APVEE SOLUTIONS

Using Adaptive Cards control from the PnP control gallery in SPFx solutions

Fabio Franzini – CEO Apvee Solutions – Microsoft MVP

Agenda

- Understanding the **reason behind the implementation of this control**
- Why I chose to use the **Adaptive Cards SDK**
- **Introduction to the control**, how it was implemented and how to use it in SPFx
- **Same code**, different scenarios
- Demo
- Next Steps
- References

Understanding the reason behind the implementation of this control

- Very often, in my development activities, customers ask me SPFx components with identically functionality, but which may differ for the UI part.
- This led me to think of a way to have the ability to "change" the UI at runtime, without having to recompile the SPFx solution.
- After several attempts and reasoning, the choice fell on Adaptive Cards.

Why I chose to use the Adaptive Cards SDK

*Adaptive Cards are **platform-agnostic** snippets of UI, **authored in JSON**, that apps and services can openly exchange. When delivered to a specific app, the **JSON is transformed into native UI** that **automatically adapts to its surroundings**. It helps design and integrate **light-weight UI for all major platforms and frameworks**.*

The goals for Adaptive Cards are:

- **Portable** - To any app, device, and **UI framework**
- **Open** - Libraries and schema are **open source** and shared
- **Low cost** - **Easy to define**, easy to consume
- **Expressive** - Targeted at the long tail of content that developers want to produce
- **Purely declarative** - **No code** is needed or allowed
- **Automatically styled** - To the **Host application UX** and brand guidelines

APVEE SOLUTIONS

Introduction to the control, how it was implemented and how to use it in SPFx

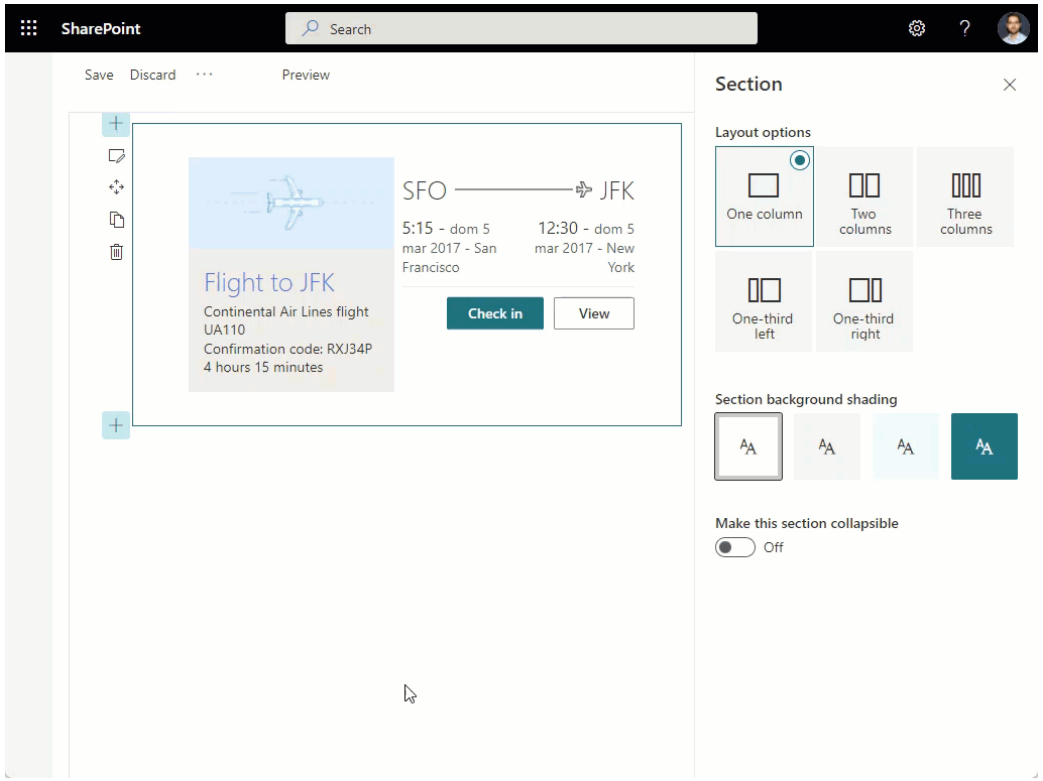
- React Control that render an Adaptive Card using the SDK
- Using a brand-new elements that use the Fluent UI Theme both for SharePoint and Microsoft Teams
- Auto apply the AC Template if data object is passed to the control
- Callbacks to intercept Actions and Errors
- Callbacks to set Custom Elements and Custom Actions
- Callback to set custom Host Capability properties

```
return (
  <AdaptiveCardHost
    card={sample.template}
    data={sampleData}
    style={null}
    className={null}
    theme={this.props.theme}
    themeType={themeType}
    hostConfig={null}
    onInvokeAction={(action: IAdaptiveCardHostActionResult) => alert(JSON.stringify(action))}
    onError={(error) => alert(error.message)}
    onSetCustomElements={(registry: CardObjectRegistry<CardElement>) => { }}
    onSetCustomActions={(registry: CardObjectRegistry<Action>) => { }}
    onUpdateHostCapabilities={(hostCapabilities: HostCapabilities) => {
      hostCapabilities.setCustomProperty("__customProperty", Date.now);
    }}
    isUniqueControlInPage={true}
  />
);
```

