Low-Code Application Development

with Microsoft Power Apps and Azure

Andrew Oram

MANNING

For online information and ordering of this and other Manning books, please visit www.manning.com. The publisher offers discounts on this book when ordered in quantity. For more information, please contact

Special Sales Department
Manning Publications Co.
20 Baldwin Road
PO Box 761
Shelter Island, NY 11964
Email: orders@manning.com

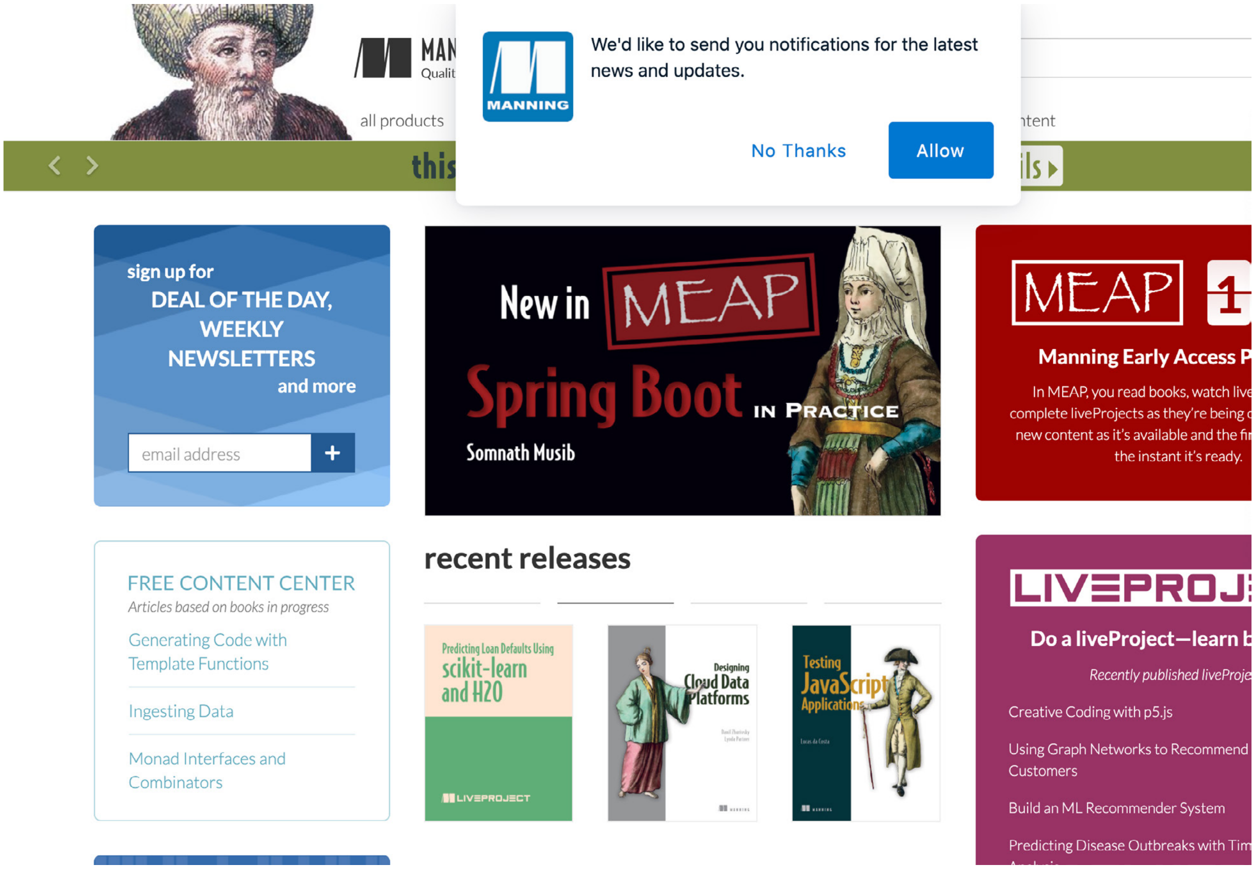
Manning Publications Co.
20 Baldwin Road
PO Box 761
Shelter Island, NY 11964

ISBN 9781617299803
Printed in the United States of America

Low-code application development with Microsoft Power Apps and Azure

The growing requirements placed on applications

These days, even a simple web page is actually a highly complex application. Take the page shown in figure 1, which plays animations, checks input for accuracy, loads elements dynamically, displays customized ads, and tracks visitors' behavior. No single web programmer writes all that code, of course. The page loads components from


We'd like to send you notifications for the latest news and updates.

sign up for DEAL OF THE DAY, WEEKLY NEWSLETTERS and more

In MEAP, you read books, watch live complete liveProjects as they're being c new content as it's available and the fi the instant it's ready.

FREE CONTENT CENTER

Articles based on books in progress

Generating Code with Template Functions

Ingesting Data

Monad Interfaces and Combinators

Recently published liveProje

Creative Coding with p5.js

Using Graph Networks to Recommend Customers

Build an ML Recommender System

Predicting Disease Outbreaks with Tim

Figure 1 Web page with many dynamic features (Manning Press web site)

multiple third parties, including libraries of prebuilt components, some of which can be quite complex.

As you get below the surface, the page is also connecting to a server and communicating with a database, each of which needs to be designed, written, and maintained. Finally, everything is connected with code written in JavaScript or another web programming language.

