BY COMMUNITY • FOR COMMUNITY •

#GLOBALAZURE

VIRTUAL

2020

TRA POCO INIZIA IL LIVE

ARXivar ACTIVATE your business

CloudGen

Microsoft

FABIOFRANZINI.COM

Adaptive Cards Deep Dive

Fabio Franzini – Microsoft MVP

#GlobalAzureVirtual2020, April 24th, 2020

Agenda

1. Overview
2. Getting Started
3. Authoring Cards
4. Play with SDK
5. Templating (Preview)
6. Resources

Overview

Adaptive Cards are an **open card exchange format** enabling developers to **exchange UI content** in a common and consistent way.

Card **Authors describe** their **content as a simple JSON** object then be **rendered natively** inside a Host Application **with the look and feel of the Host.**

Core Design Principles

FABIOFRANZINI.COM

Semantic instead of pixel-perfect

Card **Authors** own the **content**, **Host** App owns the **look and feel**

Keep it simple, but expressive

Goals

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

Supported Platforms

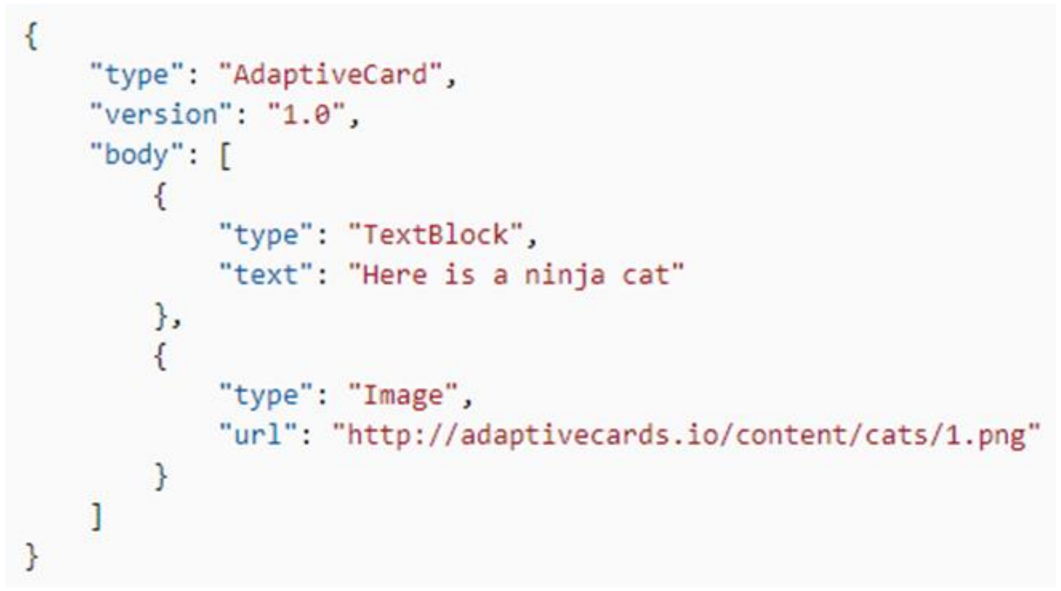
- Bot Framework WebChat
- Cortana Skills
- Windows Timeline
- Outlook Actionable Messages
- Microsoft Teams

DEMO

- http://contososcubademo.azurewebsites.net/

Authoring Cards

Adaptive Card structure



```
{
    "type": "AdaptiveCard",
    "version": "1.0",
    "body": [
        {
            "type": "TextBlock",
            "text": "Here is a ninja cat"
        },
        {
            "type": "Image",
            "url": "http://adaptivecards.io/content/cats/1.png"
        }
    ]
}
```

- AdaptiveCard - The root object describes the AdaptiveCard itself, including its element makeup, its actions, how it should be spoken, and the schema version required to render it.
- Body - The body of the card is made up of building-blocks known as elements. Elements can be composed in nearly infinite arrangements to create many types of cards.
- Actions - Many cards have a set of actions a user may take on it. This property describes those actions which typically get rendered in an "action bar" at the bottom.