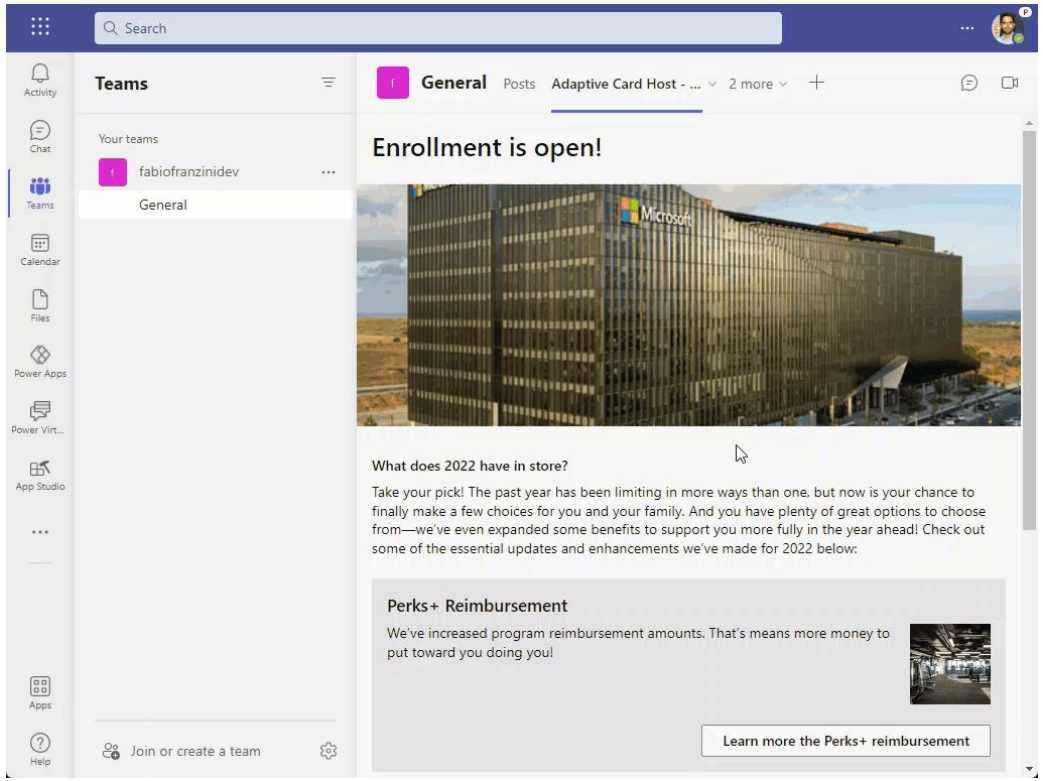
APVEE SOLUTIONS

Same code, different scenarios

SharePoint - SPFx Web Part



Teams Tab - SPFx Web Part



DEMO

Next Steps

- Implement the Designer component for the PnP Reusable React Controls Library
- Create a set of new Adaptive Cards Elements (targeted only for JS SDK)
  - Maybe using Fluent UI React
  - Maybe using Fluent UI Web Components
  - Maybe using MGT (Thanks to João Mendes)
  - Maybe using another libraries

About Me

I'm CEO at Apvee Solutions, startup focused on building innovative solutions using the Microsoft 365 ecosystem.

I'm a Microsoft MVP in Office Development and Business Applications categories.

- Mail: fabio@apvee.com
- Twitter: @franzinifabio
- LinkedIn: www.linkedin.com/in/fabiofranzini
- GitHub: https://github.com/fabiofranzini

References

- https://adaptivecards.io/
- https://pnp.github.io/sp-dev-fx-controls-react/controls/AdaptiveCardHost/
- https://github.com/pnp/sp-dev-fx-webparts/tree/main/samples/react-adaptive-card-host-control

APVEE SOLUTIONS

Thank you!

Fabio Franzini – CEO Apvee Solutions – Microsoft MVP