Low-code tools

Low-code tools start with the idea that common components, boilerplate code, and predictable application behaviors can be captured as prebuilt components and automated processes. The initial setup consists of choosing existing components and actions from menus and composing them to create the app. Then you can concentrate on adding the custom features that make your app unique.

Applications composed of standalone services that share data or have a common workflow are commonly called *service-oriented applications,* ranging from simple web applications like the ones mentioned earlier to large, enterprise-scale applications. Extending the idea of low-code tools to a cloud platform such as Microsoft Azure gives the user access to a vast array of prebuilt services that can be integrated into your application. This integration is the power behind Microsoft Power Apps.

Power Apps adds a visual interface (figure 2) to make it trivial to plug an impressive array of services into an application, including services for data analysis; artificial intelligence (AI); augmented, virtual, and mixed reality (AR/VR/MR); chatbots; popular relational and nonrelational databases; and enterprise resource planning (ERP) platforms. These tools are equally valuable to programmers and nonprogrammers such as business intelligence (BI) users.

Finally, most developers are on the hook for a lot more than creating an application. The modern development process includes testing and continuous integration, version control, issue tracking, creating input and output to devices, embedding the app in other environments (such as Power BI), and supporting multiple distribution channels (such as the web, mobile devices, and social media).

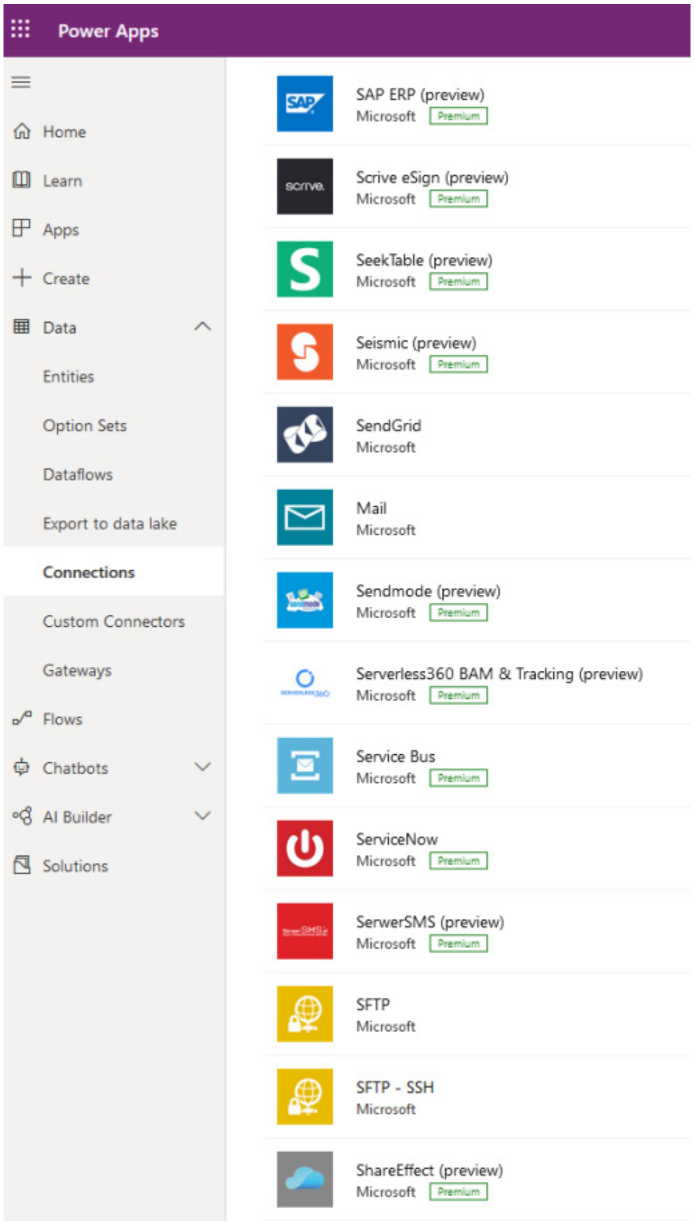
Because Power Apps is designed both for nonprogrammers who may be working alone—known as *citizen developers*—and for professional software engineers who are part of larger teams, it can support all these activities.

To summarize, today's programming environment involves

- Services and service-oriented designs
- Code reuse and third-party offerings
- Rapid development and continuous integration
- Integrated development environments (IDEs) and visual programming

Power Apps reflects those priorities, offering integration through RESTful interfaces with

- Databases, both to take in data and to provide functions to databases through triggers
- ERPs



SAP ERP (preview)
Microsoft Premium

Scrive eSign (preview)
Microsoft Premium

SeekTable (preview)
Microsoft Premium

Seismic (preview)
Microsoft Premium

SendGrid
Microsoft

Mail
Microsoft

Sendmode (preview)
Microsoft Premium

Serverless360 BAM & Tracking (preview)
Microsoft Premium

Service Bus
Microsoft Premium

ServiceNow
Microsoft Premium

SerwerSMS (preview)
Microsoft Premium

SFTP
Microsoft

SFTP - SSH
Microsoft

ShareEffect (preview)
Microsoft Premium

Figure 2 A few of the services available to Power Apps

- AI and related services, including
  - NLP
  - Chatbots
  - AR, VR, and MR
  - Scanning and interpreting receipts
- Devices in the Internet of Things (IoT)
- Development environments, notably GitHub and Azure DevOps
- Numerous channels to reach users, including
  - Websites
  - Mobile apps

- Power BI
- Microsoft Teams and Microsoft Office 360
- Power Automation and Microsoft Flows

This report describes how programmers can use Power Apps to incorporate richly featured tools into their applications and bring the applications to users in many ways. We'll look at some of the intriguing tools that tie Power Apps to its environment and end with a couple of examples that show how Power Apps can allow applications to grow over time.

