Newbot Framework for chatbot development

CAPABILITIES AND COMPARISON WITH OTHER FRAMEWORKS

# **Prerequisites**

* Node.js & NPM
* Ngrok

## **Facebook requirements**

* Facebook Page
* FB Developer Account
* FB App: contains settings for the Messenger bot
* Webhook: Actions that take place in conversation, are sent as events to the webhook. Webhook Setup: (<https://developers.facebook.com/docs/messenger-platform/getting-started/webhook-setup>)

# **About Newbot Framework**

* A framework for designing conversational scenarios. Thus, one can conceive chatbots or any system based on a human-machine conversation.
* It is based on a scalable, skill-oriented design. This allows you to organize the project, share skills with other developers and perform unit tests.
* Scenario execution can be performed on any platform if we create the adapter.
* Use of **ConverseScript**, a language adapted to the conversational scenario.
* NewBot is not a natural language system (NLP) but a framework for writing conversational scenarios. However, you can also use existing platforms like DialogFlow, Wit.ai, Microsoft LUIS, RASA NLU, etc., and NewBot Cloud!

With third-party systems, your chatbot can trigger dialogues in natural language.

## **About Newbot Framework’s Native NLP**

**Advantages:**

* Independent of third party platforms (no need for DialogFlow or others)
* You can integrate this system directly into the browser. Useful for creating an offline chatbot

**Disadvantages:**

* No visibility on the phrases used by the user. On third-party platforms, they record the new sentences in order to add them to the chatbot's understanding.
* Extraction of entities (a place, the name of a person, etc.) is less elaborate
* On the browser side, your chatbot will be heavier

## **Capabilities of Newbot**

### Newbot emulator

* 1. Terminal: No need to have accounts on different sources, it's just a simulation  
     To test the formats from another source, for example Messenger, type:   
     ***newbot emulator -s messenger***
  2. Graphic emulator:   
     ***newbot serve*** performs a listen on port 3000.   
     To change the port: ***newbot serve -p <PORT>***
  3. Test directly on Messenger

### Unit Tests

Read all files ending with *.spec.js*

***newbot test***

### Build the Sources

Build sources for NodeJS and the browser

***newbot build***

### Train the chatbot

Using the NLP system already integrated into Newbot

***newbot train***

### Assign webhooks in production for different platforms

### Compatibility of Newbot with other platforms

Google Assistant, Slack, Amazon Alexa, Telegram, Viber, Skype, Twitter, etc

## **Deploy chatbot on Newbot Cloud**

All-in-one workflow that combines deployment, integration with instant messaging platforms, conversational website creation, and natural language understanding.

* Serverless
* Free registration. No trial period.
* Integrate it on 15 different platforms
* Reclassify the messages into intentions.
* Put in production and use AI.

# **Comparison with other frameworks**

MS BotFramework, Botkit, Chatterbot

## **Why we like Newbot (**[**https://medium.com/@NewBot/5-good-reasons-to-use-newbot-framework-5fee63839a8e**](https://medium.com/@NewBot/5-good-reasons-to-use-newbot-framework-5fee63839a8e)**)**

### Conversational sites and Progressive Web App (PWA)

While most frameworks direct development to specific platforms, Newbot gives full attention to the conversation, no matter the platform. The chatbot can run directly in the browser.

Newbot can create an offline conversational site, like a PWA, **unlike** *MS BotFramework, Botkit, Chatterbot.*

### Create scenarios easily with ConverseScript

The above frameworks, use JavaScript or Python language to create a chatbot. Powerful but insufficient languages.

Some others use SuperScript, RiveScript or ChatScript. Still not enough because they luck programming capabilities (condition, loop, function, etc.) and additional training on how to use them.