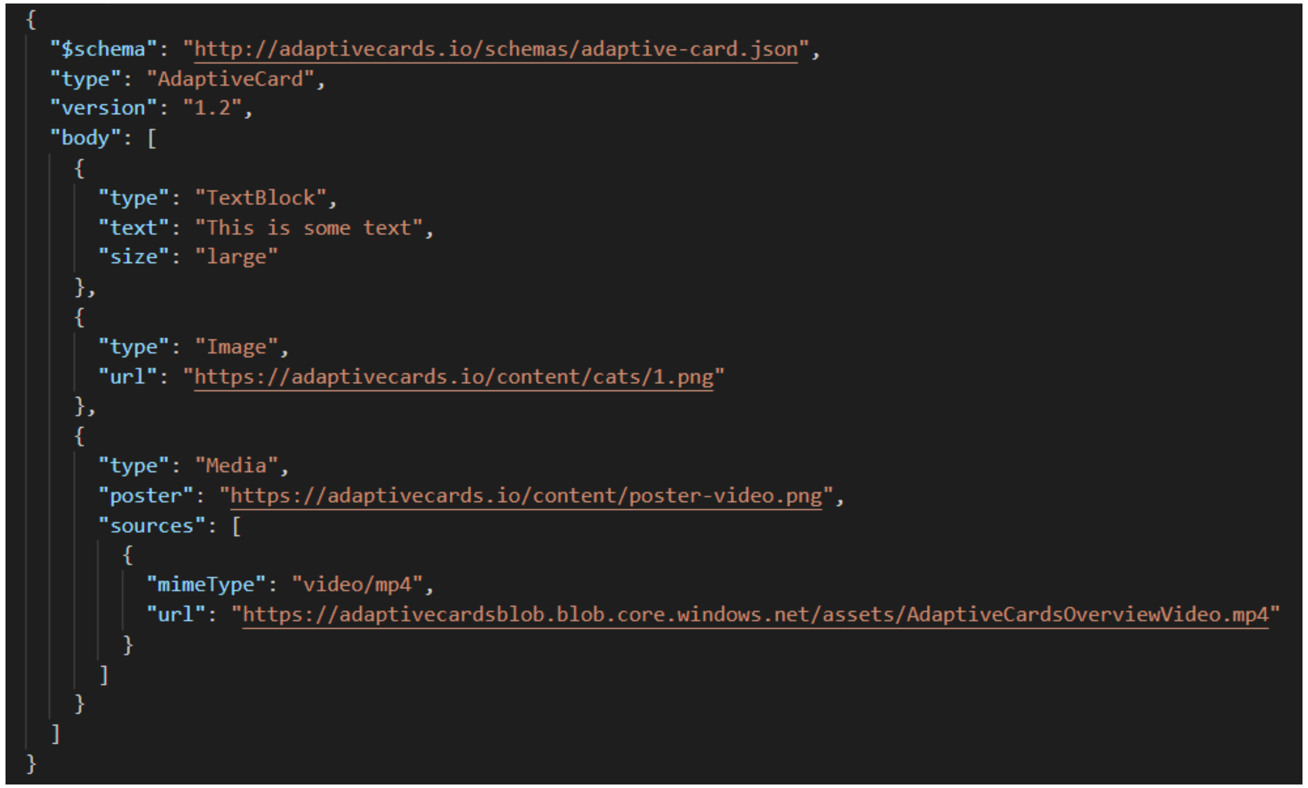
Elements (@version 1.2)

https://adaptivecards.io/explorer/

Elements	Containers	Actions	Inputs
• TextBlock	• ActionSet	• Action.OpenUrl	• Input.Text
• Image	• Container	• Action.Submit	• Input.Number
• Media	• ColumnSet	• Action.ShowCard	• Input.Date
• MediaSource	• Column	• Action.ToggleVisibility	• Input.Time
• RichTextBlock	• FactSet	• TargetElement	• Input.Toggle
• TextRun	• Fact		• Input.ChoiceSet
	• ImageSet		• Input.Choice

