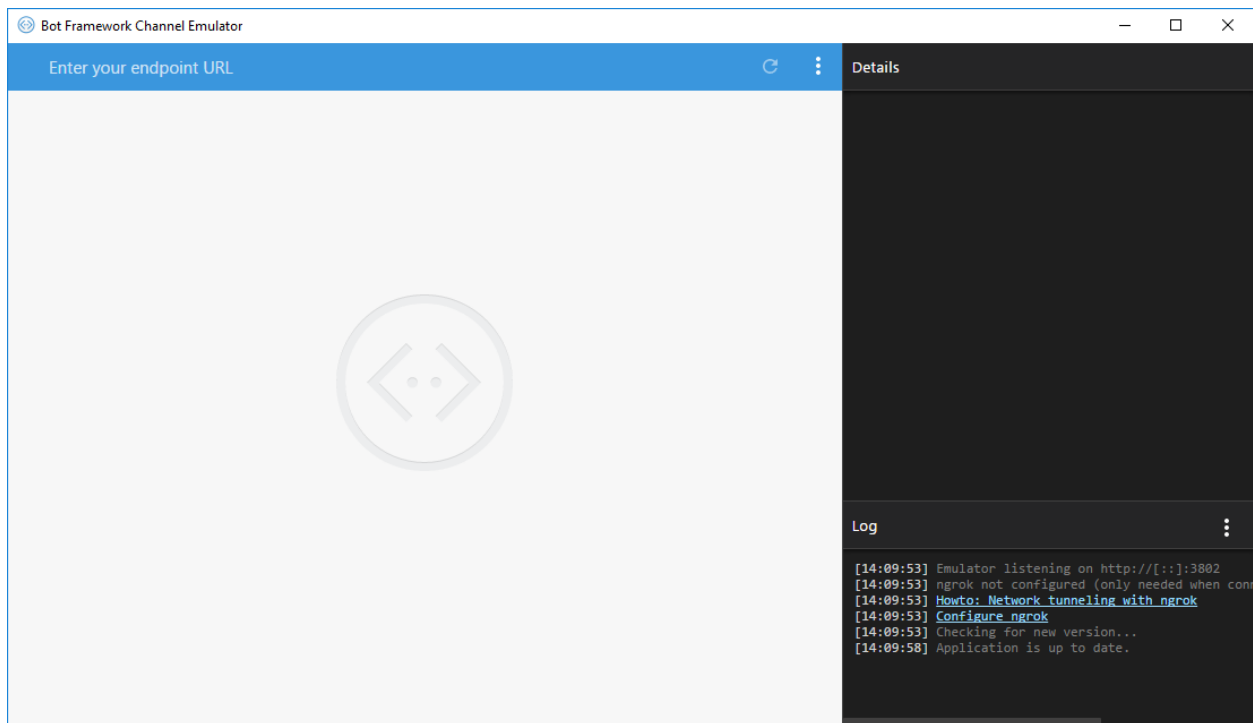
- Khi push thành công nó sẽ báo như thế này , đợi đến khi hiện ra dòng chữ màu thì mới xong hoàn toàn.
- Đây là những bước cơ bản để tạo một ChatBot, sau này, khi chỉnh sửa , bổ sung code để làm cho Bot trở nên thông minh hơn , ta chỉ việc sử dụng Git để push lên server là được.

#### Bước 4: Nhúng bot vào ứng dụng

- Và để nhúng Bot vào các ứng dụng , ta sử dụng **Bot Framework Developer Portal** của Microsoft, nó cho phép bạn kết nối Chatbot của mình tới Skype, Slack, Facebook Messenger, Kik, Office 265 mail và những kênh giao tiếp thông dụng khác. Đơn giản chỉ việc đăng ký bot, chọn kênh giao tiếp và publish bot lên Bot Directory. Chi tiết sẽ nằm ở phần sau.
- Bây giờ chúng ta thử cho con Bot trả lời một câu đơn giản **Hello World !** bằng cách thêm đoạn code sau vào :

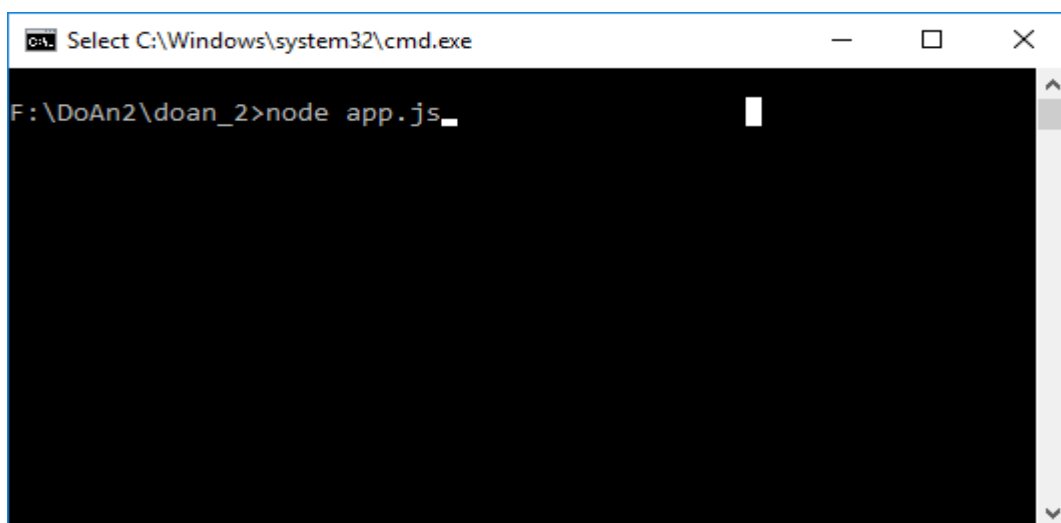
```
bot.dialog('/', function(session){
    session.send('Hello World !');
});
```

- Trong đó, **bot.dialog** dùng để gửi lại đoạn hội thoại mà chúng ta muốn con Bot trả lời, ở đây đơn giản chỉ là: Hello World !
- Để test Chatbot của mình trên localhost , ta có thể sử dụng phần mềm **Bot Framework Emulator**, download ở link : <https://docs.botframework.com/en-us/tools/bot-framework-emulator/>



## Bước 5: Tạo server

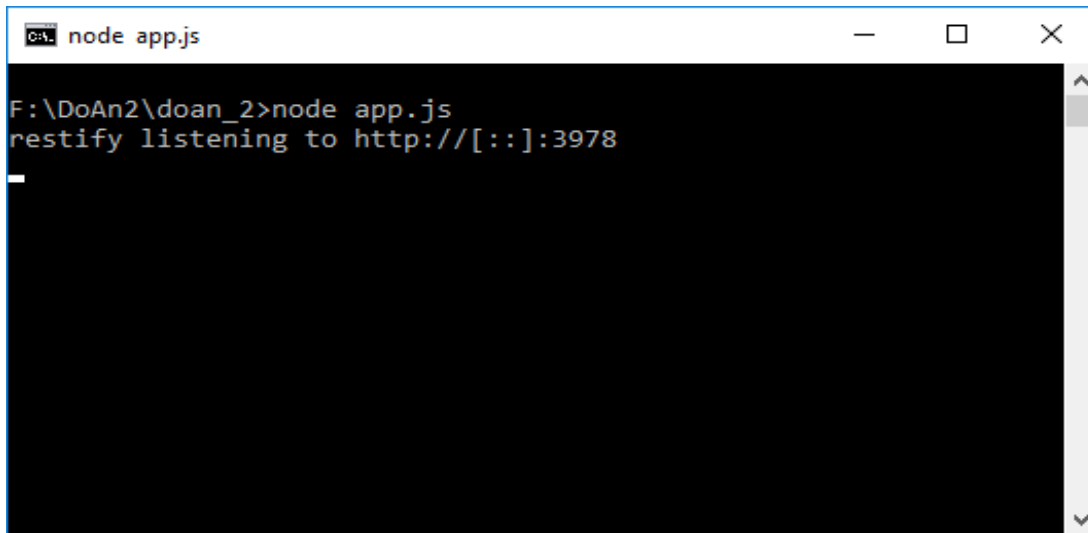
- Tiếp theo, bạn cần chạy file app.js trước để nó khởi tạo server , port , Chatbot và các thứ cần thiết đưa vào Localhost, mở giao diện Command line và gõ vào :



HÌNH 4 : GIAO DIỆN CỦA CHƯƠNG TRÌNH

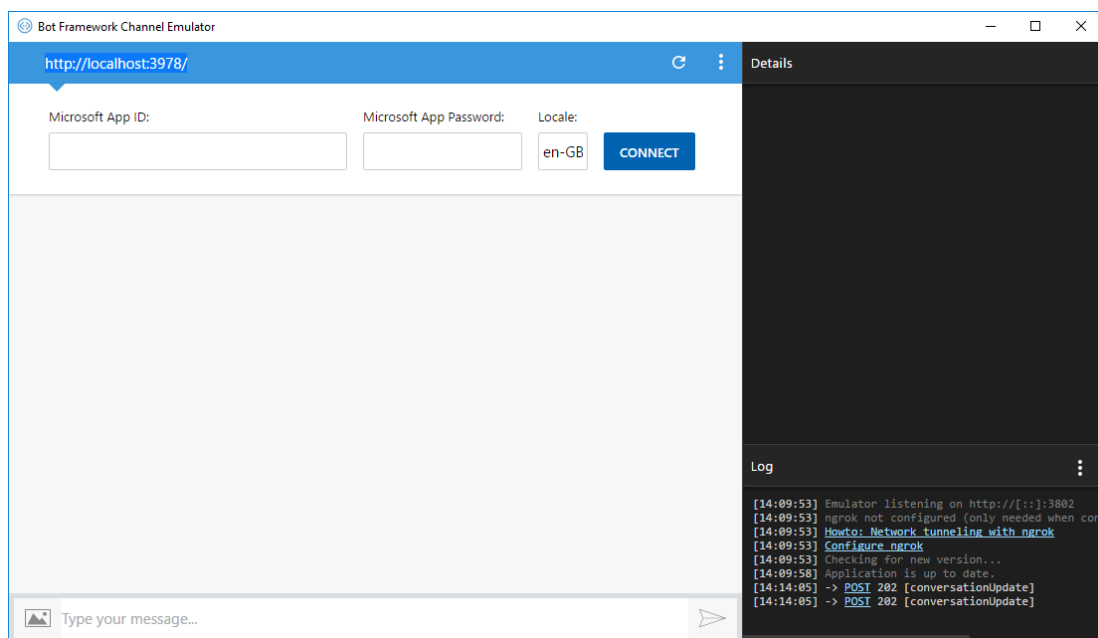


- Khi chạy thành công nó sẽ hiện như thế này:



```
node app.js
F:\DoAn2\doan_2>node app.js
restify listening to http://[::]:3978
```