The Power Apps interface

Power Apps is an example of Platform as a Service (PaaS). The Azure platform includes hundreds of prebuilt services that you can access via Power Apps. All these services come with consistent, well-documented APIs, so they're easy to incorporate into applications. Also, because they're developed and maintained by Microsoft, you don't have to worry about testing or versioning within the services; your app will always use the most current versions of tested components.

Some developers find this approach to be uncomfortable, because they're surrendering a measure of control over the internal working of applications. That said, the benefits of reduced maintenance and fewer bugs typically outweigh the need for direct control. When you need custom components, Power Apps embraces another new development model: serverless computing with Azure Functions.

Serverless computing is a broad term that also involves composing applications from components. The definition of *serverless* typically includes cloud-hosted functions such as Azure Functions: small, single-purpose applications that typically perform a well-defined task and then pass data on to another service. Each function can be developed within Azure or be developed on the programmers' systems locally with their tools of choice and then uploaded to Azure.

To develop within Azure, you can use App Service, which provides many simple visual interfaces to all the tools available. If you choose to code on your local system, you can use Visual Studio and then employ its Publish to Azure feature to install the function in your Azure account.

A function can run either on-premises or in the cloud, but integration with Azure's cloud service provides many benefits by managing the application's relationships with databases, other services, and other parts of the environment. Whether a function runs on-premises or in Azure, you expose it to other users by using an Azure graphical interface to create a custom connector. Examples of such connectors include an HTTP URL to provide a web interface, a connector to a Microsoft service such as Sharepoint or OneDrive, or a connector to an ERP or other third-party service supported by Power Apps.

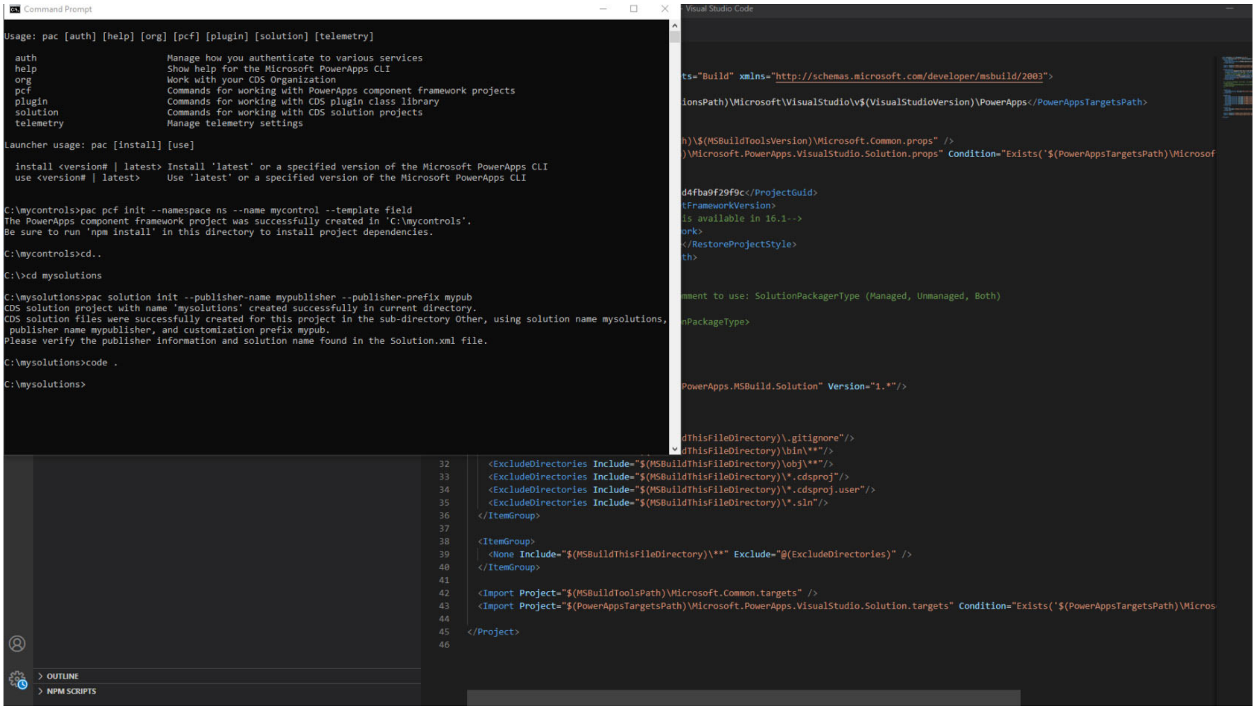
When your function is in Azure, you can load it up like a do-it-yourself ice cream sundae with all sorts of services, both those inside Azure and those to which Azure provides interfaces, such as databases, ERPs, and the GitLab version control platform.