Elements

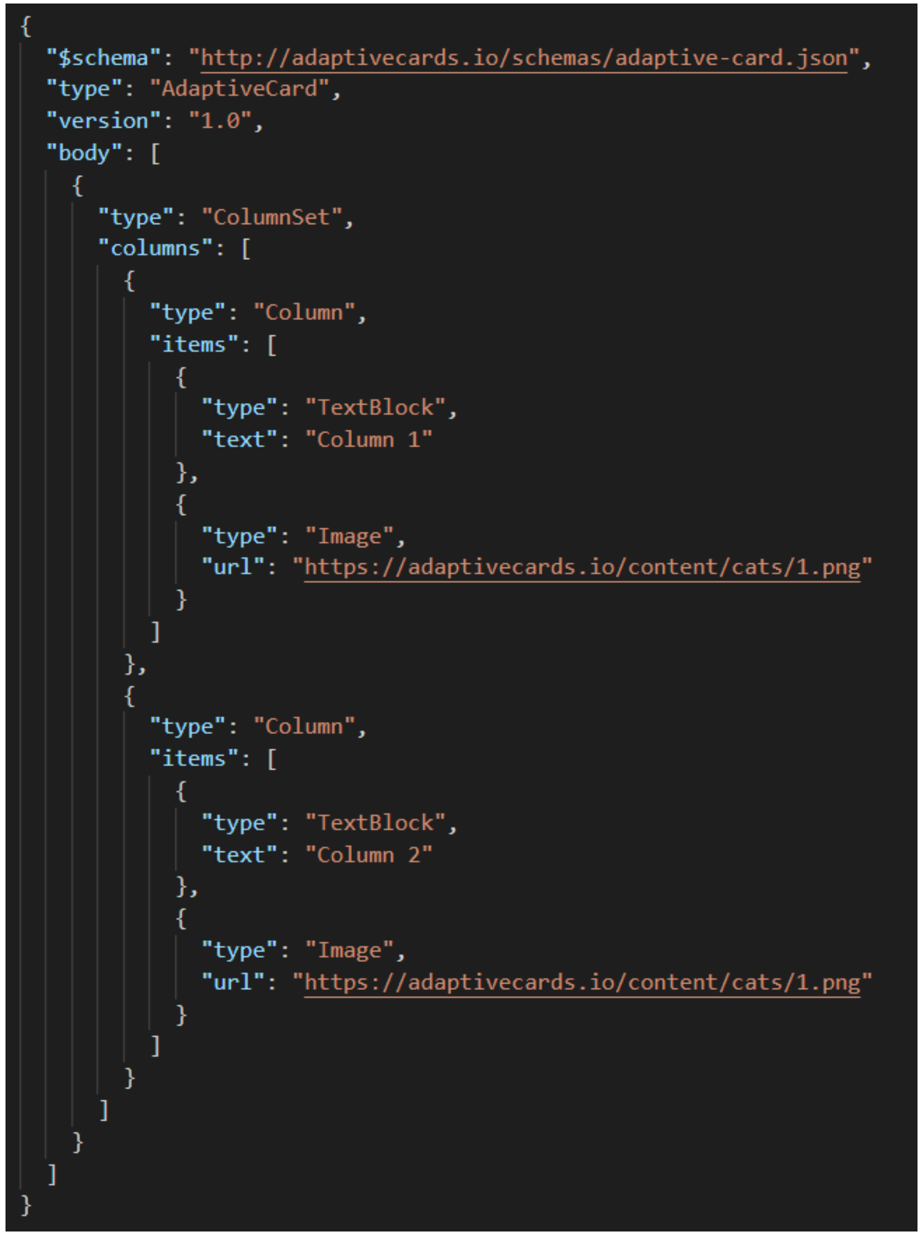
Just an example of using elements



```
{
  "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
  "type": "AdaptiveCard",
  "version": "1.2",
  "body": [
    {
      "type": "TextBlock",
      "text": "This is some text",
      "size": "large"
    },
    {
      "type": "Image",
      "url": "https://adaptivecards.io/content/cats/1.png"
    },
    {
      "type": "Media",
      "poster": "https://adaptivecards.io/content/poster-video.png",
      "sources": [
        {
          "mimeType": "video/mp4",
          "url": "https://adaptivecardsblob.blob.core.windows.net/assets/AdaptiveCardsOverviewVideo.mp4"
        }
      ]
    }
  ]
}
```