- Sau đó ta bật Bot framework emulator, ở đường dẫn nhập vào đường dẫn sau và bấm connect: <http://localhost:3978/>

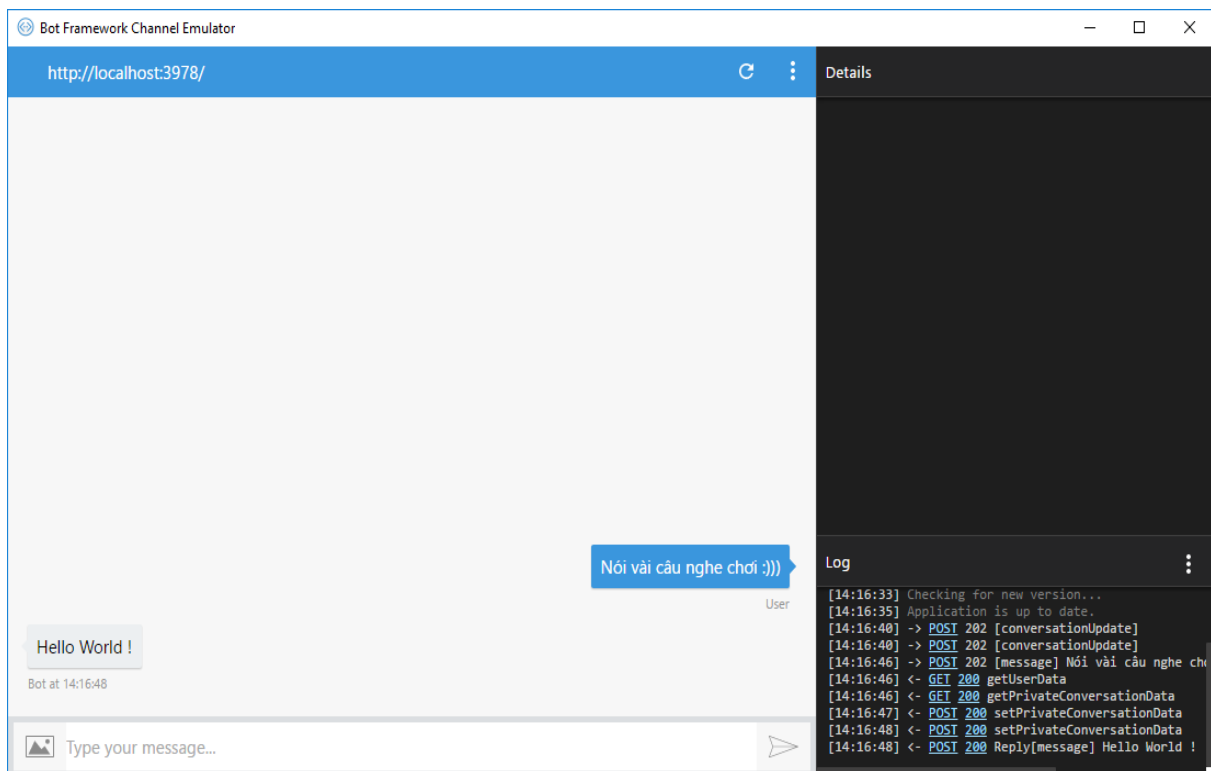


HÌNH 5

- Nếu như thành công , nó sẽ báo dòng chữ màu xanh, ngược lại nếu thất bại sẽ hiện thông báo lỗi màu đỏ:

```
[14:09:53] Emulator listening on http://[::]:3802
[14:09:53] ngrok not configured (only needed when con
[14:09:53] Howto: Network tunneling with ngrok
[14:09:53] Configure ngrok
[14:09:53] Checking for new version...
[14:09:58] Application is up to date.
[14:14:05] -> POST 202 [conversationUpdate]
[14:14:05] -> POST 202 [conversationUpdate]
```

- Thử chat với con Bot nào !!



HÌNH 6

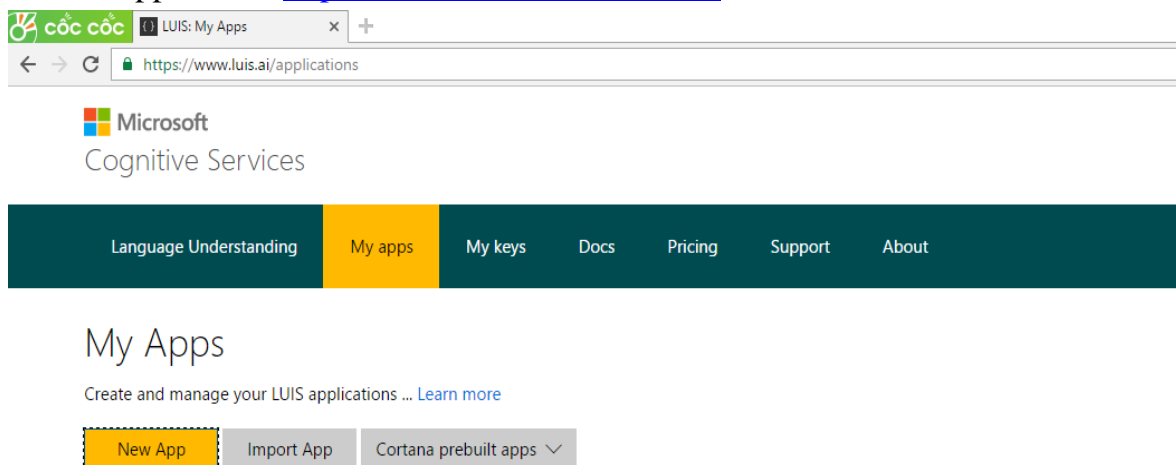
- Bot đã được sinh ra và chào cả thế giới...
- Quá đơn giản để có thể tạo 1 ChatBot tự động trả lời , nhưng vấn đề là không ai muốn tạo ra một con Bot chỉ để nói 1 câu duy nhất. Vậy thì , làm thế nào để con Bot trở nên thông minh hơn và có thể giúp ích cho con người, câu trả lời là sử dụng bộ 21 API mà Microsoft đã cung cấp được giới thiệu ở trên và áp dụng vào con Bot.

## Bước 6: Luyện bot

- Trước hết thử áp dụng 1 api để Bot hiểu được ngôn ngữ con người thông qua việc xử lý ngôn ngữ tự nhiên , đó là **Language Understanding Intelligent Service ( LUIS )** nằm trong bộ Language API của Microsoft.

### a) Tạo 1 application

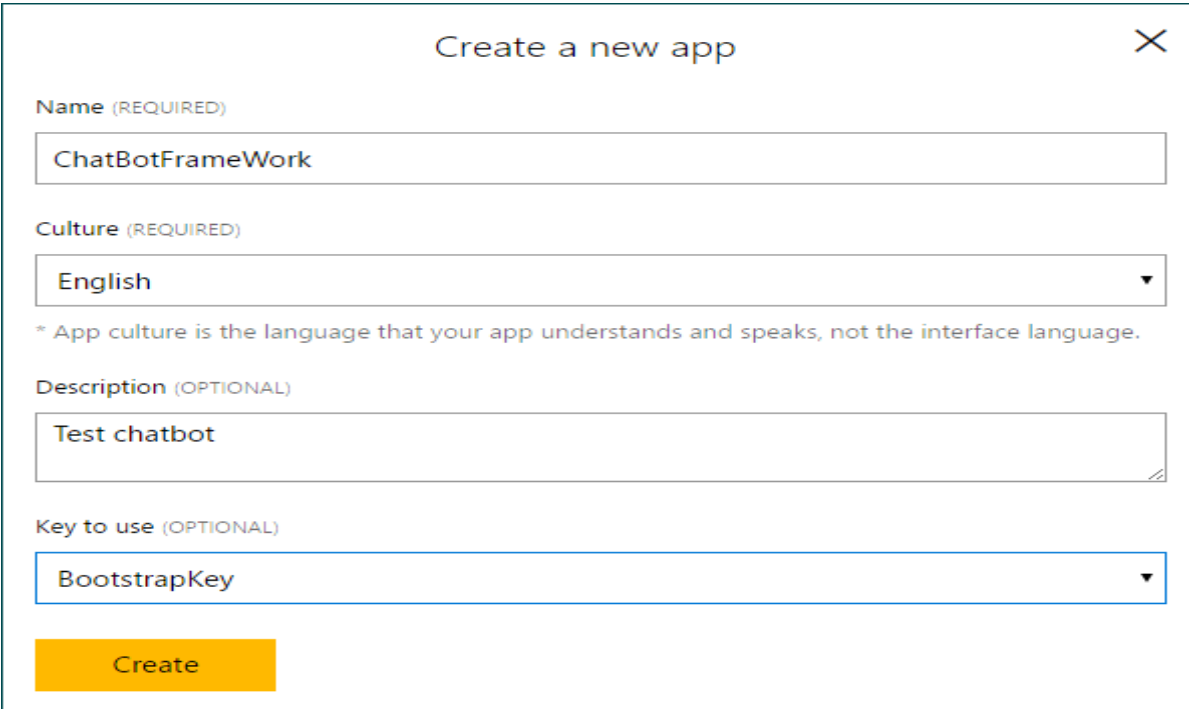
- Đầu tiên, ta vào trang chủ và đăng nhập tài khoản Microsoft rồi tạo cho mình 1 application: <https://www.luis.ai/home/index>



HÌNH 7

- Sau khi đăng nhập , chọn mục My apps và nhấn vào New Apps để tạo 1 app mới.

