Integrating BTP Gen AI Bot with Microsoft Teams

Overview

This guide walks you through integrating your Gen AI recommendation engine deployed on SAP BTP with Microsoft Teams using Azure Bot Framework.

BTP Endpoint: `https://ats-movie-recommend-fancy-serval-ik.cfapps.us10-001.hana.ondemand.com/recommend`

Prerequisites

- Azure subscription with admin access
- Microsoft Teams admin access
- SAP BTP deployed bot endpoint (already available)
- Node.js installed (v14 or higher) for middleware development
- Azure CLI installed (optional)

Architecture Overview

Microsoft Teams User
↓
Microsoft Teams Client
↓
Azure Bot Service
↓
Bot Middleware (Azure App Service/Function)
↓
SAP BTP Gen AI Bot Endpoint

Part 1: Register Bot in Azure Portal

Step 1: Navigate to Azure Portal

1. Go to https://portal.azure.com
2. Sign in with your Azure credentials
3. In the search bar at the top, type **"Azure Bot"**
4. Click on **"Azure Bot"** from the results

Show Image

Step 2: Create Azure Bot

1. Click **"+ Create"** button
2. Fill in the bot creation form:
   - **Bot handle:** `btp-genai-bot` (must be unique)
   - **Subscription:** Select your subscription
   - **Resource group:** Create new or select existing
   - **Location:** Choose nearest region
   - **Pricing tier:** F0 (Free) for testing

Show Image

3. Click **"Review + Create"**
4. Click **"Create"**
5. Wait for deployment to complete (2-3 minutes)

Step 3: Get Bot Credentials

1. Go to your newly created bot resource
2. Click on **"Configuration"** in the left menu
3. Note down:
   - **Microsoft App ID** (save this)
   - Click **"Manage Password"** to create app secret

Show Image

4. In the Certificates & secrets page:
   - Click **"+ New client secret"**
   - Add description: `Teams Integration`
   - Select expiry: 24 months
   - Click **"Add"**
   - **IMPORTANT:** Copy the secret value immediately (you won't see it again)

Show Image

Part 2: Create Middleware/Adapter Service

You need a middleware service to connect Azure Bot Service with your BTP endpoint. Here are two options:

Option A: Using Azure Functions (Recommended)

Step 4: Create Azure Function App

1. In Azure Portal, search for **"Function App"**
2. Click **"+ Create"**
3. Configure:
   - **Function App name:** `btp-bot-middleware`
   - **Runtime stack:** Node.js
   - **Version:** 18 LTS
   - **Region:** Same as bot
   - **Operating System:** Linux

Show Image

4. Click **"Review + Create"** → **"Create"**

Step 5: Deploy Middleware Code

1. Go to your Function App
2. Click **"Functions"** in left menu
3. Click **"+ Create"**
4. Select **"HTTP trigger"**
5. Name it: `messages`
6. Authorization level: **"Anonymous"** (we'll secure it differently)

Show Image

7. Click on the function and select **"Code + Test"**
8. Replace the code with:

javascript

```
const { BotFrameworkAdapter } = require('botbuilder');
const axios = require('axios');

// Bot credentials from environment variables
const adapter = new BotFrameworkAdapter({
    appId: process.env.MicrosoftAppId,
    appPassword: process.env.MicrosoftAppPassword
});

// BTP endpoint
const BTP_ENDPOINT = 'https://ats-movie-recommend-fancy-serval-ik.cfapps.us10-001.hana.ondemand.com/recommen

module.exports = async function (context, req) {
    // Process Bot Framework activity
    await adapter.processActivity(req, context.res, async (turnContext) => {
        if (turnContext.activity.type === 'message') {
            const userMessage = turnContext.activity.text;

            try {
                // Call BTP endpoint
                const response = await axios.post(BTP_ENDPOINT, {
                    query: userMessage,
                    user_id: turnContext.activity.from.id
                }, {
                    headers: {
                        'Content-Type': 'application/json'
                    },
                    timeout: 30000
                });

                // Send response back to user
                const botReply = response.data.recommendation || response.data.response || 'No recommendation available';
                await turnContext.sendActivity(botReply);

            } catch (error) {
                console.error('BTP API Error:', error);
                await turnContext.sendActivity('Sorry, I encountered an error processing your request. Please try again.');
            }
        } else if (turnContext.activity.type === 'conversationUpdate') {
            // Welcome message when bot is added
            if (turnContext.activity.membersAdded) {
```

```
            for (const member of turnContext.activity.membersAdded) {
                if (member.id !== turnContext.activity.recipient.id) {
                    await turnContext.sendActivity('Hello! I\'m your Gen AI recommendation bot. Ask me for movie recommer
                }
            }
        }
    }
  });
};
```

