Ramin Ahmadi

Content and Cloud

CERTIFICATION
Microsoft 365 Certified: Teams Application Developer Associate

CERTIFICATION
Microsoft Certified: Power Platform App Maker Associate

CERTIFICATION
MCSD: App Builder Charter Member

CERTIFICATION
MCSA: Web Applications

CERTIFICATION
Microsoft Certified Professional

CERTIFICATION
Microsoft Specialist: Programming in HTML5 with JavaScript and CSS3

CERTIFICATION
Microsoft Certified Solutions Developer: SharePoint Applications (2013)

CERTIFICATION
Microsoft® Certified Technology Specialist: SharePoint 2010, Application Development

CERTIFICATION
Microsoft® Certified Professional Developer: SharePoint Developer 2010

CERTIFICATION
Microsoft® Certified Technology Specialist: Microsoft Windows SharePoint Services 3.0, Application Development

https://ramin.expert

@raminahmadi1986

/ahmadiramin

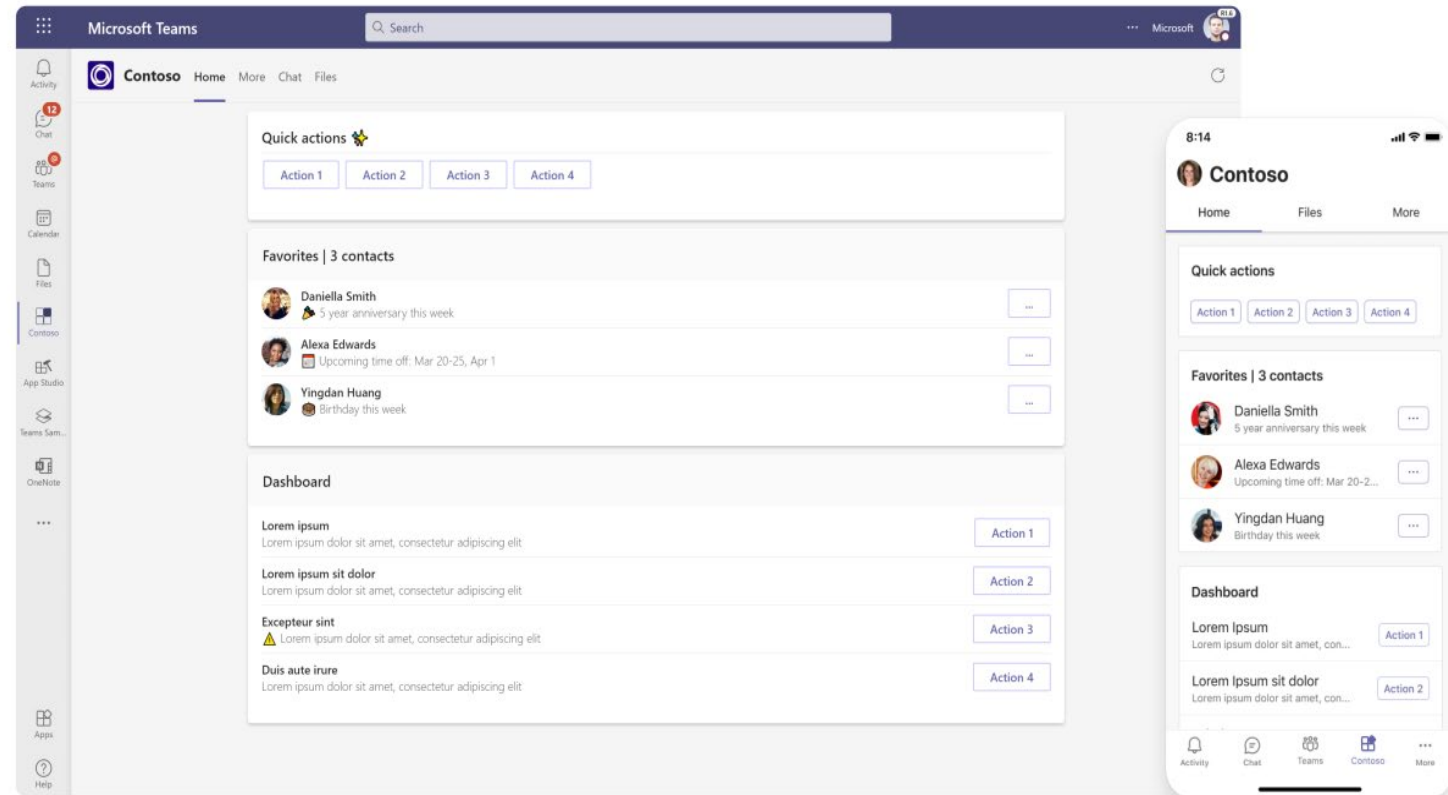/ahmadiramin

/ramin-ahmadi

Traditional methods to build tabs

- SharePoint Framework
- Yeoman generator for Teams
- Teams toolkit

- Considerations
  - HTML and CSS
  - Mobile responsiveness
  - Slow load times
  - Iframe constraints
  - Server maintenance and costs

YEOMAN

Build tabs with Adaptive Cards

- Ready-made UI building blocks
- Native UI on Desktop/Web/Mobile
- Use a bot as the backend
- The bot is responsible for accepting requests and responding
- Task modules

Prerequisites

- Be familiar with bot development, Adaptive Cards, and task modules in Teams
- Register a bot
- Configure SSO (if login is required)

Limitations

- Payload size is limited to **80KB**
- Limited adaptive cards features
- No advanced layout controls
- Only personal tabs are supported

Developer experience

Adaptive Cards

| Chat bot | Messaging extension | Task modules | Tabs |

Bot backend

Changes to app manifest

```json
"staticTabs": [
    {
        "entityId": "homeTab",
        "name": "Home",
        "contentBotId": "{{BOT-ID}}",
        "scopes": [ "personal" ]
    },
    {
        "entityId": "youtubeTab",
        "name": "YouTube",
        "contentBotId": "{{BOT-ID}}",
        "scopes": [ "personal" ]
    }
],
```

How does it work?

```json
JSON

// tab/fetch POST request: agents/{botId}/invoke
{
    "name": "tab/fetch",
    "value: {
        "tabContext": {
            "tabEntityId": "{tab_entity_id}"
        },
        "context": {
            "theme": "default"
        }
    },
    "conversation": {
        "id": "{generated_conversation_id}"
    },
    "imdisplayname": "{user_display_name}"
}
```

```json
JSON

// tab/fetch **continue** POST response:
{
    "tab": {
        "type": "continue",
        "value": {
            "cards": [
                {
                    "card": adaptiveCard1,
                },
                {
                    "card": adaptiveCard2,
                },
                {
                    "card": adaptiveCard3
                }
            ]
        },
    },
    "responseType": "tab"
}
```

S / PnP

Useful links

- [Source code](#)
- [Build tabs with Adaptive Cards](#)
- [Configure SSO for your bot](#)
- [Microsoft Search API in Microsoft Graph](#)
- [Adaptive cards designer](#)

THANK YOU