- Nhấn vào Create để tạo.
- Giao diện chính của LUIS , nó bao gồm các mục như Intents ( mục tiêu ) , Entities ( Thực thể )...



**Create a new app** ✕

**Name (REQUIRED)**

ChatBotFrameWork

**Culture (REQUIRED)**

English ▼

\* App culture is the language that your app understands and speaks, not the interface language.

**Description (OPTIONAL)**

Test chatbot

**Key to use (OPTIONAL)**

BootstrapKey ▼

**Create**

HÌNH 8



## ChatBotFram...

### Dashboard

Intents

Entities

Features

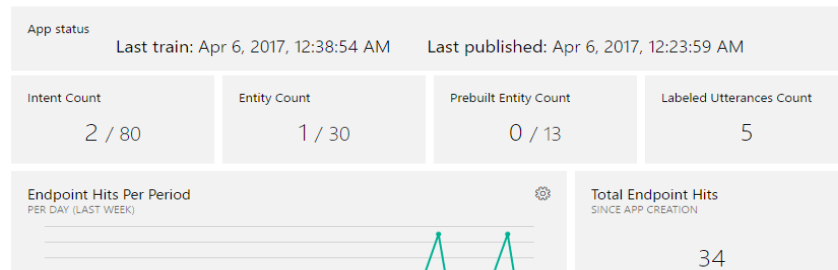
Train & Test

Publish App

← Back to App list

## Dashboard (App Id: 40c676a6-2240-4a6f-b55a-88fd87c3977)

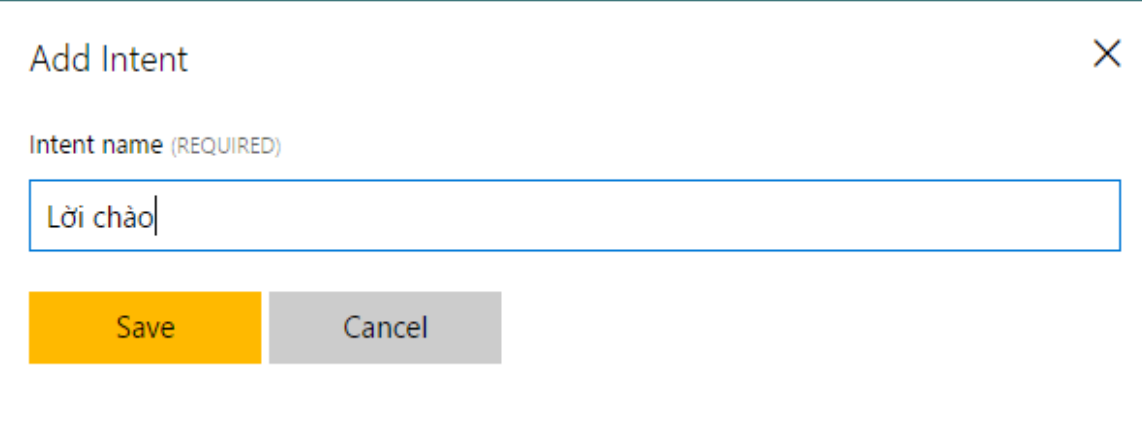
Facts & statistics about the app's data and the received endpoint hits at any period of time ... [Learn more](#)



HÌNH 9

## b) Dạy ngôn ngữ

- Việc xử lý ngôn ngữ cũng dựa trên những mục như vậy , chẳng hạn khi chúng ta nói câu: **“Tôi muốn biết thời tiết của thành phố Hồ Chí Minh”**.
- Máy tính không thể hiểu được toàn bộ câu nói , mà nó sẽ dựa trên các thành phần trong câu để đưa ra việc xử lý. Chẳng hạn , trong câu vừa rồi , máy sẽ xác định:
  - **Intents (mục tiêu)** trong câu của chúng ta là : **Thời tiết**.
  - **Entities (thực thể)** trong câu của chúng ta là : **Thành phố Hồ Chí Minh**.
- Mỗi một mục tiêu hay thực thể sẽ có một điểm số : **Score** để đánh giá mức độ ưu tiên, chẳng hạn ở đây **Thời tiết** có **Score** là **0.9** và **Thành phố Hồ Chí Minh** có **Score** là **0.6** . Máy sẽ ưu tiên điểm số cao nhất, vì mục tiêu người hỏi đang muốn biết về Thời tiết , và sẽ ưu tiên theo thứ tự giảm dần rồi dẫn tới kết quả mà ta đã định trước.
- Để minh họa cho việc xử lý , ở đây chúng ta thử tạo một Intent có tên là **Lời chào**:



HÌNH 10

- Sau đó thêm vào một số câu nói đặc trưng của lời chào như : Hi , hello , Xin chào , chào...

## Lời chào

Here you are in full control of this intent; you can manage its utterances, used entities and suggested utterances ... [Learn more](#)

Utterances (4)   Entities in use   Suggested utterances

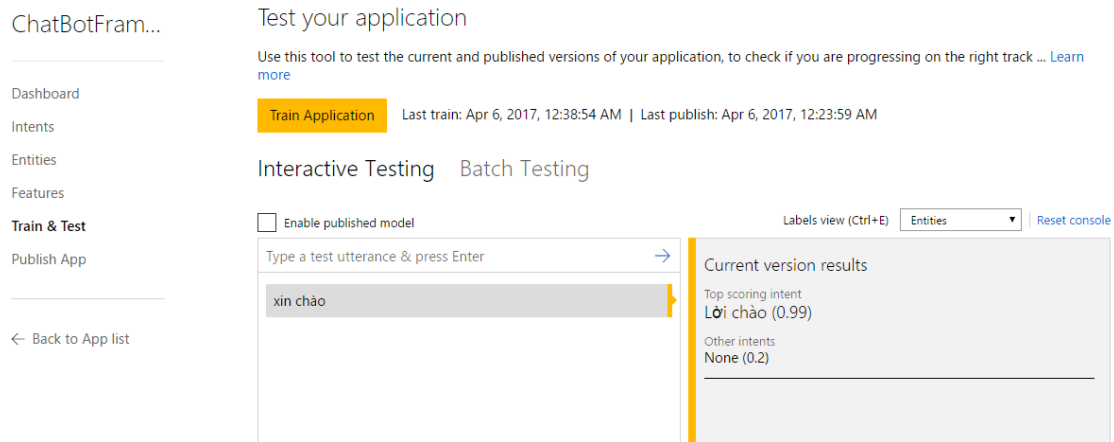
hello		×
Save   Discard   Delete   Reassign Intent ▾		Labels view (Ctrl+E):   Entities ▾
Search in utterances ... 🔍		🔍
<input type="checkbox"/>	Utterance text	Predicted Intent
<input type="checkbox"/>	chào	<b>0.99</b> Lời chào
<input type="checkbox"/>	xin chào	<b>0.99</b> Lời chào
<input type="checkbox"/>	hi	<b>0.99</b> Lời chào
<input type="checkbox"/>	hello	<b>0.99</b> Lời chào

HÌNH 11

- Điểm số ( Score ) của tất cả các câu nói ta vừa thêm vào là 0.99 , và khi người dùng nhập vào một câu nào đó có những chữ như trên , máy sẽ tính điểm và thực hiện việc chúng ta định sẵn cho nhóm đó , nếu như điểm số là cao nhất.
- Sau đó chúng ta lập trình cho máy , chỉ cần thấy những cụm từ như **Xin chào** thuộc nhóm **Lời chào** , máy sẽ trả lời lại 1 câu chào tương ứng. Tuy nhiên , khi chúng ta nói 1 câu như: **“Xin chào, cho tôi biết thời tiết của Hà Nội”**.
- Mặc dù có từ **Xin chào** trong câu, nhưng máy sẽ không chào lại ta nữa, vì ở đây có Mục tiêu là **thời tiết** và Thực thể là **Hà Nội**, vậy thì máy sẽ ưu tiên hơn vì điểm số của chúng cao hơn điểm số của **Xin chào**. Chẳng hạn máy sẽ trả lời: **“Thời tiết ở Hà Nội hiện đang mưa”**.

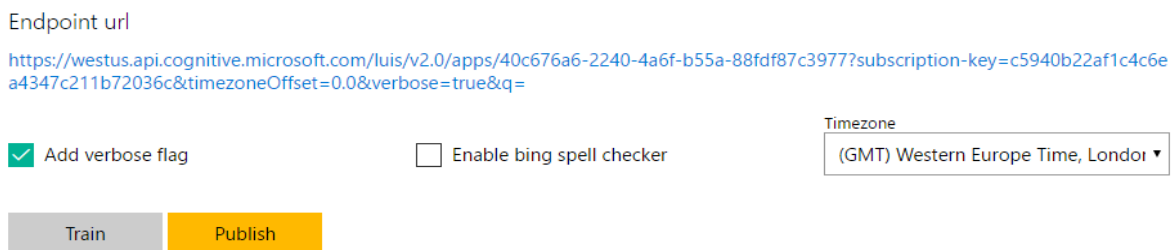
### c) Train bot

- Sau khi tạo ra một vài từ đặc trưng để máy xử lý, ta sẽ thực hiện việc Train cho máy, sau đó Publish để lấy đường dẫn và thêm nó vào trong ứng dụng của chúng ta.