9. Click **"Save"**

Step 6: Configure Function App Settings

1. Go to Function App **"Configuration"**
2. Click **"+ New application setting"** and add:
    - **Name:** `MicrosoftAppId`
    - **Value:** [Your Bot's App ID from Step 3]
3. Add another setting:
    - **Name:** `MicrosoftAppPassword`
    - **Value:** [Your Bot's App Secret from Step 3]

Show Image

4. Click **"Save"** → **"Continue"**

Step 7: Install NPM Packages

1. In Function App, go to **"Console"** (under Development Tools)
2. Navigate to function directory:

```bash
cd messages
```

3. Create package.json:

```bash
npm init -y
```

4. Install dependencies:

bash

```
npm install botbuilder axios
```

Show Image

Step 8: Get Function URL

1. Go to your **"messages"** function
2. Click **"Get Function URL"**
3. Copy the URL (e.g., `https://btp-bot-middleware.azurewebsites.net/api/messages`)

Show Image

Part 3: Configure Bot Messaging Endpoint

Step 9: Update Bot Configuration

1. Go back to your **Azure Bot** resource
2. Click **"Configuration"**
3. In **"Messaging endpoint"** field, paste your Function URL from Step 8
4. Click **"Apply"**

Show Image

Part 4: Add Teams Channel

Step 10: Enable Teams Channel

1. In your Azure Bot, click **"Channels"** in left menu
2. Click on **"Microsoft Teams"** icon
3. Read and accept the terms
4. Select **"Microsoft Teams Commercial"**
5. Click **"Agree"**
6. Click **"Apply"**

Show Image

7. The Teams channel is now activated

Part 5: Test Bot in Teams

Step 11: Test in Web Chat First

1. In Azure Bot, click **"Test in Web Chat"**

2. Type a test message: "Recommend me a sci-fi movie"
3. Verify the bot responds with recommendations from your BTP endpoint

Show Image

Step 12: Open Bot in Teams

1. Go back to **"Channels"**
2. Click on **"Microsoft Teams"** channel
3. Click **"Open in Teams"** button

Show Image

4. Microsoft Teams will open (web or desktop app)
5. Click **"Add"** to add the bot

Show Image

Step 13: Chat with Bot

1. Start a conversation with your bot
2. Type: "Recommend a movie for me"
3. Bot should respond with recommendations from your BTP Gen AI engine

Show Image

Part 6: Deploy Bot to Organization (Optional)

Step 14: Create App Package

1. Create a folder named `TeamsAppPackage`
2. Create three files inside:

manifest.json:


json

```
{
 "$schema": "https://developer.microsoft.com/json-schemas/teams/v1.16/MicrosoftTeams.schema.json",
 "manifestVersion": "1.16",
 "version": "1.0.0",
 "id": "YOUR-MICROSOFT-APP-ID",
 "packageName": "com.btp.genaibot",
 "developer": {
  "name": "Your Organization",
  "websiteUrl": "https://www.yourorg.com",
  "privacyUrl": "https://www.yourorg.com/privacy",
  "termsOfUseUrl": "https://www.yourorg.com/terms"
 },
 "name": {
  "short": "BTP Gen AI Bot",
  "full": "BTP Gen AI Recommendation Bot"
 },
 "description": {
  "short": "AI-powered recommendation engine",
  "full": "Gen AI bot deployed on SAP BTP for intelligent recommendations"
 },
 "icons": {
  "outline": "outline.png",
  "color": "color.png"
 },
 "accentColor": "#FFFFFF",
 "bots": [
  {
   "botId": "YOUR-MICROSOFT-APP-ID",
   "scopes": [
    "personal",
    "team",
    "groupchat"
   ],
   "supportsFiles": false,
   "isNotificationOnly": false
  }
 ],
 "permissions": [
  "identity",
  "messageTeamMembers"
 ],
```

```
  "validDomains": [
    "*.azurewebsites.net",
    "*.hana.ondemand.com"
  ]
}
```

3. Add two icon files:
   - **color.png** (192x192 pixels)
   - **outline.png** (32x32 pixels, transparent background)
4. Zip all three files (manifest.json, color.png, outline.png)

Step 15: Upload to Teams Admin Center

1. Go to Teams Admin Center
2. Navigate to **"Teams apps"** → **"Manage apps"**
3. Click **"+ Upload new app"**
4. Upload your zip file
5. Click **"Submit"**

Show Image

Step 16: Approve and Publish

1. Go to **"Pending approval"**
2. Find your bot
3. Click **"Approve"**
4. Set availability (Specific users/groups or Everyone)

Show Image

Troubleshooting

Bot Not Responding

1. Check Azure Function logs:
   - Go to Function App → Monitor → Logs
   - Look for errors
2. Verify BTP endpoint is accessible:

bash

```
curl -X POST https://ats-movie-recommend-fancy-serval-ik.cfapps.us10-001.hana.ondemand.com/recommend \
-H "Content-Type: application/json" \
-d '{"query": "test"}'
```

3. Check messaging endpoint in Bot Configuration

Authentication Errors

1. Verify App ID and Secret in Function App configuration
2. Regenerate secret if needed
3. Update Function App settings

Teams Channel Not Working

1. Re-enable Teams channel
2. Check bot is published in Teams Admin Center
3. Clear Teams cache

Security Enhancements

Add Authentication to BTP Endpoint

1. Update Function code to include auth headers:

javascript

```
const response = await axios.post(BTP_ENDPOINT, {
  query: userMessage
}, {
  headers: {
    'Content-Type': 'application/json',
    'Authorization': `Bearer ${process.env.BTP_API_KEY}`
  }
});
```

2. Add API key to Function App settings

Enable Bot Service Authentication

1. In Azure Bot → Configuration
2. Enable **"OAuth Connection Settings"**
3. Configure identity provider

Monitoring and Logs

View Bot Analytics

1. Azure Bot → **Analytics**
2. Monitor:
    - Active users
    - Messages sent/received
    - Channel usage

Function App Monitoring

1. Function App → **Monitor**
2. Check:
    - Execution count
    - Success rate
    - Response time

Application Insights

1. Enable Application Insights for Function App
2. View detailed telemetry and error tracking

Cost Considerations

- **Azure Bot:** F0 tier (Free) - 10,000 messages/month
- **Function App:** Consumption plan - First 1M executions free
- **BTP:** Based on your SAP contract

Next Steps

1. **Enhance bot capabilities:**
    - Add rich cards for recommendations
    - Implement conversation state
    - Add multi-turn dialogues
2. **Deploy to other channels:**
    - Slack
    - Web chat widget
    - Mobile apps
3. **Add advanced features:**
    - User preferences storage
    - Recommendation history
    - Feedback collection

Support Resources

- Azure Bot Service Documentation
- Microsoft Teams Platform
- SAP BTP Documentation

Summary

You have successfully integrated your SAP BTP Gen AI bot with Microsoft Teams. Users can now interact with your recommendation engine directly from Teams using natural conversation.