Single sign-on gives users instant access to the services you've incorporated into your application.

As previously mentioned, Power Apps provides a low-code environment as a graphical interface. Typically, you drag a component on the screen where you want it. You can drag in a web canvas element that provides the service to the web page displayed by your app, for example. A low-code approach can speed up all kinds of applications, but particularly quick apps designed for a small group of users (such as internal company apps) in which performance is not a critical concern. Many opportunities exist for such applications, but they often aren't produced in the absence of a platform like Microsoft Power Platform, which makes them simple and low-cost.

Although the graphical interface is appropriate for a low-code environment, Power Apps also lets you build apps through a command-line interface (CLI). This CLI lets you do such things as install Power Apps components (plugins), store (push) components, and make backups of your application development environment (figure 3). The CLI enables scripting, so you can define robust, standard processes such as making sure that every application has the right authorization policies. You can also perform administrative tasks by using PowerShell (figure 4).

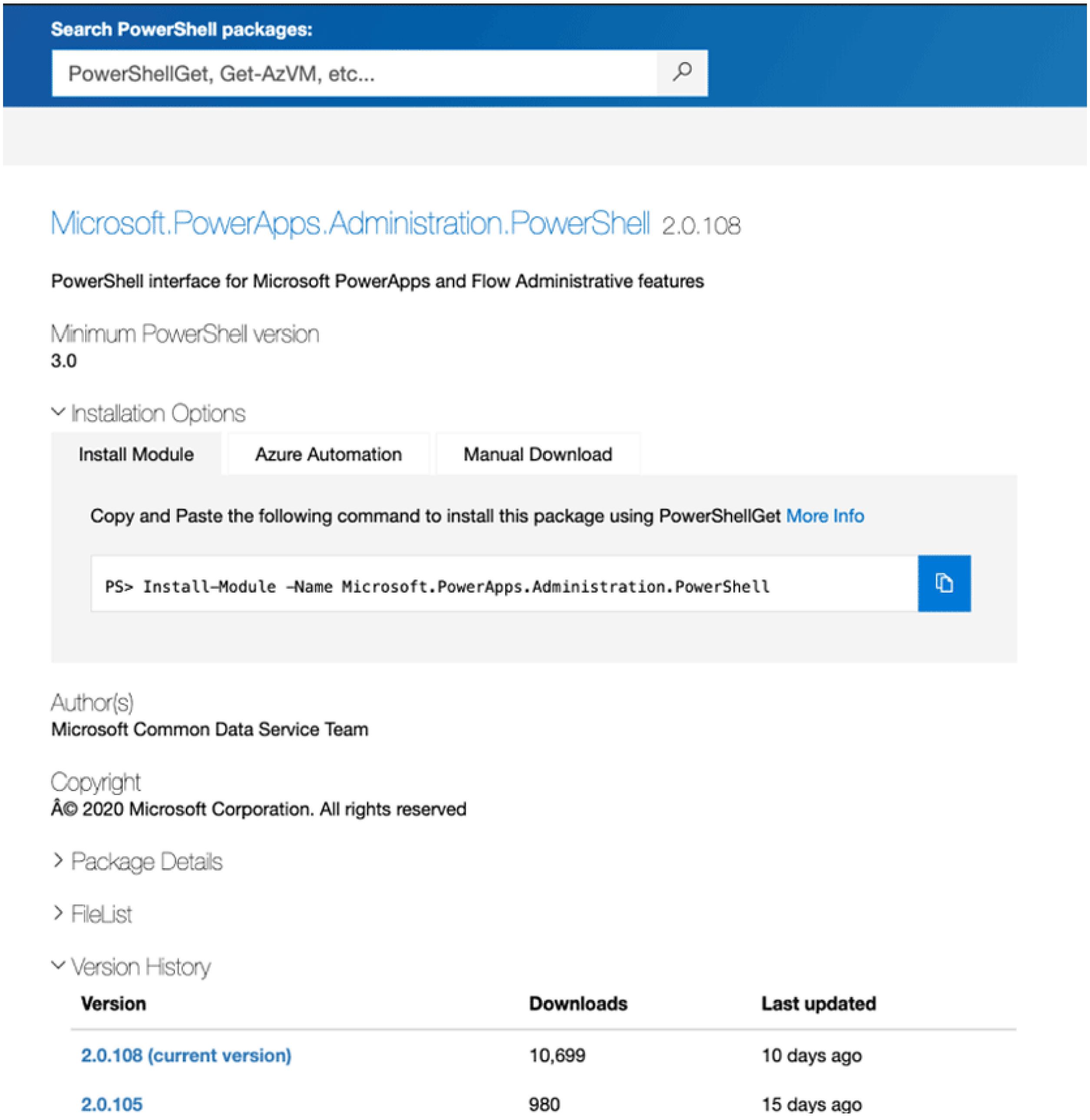
The remaining tasks that get your function working with the service indicate how to exchange data. RESTful APIs carry out this integration. The RESTful aspect of the interface allows you to write in the language of your choice and insert calls to all the services you want in a consistent manner that will soon become familiar. Data is passed between your function and the services by means of common RESTful techniques: function



```
Usage: pac [auth] [help] [org] [pcf] [plugin] [solution] [telemetry]

  auth                          Manage how you authenticate to various services
  help                          Show help for the Microsoft PowerApps CLI
  org                           Work with your CDS Organization
  pcf                           Commands for working with PowerApps component framework projects
  plugin                        Commands for working with CDS plugin class library
  solution                      Commands for working with CDS solution projects
