🚀 Real Teams Integration Testing Guide

Step 1: Update Azure Permissions

In Azure Portal:

1. Go to **Azure Portal** → **App Registrations** → Your App
2. Click **API Permissions** → **Add Permission** → **Microsoft Graph**
3. **Add these Application Permissions:**

   ✅ User.Read.All
   ✅ Chat.ReadWrite.All
   ✅ ChatMessage.Send
   ✅ People.Read.All
   ✅ Directory.Read.All

4. **Click "Grant admin consent"** for your organization
5. **Wait 5-10 minutes** for permissions to propagate

Step 2: Test Real User Search

Find Team Members by Name:

```bash
GET http://localhost:5000/api/meetings/team-members/search?q=John
```

Expected Response (Real Integration):

```json
```

```
{
  "success": true,
  "query": "John",
  "found": 2,
  "users": [
    {
      "id": "12345-abcde-67890",
      "name": "John Smith",
      "email": "john.smith@yourcompany.com",
      "jobTitle": "Software Engineer",
      "department": "Engineering"
    },
    {
      "id": "54321-edcba-09876",
      "name": "John Doe",
      "email": "john.doe@yourcompany.com",
      "jobTitle": "Product Manager",
      "department": "Product"
    }
  ]
}
```


Get All Team Members:

```
bash

GET http://localhost:5000/api/meetings/team-members?limit=20
```

Step 3: Test User Resolution

Check if Names Resolve to Real Users:

```
bash

POST http://localhost:5000/api/meetings/test-user-resolution
Content-Type: application/json

{
  "names": ["John Smith", "Sarah Johnson", "Mike Wilson"]
}
```

Expected Response (Working):

```json
{
  "mode": "real_teams",
  "realUserLookup": true,
  "input": ["John Smith", "Sarah Johnson", "Mike Wilson"],
  "foundUsers": [
    {
      "name": "John Smith",
      "email": "john.smith@yourcompany.com"
    },
    {
      "name": "Sarah Johnson",
      "email": "sarah.johnson@yourcompany.com"
    }
  ],
  "summary": {
    "requested": 3,
    "found": 2,
    "success": true
  },
  "message": "✅ Found 2/3 real Teams users!"
}
```

Step 4: Send Messages to Team Members

Send a Test Message:

```bash
POST http://localhost:5000/api/meetings/send-message
Content-Type: application/json

{
  "email": "john.smith@yourcompany.com",
  "message": "🤖 Hello! This is a test message from the AI Agent. The Teams integration is working! 🎉"
}
```

Expected Response:

```json
```

```
{
  "success": true,
  "result": {
    "success": true,
    "recipient": {
      "name": "John Smith",
      "email": "john.smith@yourcompany.com"
    },
    "chatId": "19:abc123def456_chat_id",
    "message": "Message sent successfully"
  },
  "message": "Message sent successfully to John Smith"
}
```


What the Recipient Sees:

- Gets a Teams chat message from your bot
- Can reply directly in Teams
- Shows your app name as sender

Step 5: Create Meeting with Real Users

Create Meeting by Names (Auto-Resolve):

```bash
POST http://localhost:5000/api/meetings/create-with-real-users
Content-Type: application/json

{
  "subject": "Team Sync with Real Users",
  "description": "Testing real Teams integration with auto-resolved users",
  "startTime": "2025-08-10T19:00:00.000Z",
  "endTime": "2025-08-10T19:30:00.000Z",
  "attendeeNames": ["John Smith", "Sarah Johnson"],
  "attendeeEmails": ["mike.wilson@yourcompany.com"],
  "sendInviteMessages": true,
  "autoJoinAgent": true,
  "enableChatCapture": true
}
```

Expected Response:

json


```
{
  "success": true,
  "meeting": {
    "id": "meeting-uuid",
    "subject": "Team Sync with Real Users",
    "attendees": [
      "john.smith@yourcompany.com",
      "sarah.johnson@yourcompany.com",
      "mike.wilson@yourcompany.com"
    ],
    "joinUrl": "https://teams.microsoft.com/l/meetup-join/...",
    "agentAttended": true,
    "status": "in_progress"
  },
  "realUserResolution": {
    "available": true,
    "namesRequested": 2,
    "usersResolved": 2,
    "messagesAttempted": 3,
    "messagesSent": 3,
    "inviteResults": [
      {
        "email": "john.smith@yourcompany.com",
        "status": "sent"
      },
      {
        "email": "sarah.johnson@yourcompany.com",
        "status": "sent"
      },
      {
        "email": "mike.wilson@yourcompany.com",
        "status": "sent"
      }
    ]
  },
  "message": "🚀 Real Teams meeting created with real user resolution!"
}
```


Step 6: What Real Users Experience

Attendees Receive:

1. **Calendar Invite** (standard Teams meeting invite)
2. **Teams Chat Message**:

```
🎯 Meeting Invitation: Team Sync with Real Users

🗓️ When: August 10th 2025, 7:00 PM
⏱️ Duration: 30 minutes

🔗 Join Link: https://teams.microsoft.com/l/meetup-join/...

Hope to see you there! 🤖
```

During the Meeting:

- **Real attendees** join via calendar or link
- **AI Agent** joins automatically
- **Chat messages** are captured and analyzed in real-time
- **Action items** are automatically detected

After the Meeting:

- **Meeting summary** generated with real participant data
- **Action items** assigned to real people
- **Follow-up messages** can be sent automatically

Step 7: Advanced Features Testing

Monitor Real Chat:

```bash
# Get real chat analysis
GET http://localhost:5000/api/meetings/{meeting-id}/chat-analysis
```

Send Follow-up Messages:

```bash
```

```
POST http://localhost:5000/api/meetings/send-message
{
  "email": "john.smith@yourcompany.com",
  "message": "📋 Action Item Reminder: Please complete the quarterly report by Friday. Thanks!"
}
```

Generate Meeting Summary with Real Data:

```
bash

GET http://localhost:5000/api/meetings/{meeting-id}/summary
```

Troubleshooting

Issue: "Teams service not available"

Fix: Check Azure permissions and ensure admin consent is granted

Issue: "No users found"

Test: Try searching for yourself first:

```
bash

GET http://localhost:5000/api/meetings/team-members/search?q=YourFirstName
```

Issue: "Message sending failed"

Check: User exists and has Teams enabled:

```
bash

POST http://localhost:5000/api/meetings/test-user-resolution
{
  "names": ["Your Name"]
}
```

Success Criteria ✅

When everything works, you should be able to:

- ✅ **Search team members** by name
- ✅ **Create meetings** with "John Smith" instead of emails

- ✅ **Send Teams messages** to any colleague
- ✅ **Auto-invite participants** with personalized messages
- ✅ **Capture real chat** from actual meetings
- ✅ **Generate summaries** with real participant data

Example Real Workflow

```bash
# 1. Find team members
GET /api/meetings/team-members/search?q=Sarah

# 2. Create meeting with names
POST /api/meetings/create-with-real-users
{
  "subject": "Sprint Planning",
  "attendeeNames": ["Sarah Johnson", "Mike Chen"],
  "sendInviteMessages": true
}

# 3. Meeting auto-starts, agent joins, chat captured

# 4. Send follow-up
POST /api/meetings/send-message
{
  "email": "sarah.johnson@company.com",
  "message": "Thanks for the great meeting! Action items sent via email."
}
```

Result: Full Teams integration with real users! 🎉