HÌNH 12

- Ở mục Train & Test , sau khi bấm Train Application , có thể thấy khi ta nhập vào “ xin chào ” , máy sẽ hiểu nó là 1 lời chào với điểm số 0.99.
- Sau đó ở mục Publish App , ta thực hiện Publish và copy đường link hiện ra:



HÌNH 13



- Tiếp theo, vào lại file app.js, thêm bộ api vào ứng dụng của chúng ta bằng cách gõ các dòng code:

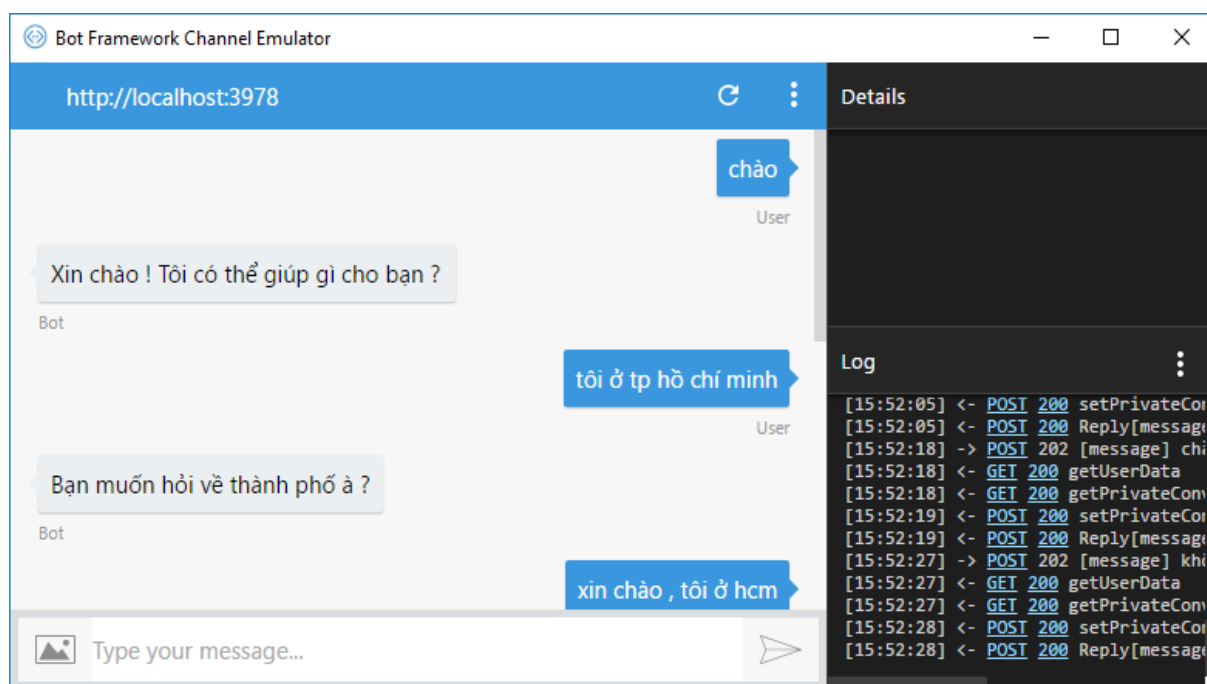
```
var model = 'https://westus.api.cognitive.microsoft.com/luis/v2.0/apps/40c676a6-2240-4a6f-b55a-  
var recognizer = new builder.LuisRecognizer(model);  
var intent = new builder.IntentDialog({ recognizers: [recognizer] });
```

- Gán model cho đường link vừa copy , sau đó chỉnh sửa để Bot hiện ra các Dialog trả lời theo ý mình.

```
//=====  
// Bots Dialogs  
//=====  
  
bot.dialog('/', intent);  
  
intent.matches('None' , function (session){  
    session.send('Tạm thời tôi chưa hiểu !');  
});  
intent.matches('Lời chào' , function (session){  
    session.send('Xin chào ! Tôi có thể giúp gì cho bạn ?');  
});  
intent.matches('Thành phố' , function (session){  
    session.send('Bạn muốn hỏi về thành phố à ?');  
});
```

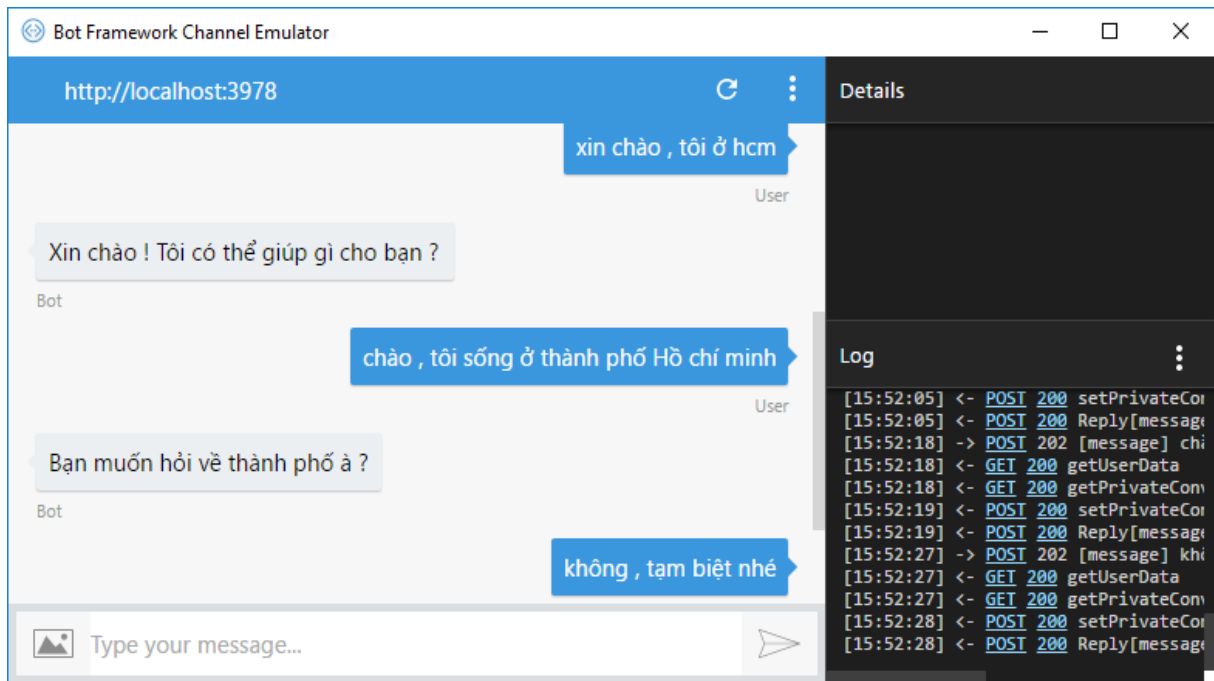
- Ở đây chúng ta sử dụng IntentDialog, nó sẽ phát hiện ra nhóm lời thoại mà ta đã thiết lập sẵn, từ đó đưa ra các xử lý phù hợp. Sau đó mở Command line gõ: **node app.js**

- Mở Bot emulator lên và test thử:



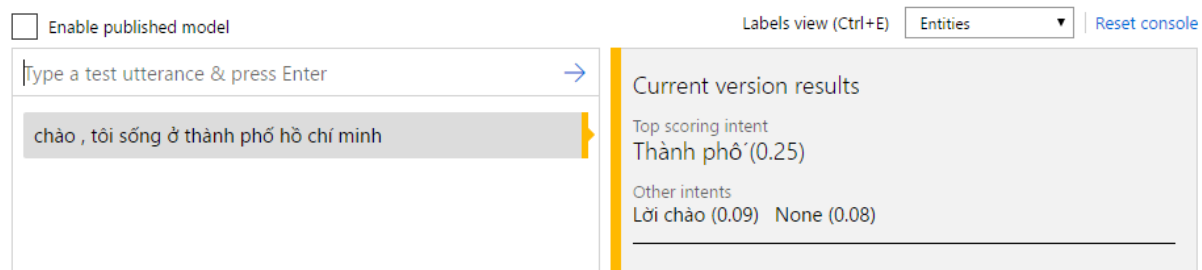
HÌNH 14

- Có thể thấy, khi chúng ta viết rõ **Xin chào** hoặc **Thành phố Hồ chí minh**, nó sẽ trả lời tương ứng, nhưng khi viết không rõ thì nó sẽ ưu tiên cái có điểm số cao hơn.



- Khi viết tắt “ hcm ” nó lại trả lời xin chào , vì lúc này Score của xin chào lớn hơn

## Interactive Testing Batch Testing



- Ở mục Train ta có thể thấy score của từng trường hợp , ở đây chỉ có 1 chữ chào , trong khi Thành phố Hồ chí minh đầy đủ thông tin , nên Score sẽ cao hơn.

#### **d) Test chatbot**

- Tiếp theo sẽ là công đoạn Test ChatBot trên host và kết nối với các ứng dụng như Facebook Messenger, Skype...
- Trước hết, truy cập trang : <https://dev.botframework.com/> và đăng nhập tài khoản Microsoft. Sau đó tạo app cho mình ở mục **Register a bot**.

- Nhập thông tin chatbot và tạo app id, app password để sử dụng , ở mục Messaging endpoint nhập vào đường dẫn đến server Heroku mà ta đã tạo:

## Configuration

Messaging endpoint:

https URL

Register your bot with Microsoft to generate a new App ID and password

Create Microsoft App ID and password

Paste your app ID below to continue

Microsoft App ID from the Microsoft App registration portal

HÌNH 15

- Nhấn vào Create Microsoft App ID an password , nó sẽ mở một trang khác , nhập tên app của ta vào và Save

App name

ChatBotTDT

App ID

Save

There are unsaved changes.

HÌNH 16

- Save xong nó sẽ hiện App Id, click vào mục generate an app pass.... Để tạo pass và save lại pass ở 1 nơi nào đó.

App ID

29f8f7b7-3439-426a-8ee0-cb8c8893e961

Generate an app password to continue

HÌNH 17

- Sau khi tạo pass , copy lại ở 1 file txt nào đó

## New password generated

Copy this password now, this is the only time when it will be displayed. Please store it securely and paste it into your bot configuration file.

xGByDgi48drpyPcmOyM8Kg3

Ok

HÌNH 18

The screenshot shows the Microsoft Bot Framework Developer Portal interface for a bot named "ChatBotTDT". The interface is divided into three main sections: Details, Test connection, and Chat.