  telemetry                     Manage telemetry settings

Launcher usage: pac [install] [use]

  install <version# | latest> Install 'latest' or a specified version of the Microsoft PowerApps CLI
  use <version# | latest>     Use 'latest' or a specified version of the Microsoft PowerApps CLI


C:\mycontrols>pac pcf init --namespace ns --name mycontrol --template field
The PowerApps component framework project was successfully created in 'C:\mycontrols'.
Be sure to run 'npm install' in this directory to install project dependencies.

C:\mycontrols>cd..

C:\>cd mysolutions

C:\mysolutions>pac solution init --publisher-name mypublisher --publisher-prefix mypub
CDS solution project with name 'mysolutions' created successfully in current directory.
CDS solution files were successfully created for this project in the sub-directory Other, using solution name mysolutions,
 publisher name mypublisher, and customization prefix mypub.
Please verify the publisher information and solution name found in the Solution.xml file.

C:\mysolutions>code .

C:\mysolutions>
```

```
ts="Build" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">

ionsPath)\Microsoft\VisualStudio\v$(VisualStudioVersion)\PowerApps</PowerAppsTargetsPath>


h)\$(MSBuildToolsVersion)\Microsoft.Common.props" />
)\Microsoft.PowerApps.VisualStudio.Solution.props" Condition="Exists('$(PowerAppsTargetsPath)\Microsof


d4fba9f29f9c</ProjectGuid>
tFrameworkVersion>
is available in 16.1-->
ork>
</RestoreProjectStyle>
th>


mment to use: SolutionPackagerType (Managed, Unmanaged, Both)

nPackageType>




PowerApps.MSBuild.Solution" Version="1.*"/>




dThisFileDirectory)\.gitignore"/>
dThisFileDirectory)\bin\**"/>
      <ExcludeDirectories Include="$(MSBuildThisFileDirectory)\obj\**"/>
      <ExcludeDirectories Include="$(MSBuildThisFileDirectory)\*.cdsproj"/>
      <ExcludeDirectories Include="$(MSBuildThisFileDirectory)\*.cdsproj.user"/>
      <ExcludeDirectories Include="$(MSBuildThisFileDirectory)\*.sln"/>
    </ItemGroup>

    <ItemGroup>
      <None Include="$(MSBuildThisFileDirectory)\**" Exclude="@(ExcludeDirectories)" />
    </ItemGroup>

    <Import Project="$(MSBuildToolsPath)\Microsoft.Common.targets" />
    <Import Project="$(PowerAppsTargetsPath)\Microsoft.PowerApps.VisualStudio.Solution.targets" Condition="Exists('$(PowerAppsTargetsPath)\Micros

  </Project>

