Why did Microsoft develop the Bot Framework?

While the Conversation User Interface (CUI) is upon us, at this point few developers have the expertise and tools needed to create new conversational experiences or enable existing applications and services with a conversational interface their users can enjoy. We have created the Bot Framework to make it easier for developers to build and connect great bots to users, wherever they converse, including on Microsoft's premier channels.

What is the v4 SDK?

Bot Builder v4 SDK builds on the feedback and learnings from the prior Bot Builder SDKs. It introduces the right levels of abstraction while enabling rich componentization of the bot building blocks. You can start with a simple bot and grow your bot in sophistication using a modular and extensible framework. You can find FAQ for the SDK on GitHub.

When will you add more conversation experiences to the Bot Framework?

We plan on making continuous improvements to the Bot Framework, including additional channels, but cannot provide a schedule at this time. If you would like a specific channel added to the framework, let us know.

I have a communication channel I’d like to be configurable with Bot Framework. Can I work with Microsoft to do that?

We have not provided a general mechanism for developers to add new channels to Bot Framework, but you can connect your bot to your app via the Direct Line API. If you are a developer of a communication channel and would like to work with us to enable your channel in the Bot Framework we’d love to hear from you.

If I want to create a bot for Skype, what tools and services should I use?

The Bot Framework is designed to build, connect, and deploy high quality, responsive, performant and scalable bots for Skype and many other channels. The SDK can be used to create text/sms, image, button and card-capable bots (which constitute the majority of bot interactions today across conversation experiences) as well as bot interactions which are Skype-specific such as rich audio and video experiences.

If you already have a great bot and would like to reach the Skype audience, your bot can easily be connected to Skype (or any supported channel) via the Bot Builder for REST API (provided it has an internet-accessible REST endpoint).

Do the bots registered with the Bot Framework collect personal information? If yes, how can I be sure the data is safe and secure? What about privacy?

Each bot is its own service, and developers of these services are required to provide Terms of Service and Privacy Statements per their Developer Code of Conduct. You can access this information from the bot’s card in the Bot Directory.

to provide the I/O service, the Bot Framework transmits your message and message content (including your ID), from the chat service you used to the bot.

Can I host my bot on my own servers?

Yes. Your bot can be hosted anywhere on the Internet. On your own servers, in Azure, or in any other datacenter. The only requirement is that the bot must expose a publicly-accessible HTTPS endpoint.

How do you ban or remove bots from the service?

Users have a way to report a misbehaving bot via the bot’s contact card in the directory. Developers must abide by Microsoft terms of service to participate in the service.

Which specific URLs do I need to whitelist in my corporate firewall to access Bot Framework services?

If you have an outbound firewall blocking traffic from your bot to the Internet, you'll need to whitelist the following URLs in that firewall:

- login.botframework.com (Bot authentication)
- login.microsoftonline.com (Bot authentication)
- westus.api.cognitive.microsoft.com (for Luis.ai NLP integration)
- state.botframework.com (Bot state storage for prototyping)
- cortanabfchanneleastus.azurewebsites.net (Cortana channel)
- cortanabfchannelwestus.azurewebsites.net (Cortana Channel)
- *.botframework.com (channels)

Can I block all traffic to my bot except traffic from the Bot Connector Service?

No. This sort of IP Address or DNS whitelisting is impractical. The Bot Framework Connector Service is hosted in Azure datacenters world-wide and the list of Azure IPs is constantly changing. Whitelisting certain IP addresses may work one day and break the next as the Azure IP Addresses change.

What keeps my bot secure from clients impersonating the Bot Framework Connector Service?

1. The security token accompanying every request to your bot has the ServiceUrl encoded within it, which means that even if an attacker gains access to the token, they cannot redirect the conversation to a new ServiceUrl. This is enforced by all implementations of the SDK and documented in our authentication reference materials.

2. If the incoming token is missing or malformed, the Bot Framework SDK will not generate a token in response. This limits how much damage can be done if the bot is incorrectly configured.

3. Inside the bot, you can manually check the ServiceUrl provided in the token. This makes the bot more fragile in the event of service topology changes so this is possible but not recommended.

Note that these are outbound connections from the bot to the Internet. There is not a list of IP Addresses or DNS names that the Bot Framework Connector Service will use to talk to the bot. Inbound IP Address whitelisting is not supported.