- Details:** This section contains the following information:
  - Bot handle: df6f2d81-0842-453e-9348-f3f8c2117970
  - Bot Framework Version: 3.0
  - Messaging endpoint: https://doan-2.herokuapp.com/
  - Microsoft App ID: df6f2d81-0842-453e-9348-f3f8c2117970
- Test connection to your bot:** This section contains a "Test" button.
- Chat:** This section contains a chat window with a "type your message..." input field and a send button.

The top of the interface features the bot's name "ChatBotTDT", the user "DaiThinh", and a "#Hello" channel. A "Publish" button is located in the top right corner.

HÌNH 19

- Đây là giao diện chính của trang <https://dev.botframework.com/> khi ta đã đăng nhập và chọn My bot. Có thể thấy nó có cả mục cho chúng ta hội thoại với Bot.



Channels						
		Test link	Issues	Enabled	Published	
	Facebook Messenger	<a href="#">Message Us</a>	0	Yes	<input type="checkbox"/> Off	<a href="#">Edit</a>
	Skype	<a href="#">Add to Skype</a>	0	Yes	<input type="checkbox"/> Off	<a href="#">Edit</a>
	Web Chat		14	Yes	<input type="checkbox"/> Off	<a href="#">Edit</a>

[Get bot embed codes](#)

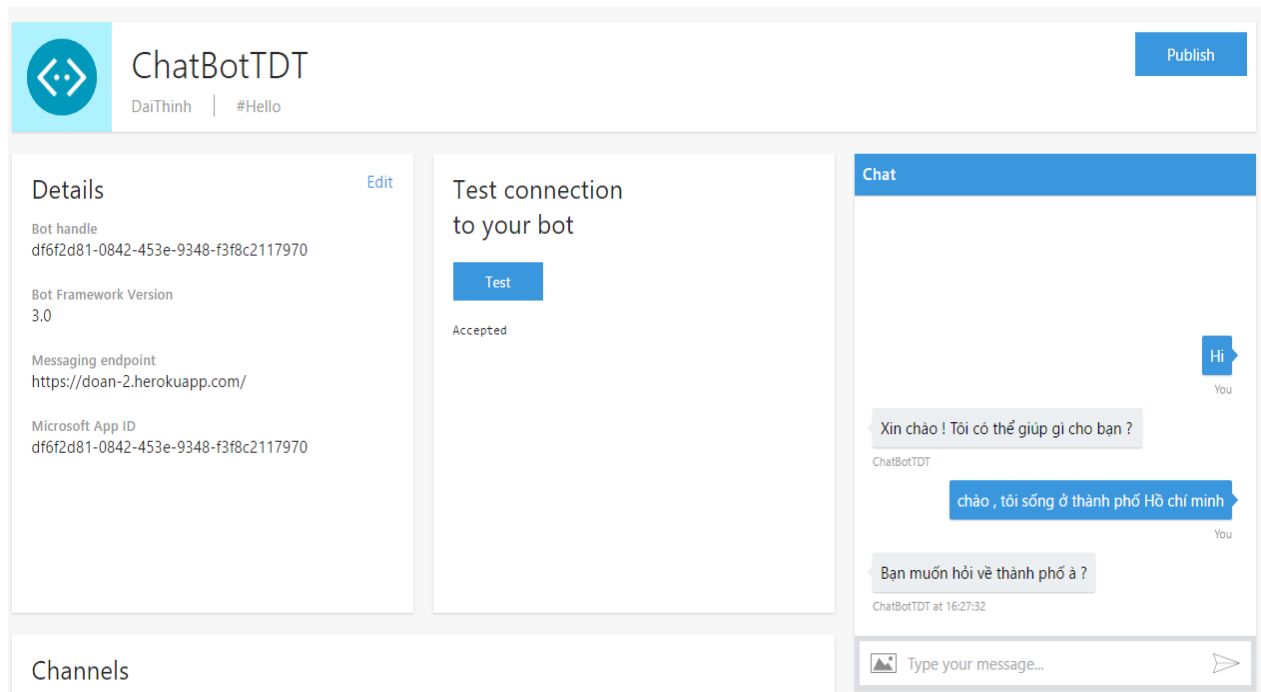
HÌNH 20

- Còn ở dưới sẽ là các kênh để ta kết nối với các ứng dụng bên ngoài , đó chính là **Bot Framework Developer Portal** mà Microsoft đã hỗ trợ rất tốt cho lập trình viên.
- Trước hết là Test thử trên Host heroku mà ta đã tạo ở trên , trước hết thêm vào app id và app password mà ta đã tạo vào file app.js , sau đó push các thay đổi lên host heroku.
- Thay đổi 1 chút ở code, khi test trên Bot emulator cũng cần phải nhập Id và Key vào.

```
const APPID = "df6f2d81-0842-453e-9348-f3f8c2117970";
const APPKEY = "FZ620n2AQpZ1K9a4H33fzZX";

// Create chat bot
var connector = new builder.ChatConnector({
  appId: APPID,
  appPassword: APPKEY
});
var bot = new builder.UniversalBot(connector);
server.post('/', connector.listen());
```

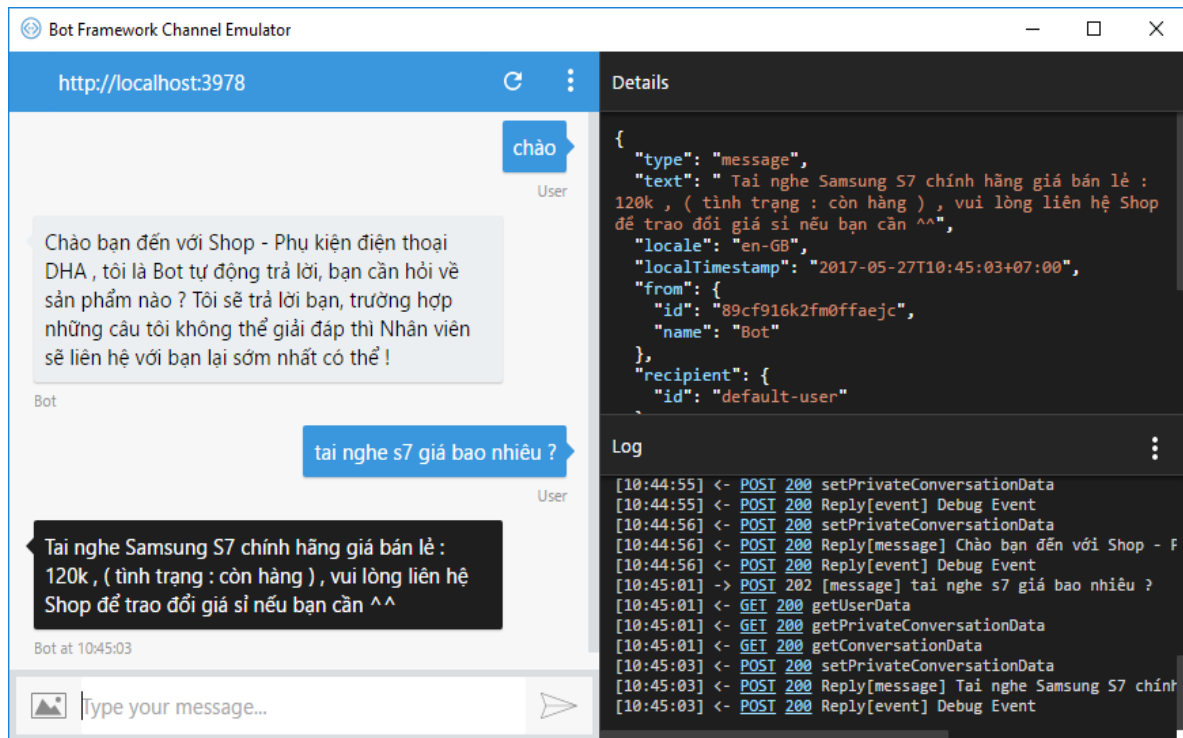
- Push lên server và xem kết quả. Nhấn Test , nếu hiện Accepted tức là đã thành công.



HÌNH 21

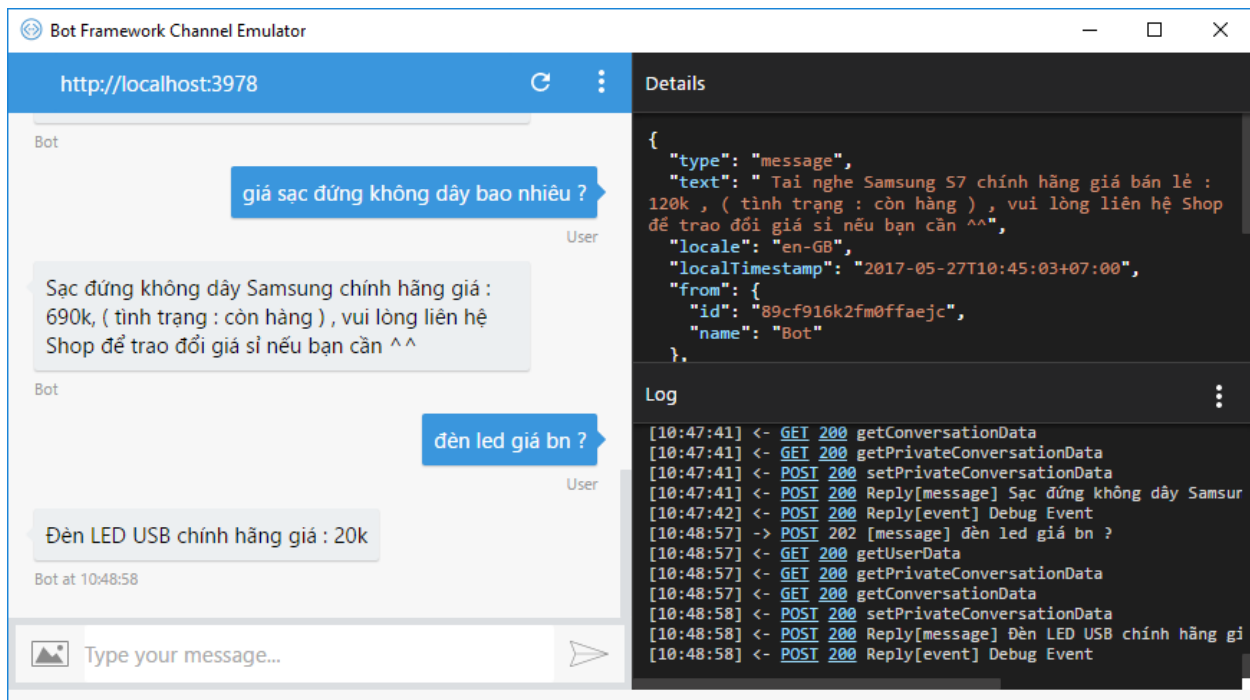
## CHƯƠNG 3: GIỚI THIỆU CHATBOT DHA

- ChatBot DHA được thiết kế cho Shop Phụ kiện điện thoại DHA , bot có thể trả lời các câu cơ bản như : hỏi giá sản phẩm , cho xem các loại sản phẩm , đặt hàng...