Containers

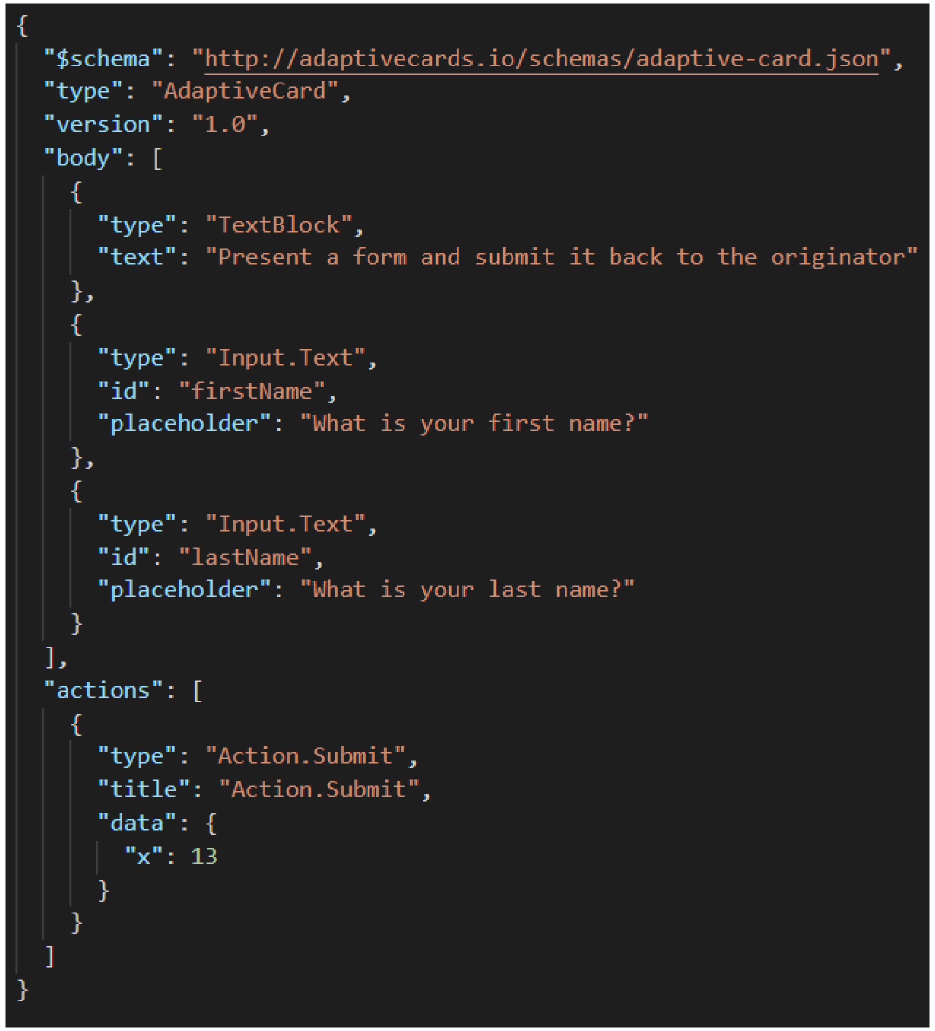
Just an example of using containers



```json
{
  "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
  "type": "AdaptiveCard",
  "version": "1.0",
  "body": [
    {
      "type": "ColumnSet",
      "columns": [
        {
          "type": "Column",
          "items": [
            {
              "type": "TextBlock",
              "text": "Column 1"
            },
            {
              "type": "Image",
              "url": "https://adaptivecards.io/content/cats/1.png"
            }
          ]
        },
        {
          "type": "Column",
          "items": [
            {
              "type": "TextBlock",
              "text": "Column 2"
            },
            {
              "type": "Image",
              "url": "https://adaptivecards.io/content/cats/1.png"
            }
          ]
        }
      ]
    }
  ]
}
```

Inputs & Actions

Just and example of using Inputs & Actions



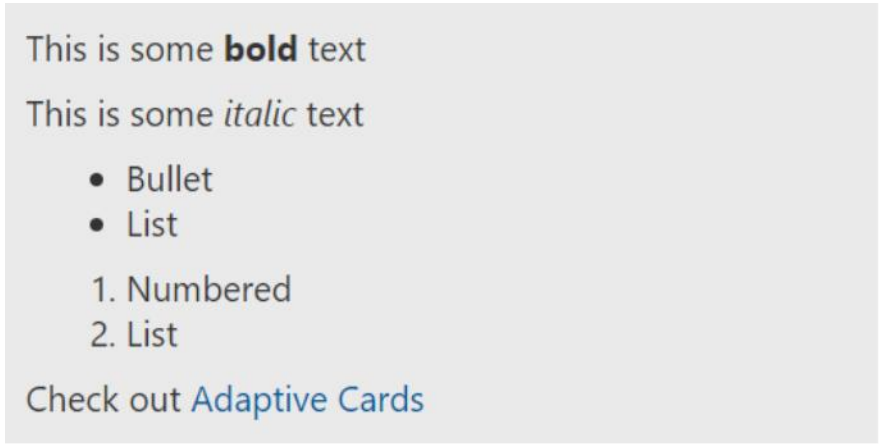
```
{
  "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
  "type": "AdaptiveCard",
  "version": "1.0",
  "body": [
    {
      "type": "TextBlock",
      "text": "Present a form and submit it back to the originator"
    },
    {
      "type": "Input.Text",
      "id": "firstName",
      "placeholder": "What is your first name?"
    },
    {
      "type": "Input.Text",
      "id": "lastName",
      "placeholder": "What is your last name?"
    }
  ],
  "actions": [
    {
      "type": "Action.Submit",
      "title": "Action.Submit",
      "data": {
        "x": 13
      }
    }
  ]
}
```

FABIOFRANZINI.COM

Text features

TextBlock offers some advanced features for formatting and localizing the text:

- Adaptive Cards support a subset of Markdown syntax.
- Date/Time formatting and localization



