

warning "Mise à jour du backend (octobre 2026)" L'inférence GitHub Models a été retirée le 30 juillet 2026. Les références ci-dessous sont conservées uniquement comme contexte historique pendant la migration de ce lab. Utilisez le Lab 013 ? Backends LLM gratuits pour configurer Foundry Local ou Ollama avant de continuer.

Lab 024 : Bot Teams AI Library

Niveau : L200 Parcours : Agent Builder ? Teams Durée : ~60 min ? Coût : Gratuit ? Programme développeur Microsoft 365 (gratuit) + GitHub Models

Ce que vous apprendrez

Construire un bot Teams code-first avec la Teams AI Library (JavaScript) Connecter le bot à GitHub Models (pas besoin d'Azure OpenAI) Exécuter localement avec le Teams Toolkit et l'outil de test Teams App Gérer les messages utilisateur, les cartes adaptatives et les actions Ajouter des réponses alimentées par l'IA via le module AI

Introduction

La Teams AI Library est un SDK JavaScript/TypeScript pour construire des bots alimentés par l'IA qui fonctionnent nativement dans Microsoft Teams. Contrairement à Copilot Studio (no-code), la Teams AI Library vous donne un contrôle programmatique complet ? logique personnalisée, intégrations webhook, gestion d'état complexe.

Ce lab construit le bot de service client OutdoorGear pour Teams.

Prérequis

Node.js 18+ (node --version) Extension VS Code Teams Toolkit (installer depuis le marketplace VS Code) Locataire du Programme développeur Microsoft 365 (gratuit sur developer.microsoft.com/microsoft-365/dev-program) GITHUB_TOKEN configuré tip "Locataire développeur M365 gratuit" Le Programme développeur M365 vous donne un locataire sandbox gratuit de 90 jours avec Teams, SharePoint et toutes les applications Microsoft 365 ? renouvelable si actif.

Exercice du lab

Étape 1 : Créer le projet

[diagram or code sample]

Ou utilisez VS Code : 1. Ouvrez VS Code ? extension Teams Toolkit (barre latérale gauche) 2. Cliquez sur Create a New App ? Bot ? AI Bot ? JavaScript 3. Nom : OutdoorGearBot

Étape 2 : Structure du projet

[diagram or code sample]

Étape 3 : Configurer GitHub Models

Éditez src/config.js :

[diagram or code sample]

Éditez src/app.js :

[diagram or code sample]

Étape 4 : Écrire le prompt système

Éditez src/prompts/chat/skprompt.txt :

[diagram or code sample]

Éditez src/prompts/chat/config.json :

[diagram or code sample]

Étape 5 : Exécuter localement avec l'outil de test Teams App

[diagram or code sample]

Cela ouvre l'outil de test Teams App ? un simulateur dans le navigateur qui vous permet de tester votre bot sans le déployer dans le vrai Teams.

Essayez : What boots do you have? Tell me about your return policy I need a tent for 2 people in winter /reset

Étape 6 : Ajouter une réponse par carte adaptative

Les cartes adaptatives rendent les réponses du bot riches et interactives :

[diagram or code sample]

Architecture

[diagram or code sample]

Prochaines étapes

Agent Teams no-code : ? Lab 011 ? Copilot Studio Déployer sur Azure : ? Lab 028 ? Déployer sur Container Apps

Modernisation de Phase 3

warning "Verified 2026-10-27" Utilisez Microsoft 365 Agents SDK et Agents Toolkit. Validez les agents déclaratifs avec le manifeste v1.9. Gardez Copilot Studio trial et Agents Playground facultatifs.

[diagram or code sample]

Microsoft 365 Agents SDK Microsoft 365 Agents Toolkit Declarative agent schema

Kit d'accompagnement

Le build du site génère des téléchargements PDF et ZIP stables. Le ZIP contient le guide du lab et tous les fichiers d'accompagnement déclarés.

Télécharger le PDF généré Télécharger le ZIP généré

Supporting file: CHEATSHEET.md

Fichiers d'accompagnement: Lab 024 : Bot Teams AI Library

Commencez par le parcours local ou gratuit documenté. Consignez les hypothèses et les preuves de validation. Gardez les services hébergés facultatifs sauf exigence explicite du lab.

Supporting file: README.md

Lab 024 : Bot Teams AI Library

Utilisez ces fichiers avec le guide du lab et le kit généré.