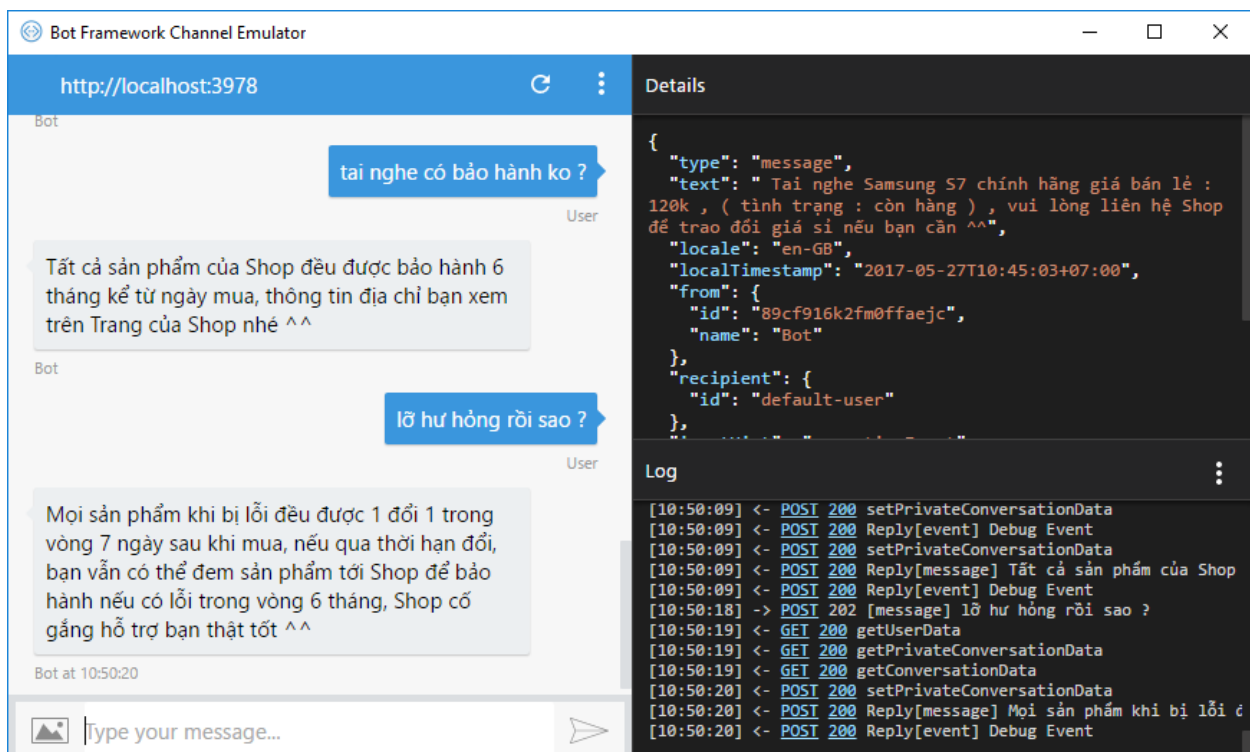
HÌNH 22

Một số câu Bot được huấn luyện để trả lời:



HÌNH 23

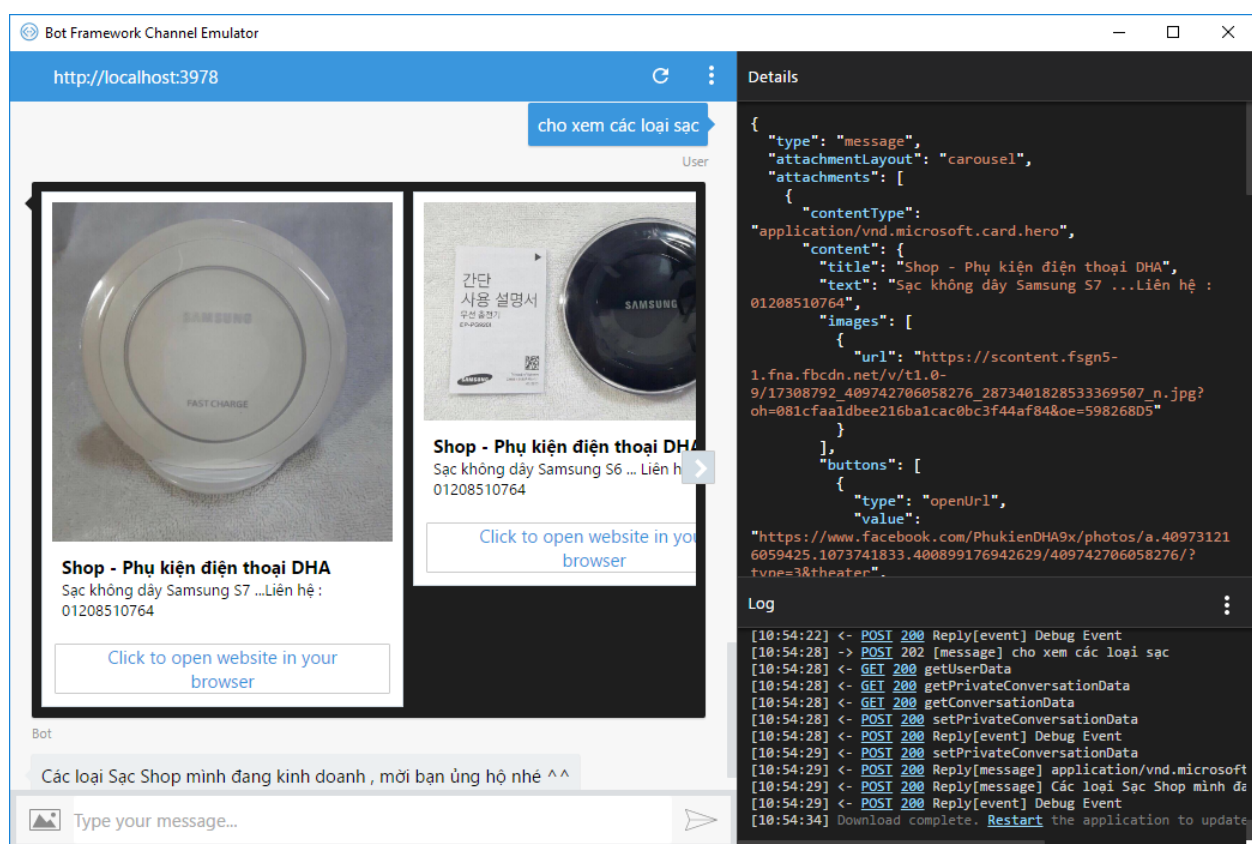
Một số câu Bot được huấn luyện để trả lời:



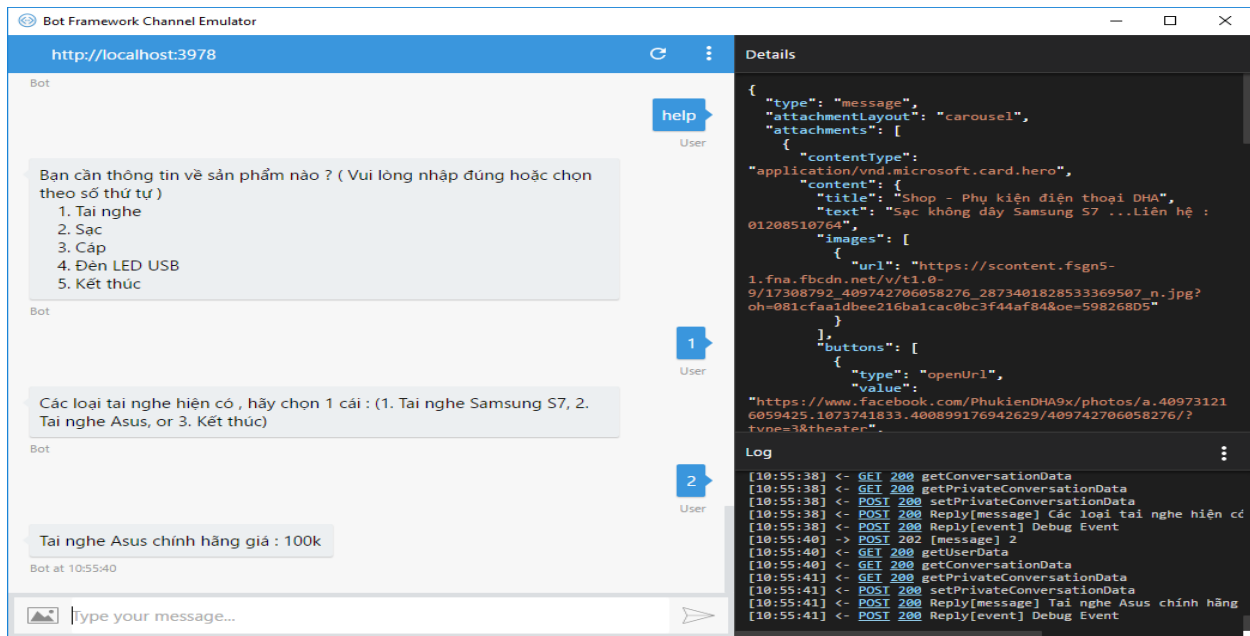
HÌNH 24



Một số câu Bot được huấn luyện để trả lời:



HÌNH 26



HÌNH 27

Đăng kí thông tin người dùng trước khi đặt hàng:

The screenshot displays the Bot Framework Channel Emulator interface. The left pane shows a chat conversation between a bot and a user. The right pane displays the JSON message object for the last user message.