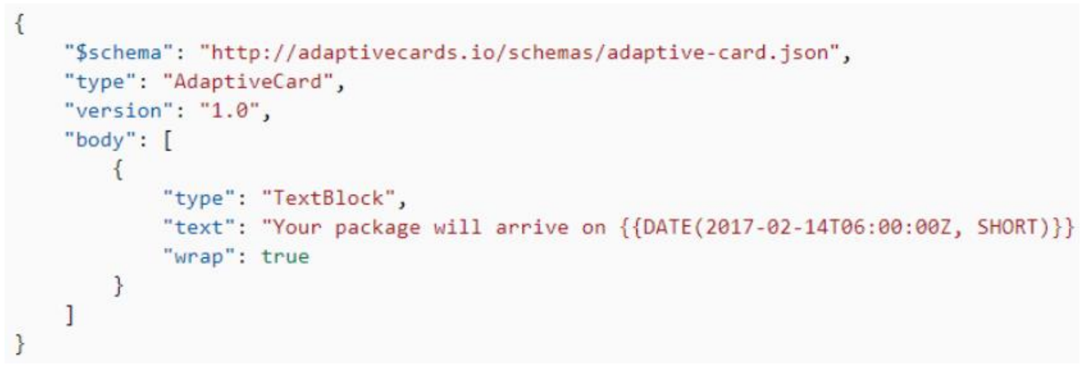
This is some **bold** text

This is some *italic* text

- Bullet
- List

1. Numbered
2. List

Check out Adaptive Cards



```
{
    "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
    "type": "AdaptiveCard",
    "version": "1.0",
    "body": [
        {
            "type": "TextBlock",
            "text": "Your package will arrive on {{DATE(2017-02-14T06:00:00Z, SHORT)}}
            "wrap": true
        }
    ]
}
```

DEMO

- https://adaptivecards.io/designer/
- Power Automate
- https://amdesigner.azurewebsites.net/

Play with SDK

Using SDK into your Applications

Rendering cards

- It's easy to render Adaptive Cards inside your application
- Install a renderer SDK:
    - JavaScript
    - .NET WPF
    - .NET HTML
    - Windows UWP
    - Android
    - iOS
- Create a renderer instance: configured with your app's style, rules, and action event handlers.
- Render a card to native UI: automatically styled to your app.

Actions

- By default, the actions will render as buttons on the card, but it's up to your app to make them behave as you expect.
- Each SDK has the equivalent of an OnAction event that you must handle.

HostConfig

HostConfig is a **cross-platform configuration object** that specifies how an Adaptive Card Renderer generates UI.

This **allows properties** which are platform agnostic **to be shared among renderers** on **different platforms** and devices.

- **AdaptiveCardConfig** - Toplevel options for AdaptiveCards
- **ActionsConfig** - Options for Actions
- **ContainerStylesConfig** - Controls styling for default and emphasis containers
- **FactSetConfig** - Controls the display of FactSets
- **FontSizesConfig** - Controls font size metrics for different text styles
- **FontWeightsConfig** - Controls font weight metrics
- **ForegroundColorsConfig** - Controls various font colors
- **ImageSetConfig** - Controls how ImageSets are displayed
- **ImageSizesConfig** - Controls Image sizes
- **MediaConfig** - Controls the display and behavior of Media elements
- **SeparatorConfig** - Controls how separators are displayed
- **ShowCardConfig** - Controls behavior and styling of Action.ShowCard
- **SpacingsConfig** - Controls how elements are to be laid out
- **TextBlockConfig** - Parameters controlling the display of text

