Contents

- Bot Framework SDK documentation
- Overview
  - What is the Bot Framework SDK?
  - What's new
- Quickstart
  - Create a bot using .NET
  - Create a bot using JavaScript
  - Create a bot using Python
  - Create a bot using Azure Bot Service
- Tutorials
  - 1. Create a basic bot
  - 2. Deploy a basic bot
  - 3. Add QnA Maker and redeploy a bot
- Samples
  - Bot Framework samples repo on GitHub
- Security
  - Security baseline
  - Bot Service encryption
  - Authentication basics
  - Authentication types
  - User authentication
    - User authentication basics
    - Identity providers
    - Single sign on
    - Add authentication to a bot
    - Add single sign on to a bot
  - Enhanced authentication
  - Security guidelines
- Concepts

- Bot
  - How bots work
  - Managing state
  - Event-driven conversations
  - Dialogs library
  - Waterfall dialogs
  - Middleware
  - Understand the structure of an echo bot
- Adaptive dialogs
  - Introduction to adaptive dialogs
  - Events and triggers in adaptive dialogs
  - Actions in Adaptive Dialogs
  - Asking for user input using adaptive dialogs
  - Recognizers in adaptive dialogs
  - Language Generation in adaptive dialogs
  - Memory scopes and managing state in adaptive dialogs
  - Handling interruptions in adaptive dialogs
  - Using declarative assets in adaptive dialogs
  - Cross training your LUIS and QnA Maker models
- Manage bot resources
- How bots for Microsoft Teams work
- Skills
  - About skills
  - About skill bots
  - About skill consumers
- Adaptive expressions
- Language Generation

How-To

- Design
  - Principles of bot design
  - First interaction
  - Design bot elements

- Design and control conversation flow
- Design bot navigation
- Design the user experience
- Patterns
  - Knowledge base
  - Handoff to human
  - Bots in apps
  - Bots in websites
- Develop
  - Send and receive text message
  - Add media to messages
  - Add buttons to guide user action
  - Save user and conversation data
  - Prompt users for input
  - Send welcome message to users
  - Send proactive notifications to users
  - Manage a long-running operation
  - Implement sequential conversation flow
  - Manage dialog complexity
  - Create advanced conversation flow using branches and loops
  - Handle user interruptions
  - Expire a conversation
  - Language understanding
    - LUIS
      - Add natural language understanding to your bot
      - Deploy LUIS resources using LUIS CLI commands
      - Update LUIS resources using LUIS CLI commands
    - QnA Maker
      - Answer user's questions using QnA Maker
      - Deploy QnA Maker knowledge base using CLI commands
      - Customized lists of synonyms in QnA Maker
    - Use multiple LUIS and QnA models

- Adaptive dialogs
  - Create a bot project for adaptive dialogs
  - Create a bot using adaptive dialogs
  - Create a bot using adaptive, component, waterfall, and custom dialogs
  - Handle user interruptions in adaptive dialogs
  - Create a bot using declarative adaptive dialogs
  - Cross train a bot to use both LUIS and QnA Maker
- Write directly to storage
- Implement custom storage for your bot
- Language Generation
  - Use language generation templates in your bot
  - Use custom functions in language generation
- Skills
  - Implement a skill
  - Write a v2.1 skill manifest
  - Write a v2.0 skill manifest
  - Use dialogs within a skill
  - Implement a skill consumer
  - Use a dialog to consume a skill
- Add telemetry to your bot
- Add telemetry to your QnA bot
- Analyze your bot's telemetry data
- Use Direct Line Speech in your bot

Test
- Unit testing bots
- Add trace activities to your bot

Debug
- Debug a bot using IDE
- Debug with the Bot Framework Emulator
- Debug a bot from any channel using ngrok
- Debug a skill or skill consumer
- Debug a bot with inspection middleware

- Debug your bot using transcript files
- Debug your bot with Adaptive Tools
- Deploy
  - Deploy your bot to Azure
  - Set up continuous deployment
- Manage
  - Manage a bot
  - Bot channels registration
  - Bot analytics
  - Configure bot settings
  - Configure speech priming
  - Channels
    - Connect a bot to channels
    - Implement channel-specific functionality
    - Cortana (retired)
    - Direct Line
      - About Direct Line
      - Connect to Direct Line
      - Connect to Direct Line Speech
      - Direct Line App Service Extension
        - Configure .NET bot for extension
        - Configure Node.js bot for extension
        - Create .NET Client with extension
        - Use extension with WebChat
        - Use extension within VNET
    - Alexa
    - Email
    - Facebook
    - GroupMe
    - Kik
    - LINE
    - Microsoft Teams