

AUTOMATE **Advanced** 60 MIN · UPDATED 2026-09-30

Expose a line-of-business system inside the Copilot canvas.

Build a SharePoint Framework component — like an inventory lookup or a PTO request form — that renders directly inside a Copilot conversation instead of sending people to another app.

MICROSOFT 365

SharePoint

WORKS WITH

Copilot

Business Outcome

Fewer app switches for routine line-of-business tasks, since people can act without leaving the Copilot conversation.

Workflow Steps

Step 1: Pick a narrow, high-frequency task

SharePoint

Pull usage data from the underlying system (ticket volume by type, form submission counts) to find the single interaction people do most often — checking inventory on an item, submitting a PTO request, looking up a ticket status — and scope the first component to exactly that one action.

Trying to surface an entire system's functionality in one build multiplies the testing and permission work in steps 3-4 for marginal benefit.

Step 2: Build the SharePoint Copilot App component

SharePoint

Have a developer build the SPFx-based component that connects to the underlying system's API and renders as an interactive card inside the Copilot conversation, following Microsoft's SharePoint Copilot Apps pattern.

PROMPT

Show me the current inventory level for SKU 4471-B and let me submit a restock request directly from here if it's below 20 units.

Step 3: Test the interaction inside an actual conversation

Copilot

Confirm the component renders correctly, the data it displays is live (not cached from build time), and any action a user takes (like submitting a request) actually reaches the underlying system.

Step 4: Roll out to the intended audience with the right permissions

SharePoint

Publish the component scoped to the audience who should have access, matching whatever permission model the underlying line-of-business system already enforces — don't let the Copilot surface become a way around existing access controls.

✓ Check the work

- Confirm displayed data (inventory counts, ticket status) matches the source system in real time, not a stale cache.
- Verify an action taken through the component (a submitted request, an update) actually lands in the underlying system by checking there directly.
- Test with a user account that has restricted permissions in the source system to confirm the component respects that restriction.



Expected Outcome

A working interactive component (like a PTO request form or an inventory lookup) that renders and functions inside a live Copilot conversation.

Want help rolling this out across your team? **Go to Liminals.Tech** ■