```

Figure 3 Using the command-line interface for administration



Search PowerShell packages:

PowerShellGet, Get-AzVM, etc...

Microsoft.PowerApps.Administration.PowerShell 2.0.108

PowerShell interface for Microsoft PowerApps and Flow Administrative features

Minimum PowerShell version
3.0

Installation Options

Copy and Paste the following command to install this package using PowerShellGet More Info

```
PS> Install-Module -Name Microsoft.PowerApps.Administration.PowerShell
```

Author(s)
Microsoft Common Data Service Team

Copyright
Â© 2020 Microsoft Corporation. All rights reserved

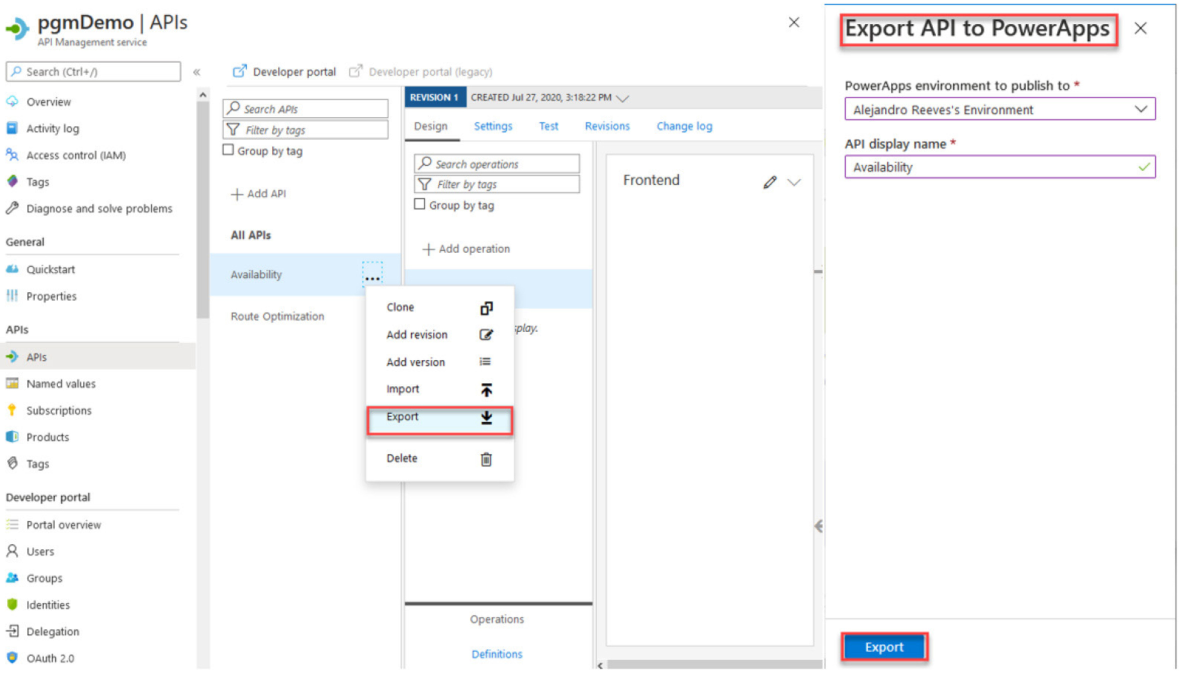
Version History

Version	Downloads	Last updated
2.0.108 (current version)	10,699	10 days ago
2.0.105	980	15 days ago

Figure 4 PowerShell administrative interface for Power Apps

parameters and human-readable JSON data structures. Power Apps also uses the standard JSON format for configuration, in a type of file commonly known as a *manifest.* The manifest provides the endpoints and related information that the function needs to communicate with the service.

Azure API Management (APIM) makes applications discoverable to other Azure users through a catalog. When you are ready to share your application, you can put it in this catalog by exporting it to the API Management system (figure 5). APIM can list both applications stored in Azure and those stored on-premises.



PowerApps environment to publish to * Alejandro Reeves's Environment

API display name * Availability

Figure 5 Exporting your API to Azure API Management

Popular services available through Power Apps

The list of services provided through Azure to Power Apps is huge and diverse. The world of services keeps growing, and Azure keeps up with new offerings all the time. Some of the services that you may find to be particularly valuable are listed in this section. Later sections of this report show a few examples of tying together various services, creating a potent combination that offers convenience to users.

Databases

Most applications query and update databases. Power Apps makes it easy to perform these functions on popular relational and nonrelational databases. But a Power Apps function can also be used within a database to provide functions that you code in the language of your choice, substituting for database-specific languages such as PL/SQL in Oracle or Transact SQL in Microsoft SQL Server.

When you drag and drop a database in your Azure account into your application, Power Apps gives you a form where you can enter the connection information, such as server and username. After establishing a connection, Power Apps allows you to choose a table and creates a simple visual interface to all the fields. You can choose which ones to query or update by clicking them.

You can also use functions within a database by means of triggers. Database administrators can program all kinds of sophisticated operations. Here are a few examples:

- Each time text is inserted into a character field, it could be passed to an AI program to determine what language it is and then translated into another language.

- Queries can be passed to a program that looks at who is making the query and run security checks that might not be available within the database.
- Sophisticated logging can be done on selected types of queries.

Active Directory

Directories such as LDAP and Active Directory (AD) are like telephone switchboards for businesses, allowing you to find employees or customers and control their access to different corporate functions. The Conditional Access Policies in AD allow an administrator to control access to an application by a user, a device, a location, and so on.

Power Apps lets you use the APIs for AD, which you can locate either on-premises or in Azure, to employ its search and security features. Suppose that you want to retrieve the user credentials for a user of Microsoft Office 365. You could connect Power Apps to the AD containing that user's record and retrieve information such as their email address, username, and anything else that the AD stores. By performing these tasks through Power Apps, you avoid a lot of tedious routing, authentication, and the like.

Enterprise resource providers

Another crucial service that most businesses use is an ERP provider. The APIs for numerous popular ERPs are available to Power Apps through Azure. (A few examples appear in figure 2 earlier in this report.) Thus, you can quickly assemble an application that does things such as the following:

- Consult the ERP for office locations and find the one closest to a remote employee through geolocation
- Run analytics on sales to determine the rise or fall in the popularity of products and compare this behavior with the behavior of factories, tracked by the ERP