**Chat Conversation:**

- Bot:** http://localhost:3978
- User:** đặt 2 cái tai nghe s7
- Bot:** Vui lòng cho biết thông tin của bạn trước khi đặt hàng bằng cách nhập : "Đăng kí"
- User:** đăng kí
- Bot:** Bạn vui lòng cho Shop biết tên của bạn nhé ^^
- User:** thịnh
- Bot:** Xin chào thịnh , cho mình xin số để liên hệ nhé ^^
- User:** 01111111111
- Bot:** Xong rồi , bây giờ bạn có thể đặt hàng dựa trên thông tin bạn đã nhập , sau đó Shop sẽ liên hệ lại nhé.

**JSON Message Object:**

```
{
  "type": "message",
  "attachmentLayout": "carousel",
  "attachments": [
    {
      "contentType": "application/vnd.microsoft.card.hero",
      "content": {
        "title": "Shop - Phụ kiện điện thoại DHA",
        "text": "Sạc không dây Samsung S7 ...Liên hệ : 01208510764",
        "images": [
          {
            "url": "https://scontent.fsgn5-1.fna.fbcdn.net/v/t1.0-9/17308792_409742706058276_2873401828533369507_n.jpg?oh=081cfaa1dbec216ba1cac0bc3f44af84&oe=598268D5"
          }
        ],
        "buttons": [
          {
            "type": "openUrl",
            "value": "https://www.facebook.com/PhukienDHA9x/photos/a.409731216059425.1073741833.400899176942629/409742706058276/?tvtne=3&theater"
          }
        ]
      }
    }
  ]
}
```

**Log:**

```
[10:57:03] <- GET 200 getPrivateConversationData
[10:57:03] <- GET 200 getConversationData
[10:57:03] <- POST 200 setPrivateConversationData
[10:57:03] <- POST 200 Reply[message] Xin chào thịnh , cho mình
[10:57:04] <- POST 200 Reply[event] Debug Event
[10:57:08] -> POST 202 [message] 01111111111
[10:57:08] <- GET 200 getUserData
[10:57:08] <- GET 200 getPrivateConversationData
[10:57:08] <- GET 200 getConversationData
[10:57:09] <- POST 200 setPrivateConversationData
[10:57:09] <- POST 200 Reply[message] Xong rồi , bây giờ bạn có
[10:57:09] <- POST 200 Reply[event] Debug Event
```

HÌNH 28





Bot Framework Channel Emulator

http://localhost:3978

⌂

⋮

Bot

đặt tiếp 3 cái sạc đứng không dây

User

thịnh

Điện thoại : 01111111111

Hình thức : Thanh toán khi nhận hàng

Tên sản phẩm Giá

3

690

Total 2070

Bot

Vui lòng xem lại bảng đặt hàng và xác nhận bằng cách gõ bất kì , nếu có sai sót hoặc muốn hủy , vui lòng nhập "Hủy"

Bot

hủy

User

Đã hủy đặt hàng , bạn có thể đặt lại bất cứ lúc nào ^^

Type your message...

➤

Details

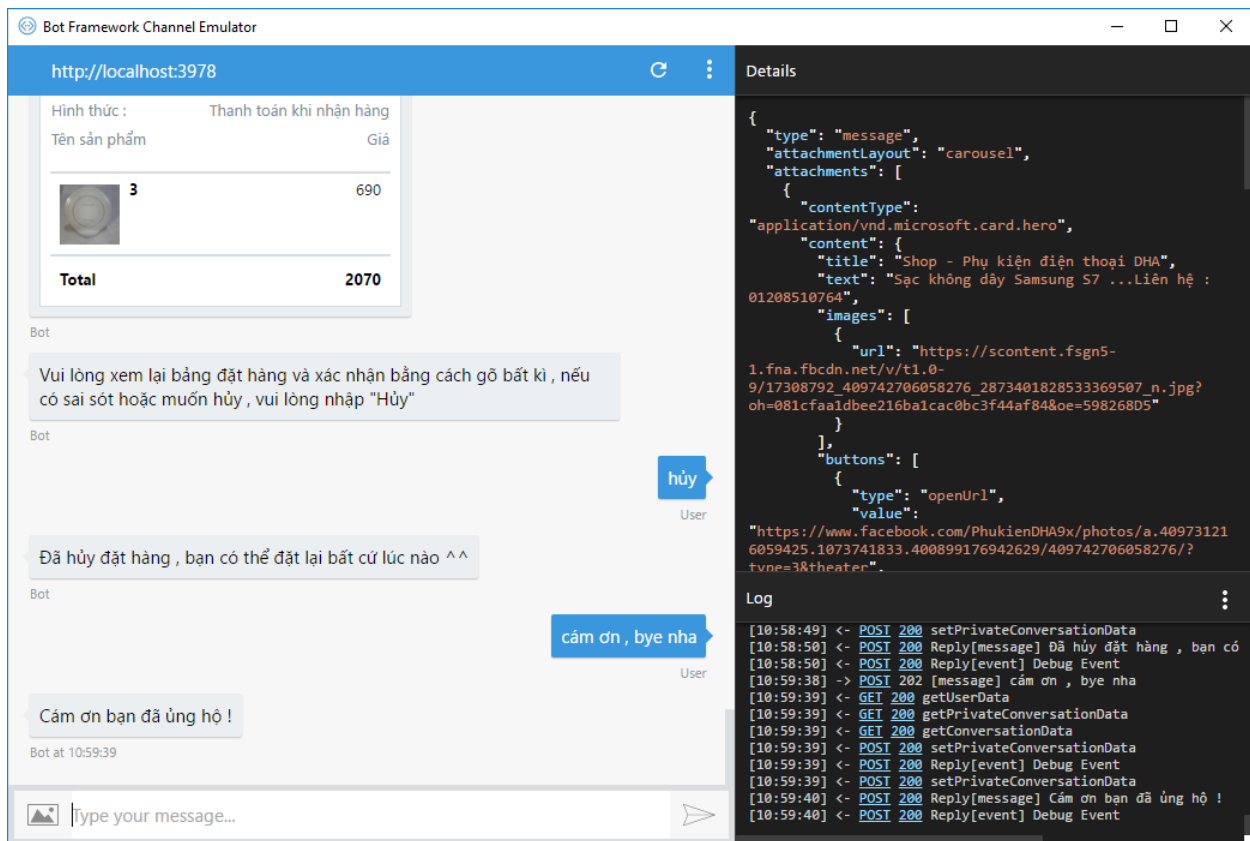
```
{
  "type": "message",
  "attachmentLayout": "carousel",
  "attachments": [
    {
      "contentType":
"application/vnd.microsoft.card.hero",
      "content": {
        "title": "Shop - Phụ kiện điện thoại DHA",
        "text": "Sạc không dây Samsung S7 ...Liên hệ :
01208510764",
        "images": [
          {
            "url": "https://scontent.fsgn5-
1.fna.fbcdn.net/v/t1.0-
9/17308792_409742706058276_2873401828533369507_n.jpg?
oh=081cfaa1dbee216ba1cac0bc3f44af84&oe=598268D5"
          }
        ],
        "buttons": [
          {
            "type": "openUrl",
            "value":
"https://www.facebook.com/PhukienDHA9x/photos/a.40973121
6059425.1073741833.400899176942629/409742706058276/?
tvp=3&theater"
          }
        ]
      }
    }
  ]
}
```

Log

```
[10:58:42] <- POST 200 Reply[event] Debug Event
[10:58:42] <- POST 200 setPrivateConversationData
[10:58:42] <- POST 200 Reply[message] application/vnd.microsoft
[10:58:43] <- POST 200 Reply[message] Vui lòng xem lại bảng đặt
[10:58:43] <- POST 200 Reply[event] Debug Event
[10:58:49] -> POST 202 [message] hủy
[10:58:49] <- GET 200 getUserData
[10:58:49] <- GET 200 getPrivateConversationData
[10:58:49] <- GET 200 getConversationData
[10:58:49] <- POST 200 setPrivateConversationData
[10:58:50] <- POST 200 Reply[message] Đã hủy đặt hàng , bạn có
[10:58:50] <- POST 200 Reply[event] Debug Event
```

HÌNH 30

43



HÌNH 31

Phía trên là một số câu đơn giản Bot đã được huấn luyện để trả lời. Dưới đây là các thông tin mà Bot đã được train.

## DHA\_Shop

Version: 0.1

Settings

Dashboard

**Intents**

Entities

Prebuilt domains PREVIEW

Features

Train & Test

Publish App

← Back to App list

## Intents

A listing of intents in the application. Click an intent to view/edit its details, or add a new intent ... [Learn more](#)

Add Intent

Add prebuilt domain intents

Search for intent ...