Newbot uses a special language for scenario creation, **ConverseScript.**

* + Easy to learn syntax, very close to JavaScript and Typescript.
  + Create variables, conditions, loops, functions, use JavaScript functions, and more.
  + Keep track of the user’s progress in the script. Save it to a database or local storage in the browser.
  + Integrate multiple conversation triggering systems: NLP and various events.
  + VS Code extension for syntax color

### Modular

Build the structure of the skills like in Web frameworks (Angular, VueJS, etc.).

Build each skill as a different module, making it reusable and sharable. Each skill has its own conversational script, its own JavaScript configuration.

### Unit Tests

Most frameworks don’t offer unit tests. You usually have to go through third-party repositories that sometimes are no longer maintained.

With Newbot, we can easily run a unit test per skill, testing it. (*newbot test*)

### Internationalization

A lot of chatbot frameworks need the use of external libraries for internationalization.

Newbot, uses an external file to group all the responses of the chatbot into the desired language. So, depending on the language of the user, the chatbot will answer in the correct language.

## Choosing the right chatbot framework

Go down a proof of concept path first. The core questions we need to ask ourselves:

* What is the business goal?
* How custom do we need the chatbot to be?
* How data-secure does the chatbot need to be?
* How much budget?
* How much time?

Need to choose a framework that makes the most sense for your objectives, and the best way to make an informed choice is to have a solid understanding of different frameworks.

### **Tables of Comparisons**

#### Comparison of different types of chatbot frameworks

Table 1: Different styles of chatbot frameworks

|  |  |  |  |
| --- | --- | --- | --- |
|  | **DIY Framework** | **Proof of Concept (PoF) Framework** | **Tailored Framework** |
| **Core Features** |  |  |  |
| NLP | Limited & Preset | Limited & Custom | Unlimited & Custom |
| Small talk | Limited & Preset | Limited & Custom | Unlimited & Custom |
| Exit feature | No | Yes | Yes |
| FAQs | Limited & Custom | Unlimited & Custom | Unlimited & Custom |
| Data Ownership | No | Not fully | Yes |
| **Advanced** |  |  |  |
| Machine Learning | No | No | yes |
| Sentiment Analysis | No | No | Yes |
| Authentication | No | Yes | Yes |
| Integrations | No | Yes | Yes |
| Webhooks | Yes | Yes | Yes |
| APIs | No | Yes | Yes |
| Voice | No | Yes | Yes |
| Mobile Marketing | Limited | Unlimited | Unlimited |
| **General** |  |  |  |
| Cost | None/Low | Medium | High |
| Support | None/Low | High | High |

### DIY chatbot framework

By using a platform that allows users to quickly and cheaply build ‘chatbots’ themselves

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| * Cheap | * Limited NLP |
| * Easy to use | * Limited functionalities |
| * Great for a quick mockup | * Limited (or nonexistent) support |
| * Great for wider technology adoption | * No data ownership |
|  |  |

### Proof of concept chatbot framework

Sort of the middle ground between the DIY framework (free/cheap but rigid and limited) and the tailored framework (built to the exact requirement, unlimited features & advanced AI, but requires budget)

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| * Cheaper than a tailored solution | * Not fully tailored |
| * Up and running in a matter of weeks | * Limited APIs/webhooks |
| * Focused on learning (speedier learning process) |  |
| * Metric-oriented (measure the specific metrics that matter to the business) |  |

### Tailored chatbot framework

Perfect for businesses that know what they want, have specific needs in terms of language APIs, data structure, reporting and more.

|  |  |
| --- | --- |
| **Pros** | **Cons** |
| * Suits precise needs | * Full investment in terms of time and money |
| * Performance-driven | * Planning |
| * Unique to you | * Long term project |
| * Own everything that goes in and out of the chatbot (data etc) |  |
| * Complete control |  |
| * Scalable |  |

#### **Comparison of specific frameworks**

**Refs:**<https://discover.bot/bot-talk/guide-to-bot-buiding-frameworks/>  
<https://medium.com/@NewBot/5-good-reasons-to-use-newbot-framework-5fee63839a8e>  
<https://www.spaceo.ca/top-ai-chatbot-frameworks/>   
<https://slides.com/samuelronce/newbot-framework-presentation-en>