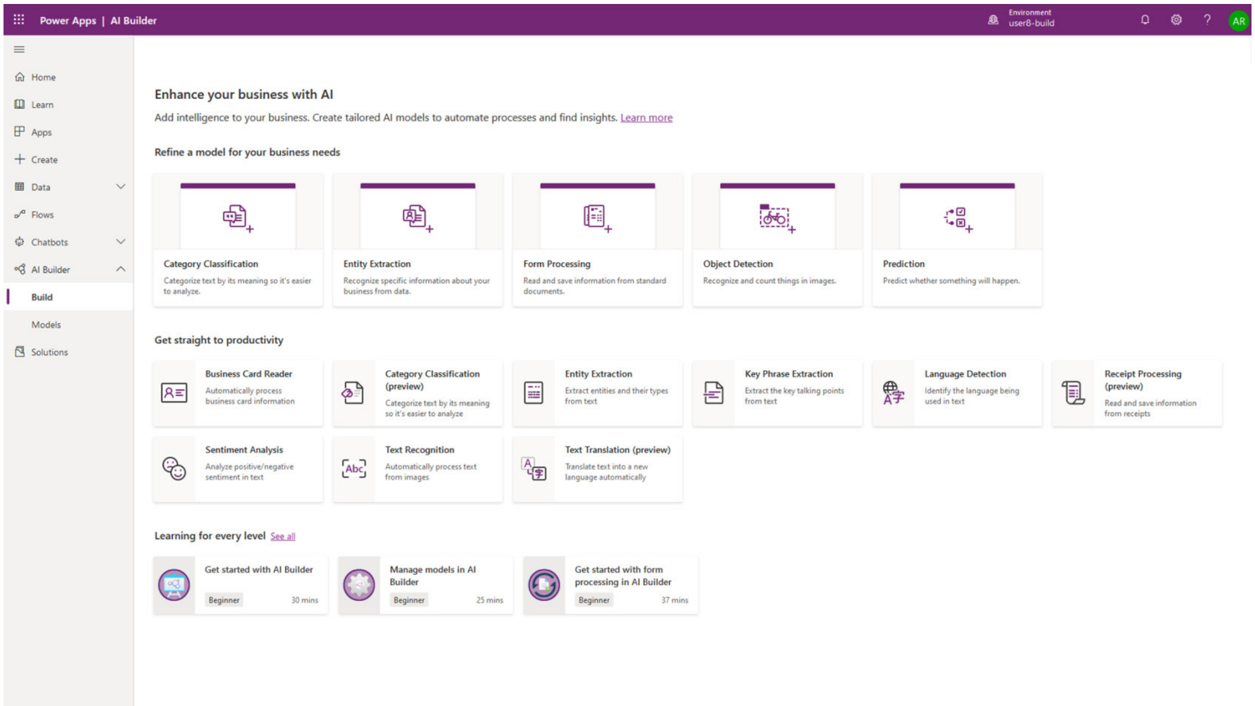
Artificial intelligence

Over the past decade, machine learning has solved perplexing tasks such as speech recognition and face identification. Azure has integrated many business uses for AI into turnkey components (figure 6). Although these powerful functions must be vetted carefully for accuracy, lack of bias, and respect for privacy, they are creating a better world that gives people new capabilities. Voice interfaces, for example, are a boon to everyone from a busy parent holding a baby to a vision-impaired person to anyone who is too busy to figure out a graphical interface.

Feature-rich AI services can be loaded from Azure into Power Apps through a visual interface (figure 7).

AI services available to Power Apps include

- *NLP—natural language processing* is a broad term for techniques and technologies that allow a computer to interpret, respond to, and generate human language. NLP services can be used with written text as well as spoken words. This complex domain involves everything from smarter search to automatic summarization of



Enhance your business with AI

Add intelligence to your business. Create tailored AI models to automate processes and find insights. Learn more

Refine a model for your business needs

Category Classification
Categorize text by its meaning so it's easier to analyze.

Entity Extraction
Recognize specific information about your business from data.

Form Processing
Read and save information from standard documents.

Object Detection
Recognize and count things in images.

Prediction
Predict whether something will happen.

Get straight to productivity

Business Card Reader
Automatically process business card information

Category Classification (preview)
Categorize text by its meaning so it's easier to analyze

Entity Extraction
Extract entities and their types from text

Key Phrase Extraction
Extract the key talking points from text

Language Detection
Identify the language being used in text

Receipt Processing (preview)
Read and save information from receipts

Sentiment Analysis
Analyze positive/negative sentiment in text

Text Recognition
Automatically process text from images

Text Translation (preview)
Translate text into a new language automatically

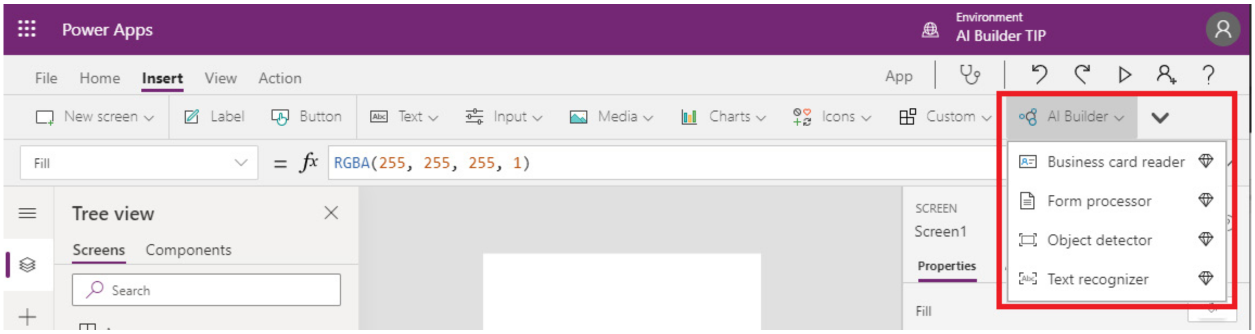
Learning for every level See all

Get started with AI Builder
Beginner 30 mins

Manage models in AI Builder
Beginner 25 mins

Get started with form processing in AI Builder
Beginner 37 mins

Figure 6 Numerous AI tools of use to businesses are available to Power Apps on a turnkey basis