Intent Name	Utterances	
Bye	4	<a href="#">Edit</a> <a href="#">Delete</a>
Greetings	4	<a href="#">Edit</a> <a href="#">Delete</a>
Help	1	<a href="#">Edit</a> <a href="#">Delete</a>
None	0	
Order	43	<a href="#">Edit</a> <a href="#">Delete</a>
ProductCost	74	<a href="#">Edit</a> <a href="#">Delete</a>
ProductInfo	84	<a href="#">Edit</a> <a href="#">Delete</a>
SignUp	5	<a href="#">Edit</a> <a href="#">Delete</a>

HÌNH 32





Mục Intents : các mục tiêu Bot được hướng tới để trả lời

<div> <div>Save</div> <div>Discard</div> <div>Delete</div> <div>Reassign Intent</div> </div> <div> <div>Labels view (Ctrl+E):</div> <div>Entities</div> <div>Search in utterances ...</div> </div>		
<input type="checkbox"/>	Utterance text	Predicted Intent
<input type="checkbox"/>	cái [\$Tai nghe::Tai nghe Samsung S7 ] bn vậy ?	Loading
<input type="checkbox"/>	[\$Tai nghe ] samsung nhieu /	Loading
<input type="checkbox"/>	[\$Sạc::Sạc đứng không dây SS ] bao nhiêu ?	Loading
<input type="checkbox"/>	cho hỏi giá [\$Cáp::Cáp 1m chuẩn CE ]	Loading
<input type="checkbox"/>	[\$Cáp::Cáp 1m chuẩn CE ] bao nhiêu ?	Loading
<input type="checkbox"/>	giá [\$Tai nghe::Tai nghe Asus ] là bao nhiêu ?	Loading
<input type="checkbox"/>	[\$Tai nghe::Tai nghe Asus ] bao nhiêu ?	Loading
<input type="checkbox"/>	cho mình hỏi [\$Tai nghe::Tai nghe Samsung S7 ] giá bao nhiêu ?	Loading
<input type="checkbox"/>	sợi dây [\$Cáp ] giá bao nhiêu ?	Loading
<input type="checkbox"/>	[\$Tai nghe ] ss nhieu ?	Loading

HÌNH 33

Một số câu mà Bot được Train để trả lời, không cần phải train tất cả mọi câu có thể, chỉ cần train từ 1 – 2 lần , Bot sẽ có thể nhận diện được những câu tiếp theo và tự đưa vào loại Intent phù hợp.









Utterances (74)   Entities in use (11)   Suggested utterances

<div>  Save            Discard            Delete           Reassign Intent ▾         </div> <div>           Labels view (Ctrl+E):           <div>Entities ▾</div> <div>Search in utterances ... </div> </div>		
<input type="checkbox"/>	Utterance text	Suggested Intent
<input type="checkbox"/>	cho xem [ \$Tai nghe::Tai nghe Samsung S7 ]	<b>1</b> ProductInfo
<input type="checkbox"/>	thế còn sạc ko dây giá bn ?	<b>1</b> ProductCost
<input type="checkbox"/>	cho xem [ \$Tai nghe ]	<b>1</b> ProductInfo
<input type="checkbox"/>	chào	<b>0.95</b> Greetings
<input type="checkbox"/>	cho tôi đặt [ \$number ] cục [ \$Sạc::Sạc đứng không dây SS ]	<b>0.84</b> Order
<input type="checkbox"/>	cho xem [ \$Tai nghe::Tai nghe Samsung S7 ] đi	<b>1</b> ProductInfo
<input type="checkbox"/>	nhận xét về anh Đại thịnh xem nào	<b>0.68</b> ProductInfo
<input type="checkbox"/>	đặt [ \$number ] cái [ \$Sạc::Sạc nằm không dây SS ]	<b>1</b> Order

HÌNH 34

Những câu người dùng nhập vào mà Bot không hiểu sẽ nằm trong phần Suggested utterances, ở mỗi câu Bot sẽ đưa ra gợi ý nó thuộc loại Intent nào, nếu như đúng ta chỉ cần save lại và train cho Bot, nếu như không, ta chỉ việc đưa nó vào trong Intent mình muốn, và Bot sẽ được train điều đó.

Entities list   Labeled utterances   Suggested utterances

Add custom entity	Add prebuilt entity	Add prebuilt domain entity
Entity Name ↓	Entity Type	
Bảo hành	Simple	
Cáp has: [Cáp 1,5m chuẩn CE, Cáp 1m chuẩn CE]	Hierarchical	
Lỗi	Simple	
Sạc has: [Sạc đứng không dây SS, Sạc nhanh chuẩn CE, Sạc nằm không dây SS, Sạc 5V 2A chuẩn CE]	Hierarchical	
Tai nghe has: [Tai nghe Asus, Tai nghe Samsung S7]	Hierarchical	
number	Prebuilt	
phonenumber	Prebuilt	
Đèn LED USB	Simple	

HÌNH 35

Trong mục Entites, ta sẽ train cho Bot các thực thể có thể nhận diện trong câu. Chẳng hạn, khi người dùng hỏi về giá sản phẩm, Bot sẽ đưa về Intent là **ProductCost**, và thực thể ( Entity ) trong câu sẽ là các sản phẩm như : Tai nghe s7, sạc....


















Điều này giúp cho Bot xác định chính xác cái mà người nói đang muốn hướng đến, chính là : Giá ( Intent : ProductCost ) của tai nghe s7 ( Entity : Tai nghe s7 ) là bao nhiêu.

## Features

Use these advanced features to improve performance and avoid mistakes in identifying and interpreting utterances ... [Learn more](#)

Phrase list features (8/10)   Pattern features (0/10)

Add phrase list

Active	Phrase list 	Mode	
	<b>Bảo hành:</b> [bảo hành,bh]	Exchangeable	
	<b>Cost:</b> [giá,bao nhiêu,nhiều tiền,nhiu,bn]	Exchangeable	
	<b>Cáp 1,5m:</b> [cáp 15m,cáp 1,5m,5m chuẩn CE]	Exchangeable	
	<b>Cáp 1m:</b> [cáp 1m,dây cáp 1m,cáp 1m chuẩn CE]	Exchangeable	
	<b>Info:</b> [cho biết,cho hỏi,thông tin,thế nào,thì sao,tư vấn,cho coi,cho xem,gửi hình,xem...]	Exchangeable	
	<b>Lỗi:</b> [lỗi,hư,hỏng,không nghe,không vào điện,hư một bên]	Exchangeable	
	<b>Sạc đúng không dây chuẩn CE:</b> [Sạc đúng không dây chuẩn CE,sạc đúng,sạc s7]	Exchangeable	
	<b>Sản phẩm:</b> [sạc nằm,sạc nằm không dây,sạc không dây nằm,sạc không dây loại nằm...]	Exchangeable	

HÌNH 36

Mục Features sẽ bao gồm các nhóm từ tương đồng, chẳng hạn khi hỏi về giá sản phẩm, người dùng có thể dùng các từ đồng nghĩa như : giá, bao nhiêu, nhiều tiền... hoặc thậm chí là các chữ viết tắt như : “ bn ”.....

Điều này giúp Bot thông minh hơn , có thể nhận ra các từ tương đồng nhằm giúp phân biệt ý định của người nói một cách chính xác.



Use this tool to test the current and published versions of your application, to check if you are progressing on the right track ... [Learn more](#)

Train Application

Last train: May 21, 2017, 5:41:14 PM | Last publish: May 21, 2017, 5:41:17 PM

## Interactive Testing Batch Testing

☒ Enable published model

Labels view (Ctrl+E) Entities [Reset console](#)

Type a test utterance & press Enter →

`[ $Đèn LED USB ]` bao nhiêu vậy ?

```
},
{
  "intent": "Help",
  "score": 2.31358523E-08
},
{
  "intent": "Bye",
  "score": 7.064445E-10
}
],
"entities": [
  {
    "entity": "đèn led",
    "type": "Đèn LED USB",
    "startIndex": 0,
    "endIndex": 6,
    "score": 0.9991802
  }
]
}
```

Mục Train & Test để người dùng kiểm tra lại những gì mình đã train cho Bot, khi bấm Train, Bot sẽ chính thức học những câu được huấn luyện.

## Tài liệu tham khảo:

- Video giới thiệu ChatBot, link youtube:  
<https://www.youtube.com/watch?v=sbidsuV3esU&t=211s>
- Tài liệu tham khảo: Bài viết giới thiệu Bot Framework của Microsoft Azure  
<http://azurevn.net/2016/11/18/tong-quan-ve-microsoft-bot-framework/>
- Tổng hợp từ nhiều nguồn khác nhau về ChatBot Framework.

## Bảng đánh giá mức độ hoàn thành của sinh viên:

Họ tên	Phân chia công việc	Đã làm được	Chưa làm được
Nguyễn Đại Thịnh	Tìm hiểu về các khái niệm ChatBot, các Framework về ChatBot	Hiểu được cơ bản thế nào là Chatbot, hiểu cơ bản về cách làm việc với Framework	Còn nhiều thứ chưa nắm kĩ. Thiếu sót về lượng kiến thức cần đạt được.
	Xây dựng ứng dụng ChatBot minh họa	Sử dụng bộ Framework của Microsoft và các API mà framework này cung cấp để tạo Chatbot.	Chatbot chưa thể trả lời được nhiều. Mới chỉ dừng lại ở những mục tiêu đơn giản do người lập trình đặt ra.
Nguyễn Trí Quang	Tìm hiểu khái niệm Framework, khái niệm Chatbot.	Hiểu được cơ bản về Chatbot.	Vẫn còn thiếu sót về lượng kiến thức cần đạt được.
	Phân tích các dịch vụ có sẵn (các API) mà Microsoft cung cấp, chỉ ra những cái có thể áp dụng để xây dựng ứng dụng cho đề tài này.	Sử dụng được dịch vụ Language Understanding Intelligent Service (LUIS)	Chưa thể áp dụng các dịch vụ còn lại. Chưa sử dụng triệt để dịch vụ LUIS.

## Phụ lục

- Chương trình chạy trên công cụ Botframework Emulator như phần trên bài viết đã đề cập. Để chạy được source code, vào thư mục chứa source nhấp chuột phải, chọn Open command window here sau đó gõ dòng lệnh:
  - **node app.js**
- Sau đó mở Bot Emulator lên, nhập vào ID và Pass rồi test thử Bot.
  - **APPID** = "f7fa8691-b494-4f6f-b395-e2f8d41bc9e8";
  - **APPKEY** = "CZNmqLtj4oTvPwRb8BZQQbt";