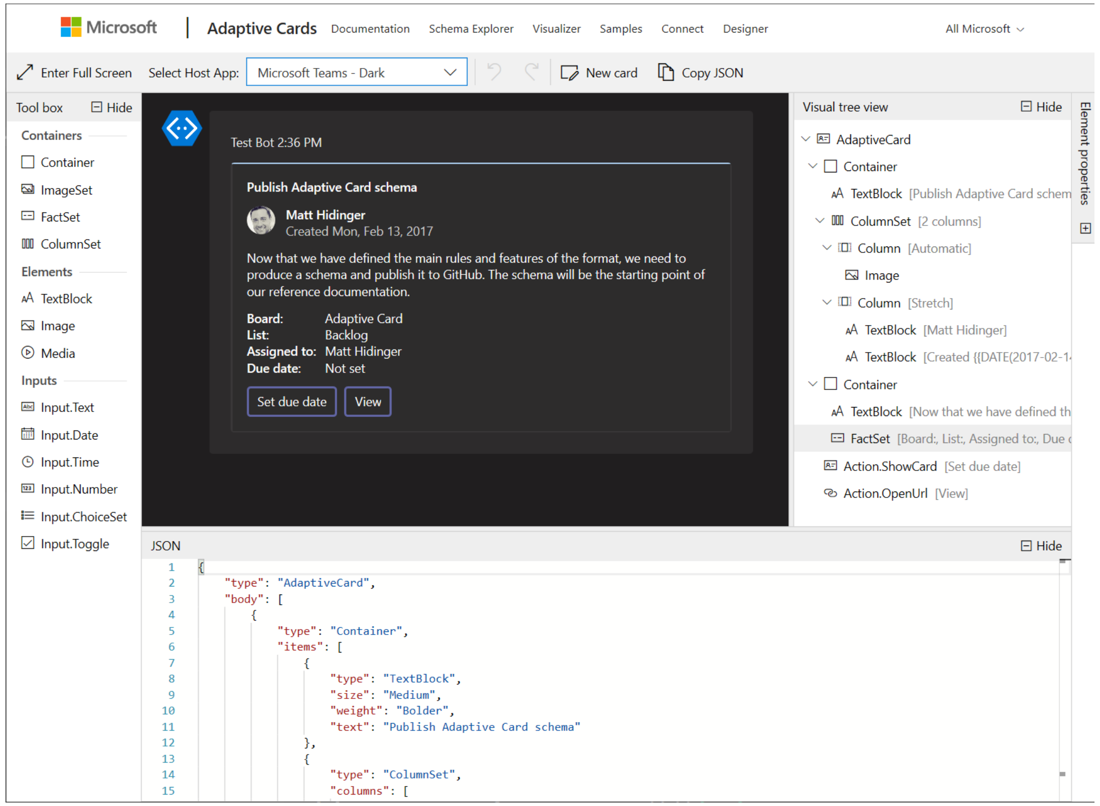
Extensibility

- Each SDK allows you to override the rendering of any element, or even add support for entirely new elements that you define.
- For example, you can change the Input.Date renderer to emit your own custom control while still retaining the rest of the output of the renderer.
- Or you can add support for a custom Rating element to you define.

Designer

- The Adaptive Card Designer provides **a rich, interactive design-time experience** for authoring adaptive cards.

- The designer **SDK is currently in preview** and may have breaking changes in the public API as we get feedback.



Test Bot 2:36 PM

Publish Adaptive Card schema

Matt Hidinger
Created Mon, Feb 13, 2017

Now that we have defined the main rules and features of the format, we need to produce a schema and publish it to GitHub. The schema will be the starting point of our reference documentation.

Board:	Adaptive Card
List:	Backlog
Assigned to:	Matt Hidinger
Due date:	Not set

Set due date · View

```
{
    "type": "AdaptiveCard",
    "body": [
        {
            "type": "Container",
            "items": [
                {
                    "type": "TextBlock",
                    "size": "Medium",
                    "weight": "Bolder",
                    "text": "Publish Adaptive Card schema"
                },
                {
                    "type": "ColumnSet",
                    "columns": [
```

SDK - DEMO

- Using JavaScript SDK into SPFx

Templating (Preview)

Templating (Preview)

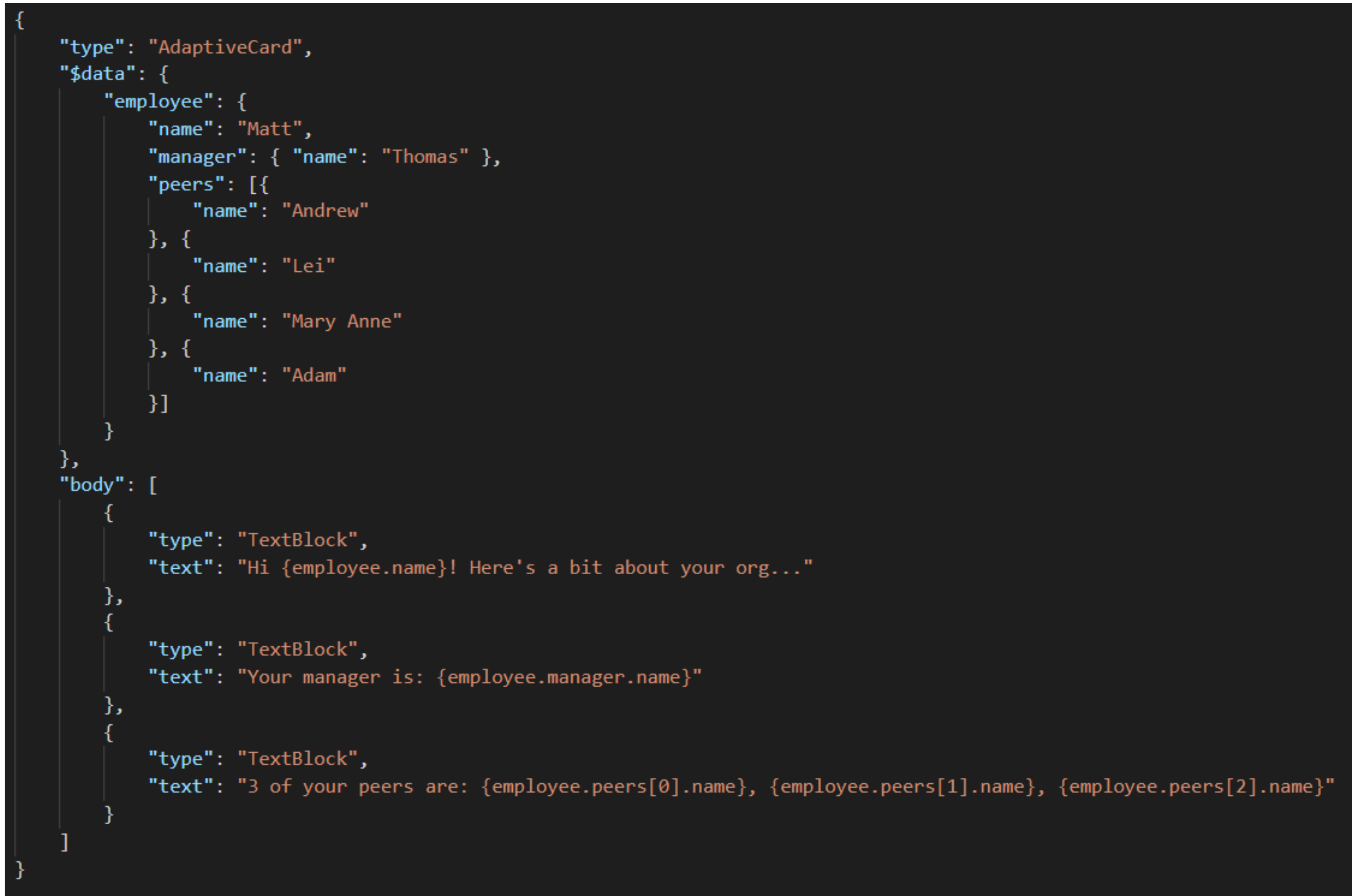
The **Template Language** is the syntax used for authoring a template. The Designer even lets you preview your templates at design time by including "sample data".