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Microsoft Bot\* | Pandorabots | DialogFlow\* | IBM Watson Assistant | RASA\* | Wit.ai\* | Newbot |
| Developer Usability | Enterprise-grade, requires training | Requires some training | Easy to understand and learn, even for beginners | Easy to use with Watson Assistant GUI | Not suitable for beginners | Minimum training | Very easy to learn, no extensive training required |
| Support | Many tutorials & API support | Tutorials & API support | Tutorials & forum | Tutorials & forum | Tutorials & Active forum | Many tutorials & active forum | Few tutorials & no forum |
| Platform Limitations | Node.js, C# | Java, Node.js, Python, Ruby, PHP, Go | Node.js, Python, Java, Go, Ruby, C#, PHP | Java, Node.js, Python, Ruby, .NET, Android SDK, OpenWhisk SDK, Salesforce SDK, Unity SDK | HTTP API, Python | Node.js, Python, Ruby, Go, HTTP API | Node.js, JavaScript |
| Platforms/ Channels | Skype, Slack,  Messenger, Website, Cortana, Microsoft Team, Kik | Messenger, Slack, Line, Telegram, Viber, Twilio, WeChat, Twitter | Messenger, Google Assistant, Slack, Alexa, Cortana, Kik, Line, Skype, Telegram, Twilio, Viber | Messenger, Slack, Preview Link, Web Chat, Salesforce\*, Zendesk\*, Intercom\*,  WordPress, Custom app, Voice Agent (Telephony) | Website, Messenger, Slack, Telegram, Twilio, MS Bot, Cisco Webex Teams, RocketChat, Mattermost, Google Hangouts | Messenger, Website, Slack, (integrate into app with a wit.ai client), Wearable devices, Home automation (Smart Home) | Messenger, Google Assistant, Slack, Amazon Alexa, MS Teams, KiK, Telegram, Viber, Skype, Twitter, Twilio, Line, Cortana, GroupMe, Website (15 platforms) |
| Database Integration |  |  |  | No need (?) | (Container) PostgreSQL  Oracle >= 11.0 | No need (?) | Mostly works with JSON exports |
| Open Source | Customized code | Open standard | No | No | Yes | Yes | Yes |
| Active Development | No | No | No | No | Yes | Yes | Yes  (active [roadmap](https://github.com/newbotjs/newbot-framework/projects/1)) |
| Reusability | Yes |  | Yes | Yes | Yes | Yes | Yes (Modular) |
| Integrations (third-party) |  | Wolfram Alpha, custom API or DB | IoT Integration for home automation |  |  |  | DialogFlow, Wit.ai, Microsoft LUIS (MS Bot), RASA NLU, etc. |
| Usable offline | No | No | No | No | No | No | Yes |
| Unit Tests | No | No | No | No | No | No | Yes |
| Pricing | Free & Paid Version | Free | Free & Paid (if need to handle a lot of queries) | Free & Paid | Free & Paid (for enterprises that need heavy functionalities) | Free | Free |

#### Main Drawbacks:

The main problem with the Newbot Framework is that it does not have an active forum for questions. Currently, we can only open an issue on the Github project (open source) – for direct contact with the creators (according to previous questions, they seem to be answering quickly).

#### Main Advantages:

* Easily run a unit test per skill
* Universal code, not depending on the platform to be deployed
* Use with MS BotFramework/Google Actions
* Integrate external platforms with expertise in NLP:  
  DialogFlow, Wit.ai, Microsoft LUIS, RASA NLU etc

# Use with Bottender

Newbot uses Bottender for running the chatbot on multiple platforms.

Bottender is a modern chatbot framework for Node.js that supports multiple platforms such as Messenger, Slack, Telegram, LINE and Viber. It is optimized for production usage.