The **Templating SDK's** will exist on all supported Adaptive Card platforms. These SDKs allow you to populate a template with real data, on the back-end or directly on the client.

The **Template Service** is a proof-of-concept service that allows anyone to find, contribute to, and share a set of well-known templates.

Template Language (Preview)

Just and example of using the Template Language

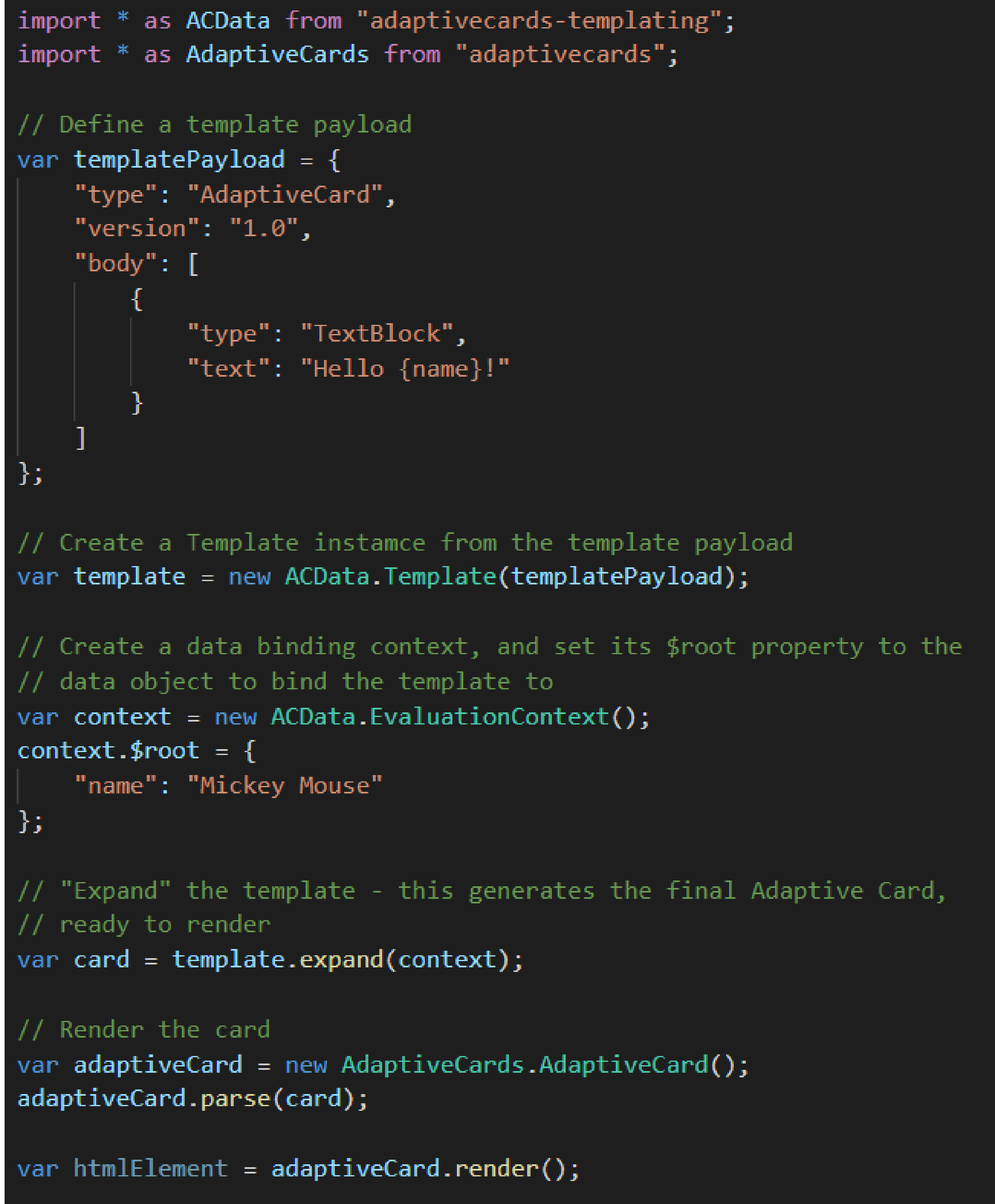


```
{
    "type": "AdaptiveCard",
    "$data": {
        "employee": {
            "name": "Matt",
            "manager": { "name": "Thomas" },
            "peers": [{
                "name": "Andrew"
            }, {
                "name": "Lei"
            }, {
                "name": "Mary Anne"
            }, {
                "name": "Adam"
            }]
        }
    },
    "body": [
        {
            "type": "TextBlock",
            "text": "Hi {employee.name}! Here's a bit about your org..."
        },
        {
            "type": "TextBlock",
            "text": "Your manager is: {employee.manager.name}"
        },
        {
            "type": "TextBlock",
            "text": "3 of your peers are: {employee.peers[0].name}, {employee.peers[1].name}, {employee.peers[2].name}"
        }
    ]
}
```

FABIOFRANZINI.COM

Template SDK (Preview)

Just and example of using the Template SDK



```
import * as ACData from "adaptivecards-templating";
import * as AdaptiveCards from "adaptivecards";

// Define a template payload
var templatePayload = {
    "type": "AdaptiveCard",
    "version": "1.0",
    "body": [
        {
            "type": "TextBlock",
            "text": "Hello {name}!"
        }
    ]
};

// Create a Template instamce from the template payload
var template = new ACData.Template(templatePayload);

// Create a data binding context, and set its $root property to the
// data object to bind the template to
var context = new ACData.EvaluationContext();
context.$root = {
    "name": "Mickey Mouse"
};

// "Expand" the template - this generates the final Adaptive Card,
// ready to render
var card = template.expand(context);

// Render the card
var adaptiveCard = new AdaptiveCards.AdaptiveCard();
adaptiveCard.parse(card);

var htmlElement = adaptiveCard.render();
```

Template Service (Preview)

The card template service is a simple REST endpoint that helps:

- **Find** a template **by analyzing** the structure of your **data**
- **Get a template** so you can bind it **directly on the client**, *without sending your data to the server or ever leaving the device*
- **Populate a template** on the **server**, when **client-side** data binding isn't appropriate or possible

Behind it all, is:

- **A shared, open-source template repository backed by GitHub**. *(The repo is currently private but will be made public as soon as we tie up some loose ends)*
- **All the templates are flat JSON files in the repo**, which makes editing, contributing, and sharing a natural part of a developer workflow.
- **The code for the service will be made available** so you can host wherever makes the most sense to you.

More infos here: https://docs.microsoft.com/en-us/adaptive-cards/templating/service

Resources

- Adaptive Cards: https://adaptivecards.io/
- Schema Explorer: https://adaptivecards.io/explorer/
- Samples: https://adaptivecards.io/samples/
- Designer: https://adaptivecards.io/designer/
- GitHub Repo: https://github.com/microsoft/AdaptiveCards
- Contoso Scuba Bot: https://github.com/matthidinger/ContosoScubaBot

Fabio Franzini

Microsoft Office Development MVP
Indipendent Senior Consultant & MCT Trainer

Mail: fabio@fabiofranzini.com
Web: http://fabiofranzini.com/
Twitter: @franzinifabio
LinkedIn: https://www.linkedin.com/in/fabiofranzini/

Microsoft® Most Valuable Professional

aOS

TOP 25 EUROPEAN SHAREPOINT INFLUENCERS 2018

Q & A