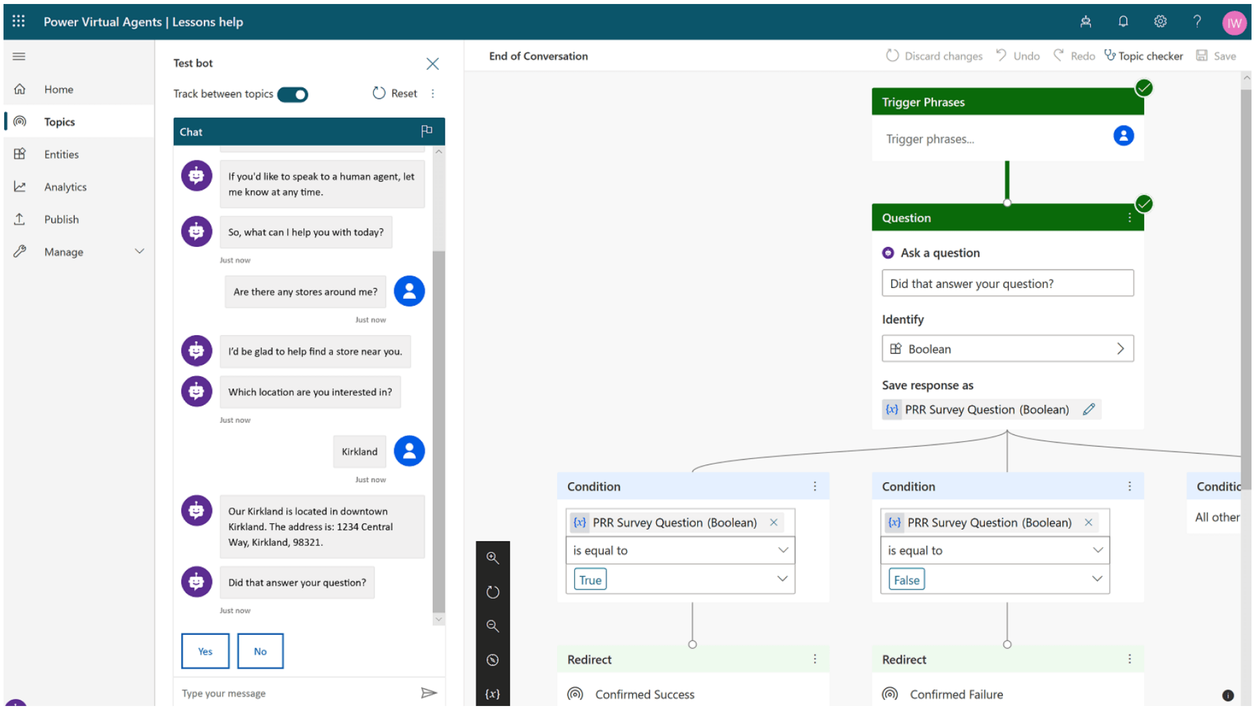
```
= fx RGBA(255, 255, 255, 1)
```

Figure 7 A pull-down menu allows you to choose AI components and load them into your application.

long documents. Used well, NLP makes drop-down menus look old and tired. A user can type or speak their request and be taken directly to the web page or function they need.

- *Chatbots*—Chatbots build on NLP to guide a user through questions and answers to what the user needs. Chatbots are being widely employed to save the cost of employing support staff to talk to a client and find out what the client is seeking. Power Apps makes it easy to incorporate a chatbot, along with an API for providing the hierarchical flow of questions and answers that drive the chatbot.

  Power Virtual Agents, which are part of the Power Platform along with Power Apps, create sophisticated chatbots quickly. A sample chatbot is shown in figure 8, along with the visual interface in which the designer provides chatbot



If you'd like to speak to a human agent, let me know at any time.

So, what can I help you with today?

Just now

Are there any stores around me?

Just now

I'd be glad to help find a store near you.

Which location are you interested in?

Just now

Kirkland

Just now

Our Kirkland is located in downtown Kirkland. The address is: 1234 Central Way, Kirkland, 98321.

Did that answer your question?

Just now

Yes No

Ask a question: Did that answer your question?

Identify: Boolean

Save response as: PRR Survey Question (Boolean)

Condition: PRR Survey Question (Boolean) is equal to True → Redirect: Confirmed Success

Condition: PRR Survey Question (Boolean) is equal to False → Redirect: Confirmed Failure

Figure 8 Chatbot designed through the Power Apps graphical interface

conversations. Coca-Cola used Power Virtual Agents to create a chatbot as an appealing interface to a touchscreen beverage dispenser, for example. The Power Apps platform made integration with the SAP management service easy.

- *AR, VR, and MR*—*AR* provides not only a powerful interface for interacting with a person's immediate environment, but also new possibilities for training and exploration. VR is known mainly for games but has other valuable applications, such as teaching surgical techniques, allowing chemists to explore large molecules, and stimulating the minds of homebound elderly people. MR is a recent feature of Power Apps that can be combined with other functions to provide information about the user's environment (figure 9).
- *Scanning*—Power Apps has access to functions for scanning and interpreting everyday documents such as receipts and business cards. Scanning can be integrated with an application to provide such benefits as pulling up the LinkedIn profile of a person the client just met. Figure 10 shows a scanning tool that extracts requested fields from a scanned document.

Devices in the Internet of Things

Whether they're smart lights that save electricity by turning themselves off, cameras used for security, sensors that report air quality from hour to hour, or monitors used in factory automation, embedded devices provide both valuable data and the ability to control our environment. The same RESTful model used for other services can be used to connect to a device in the field, read its data, and send it instructions.

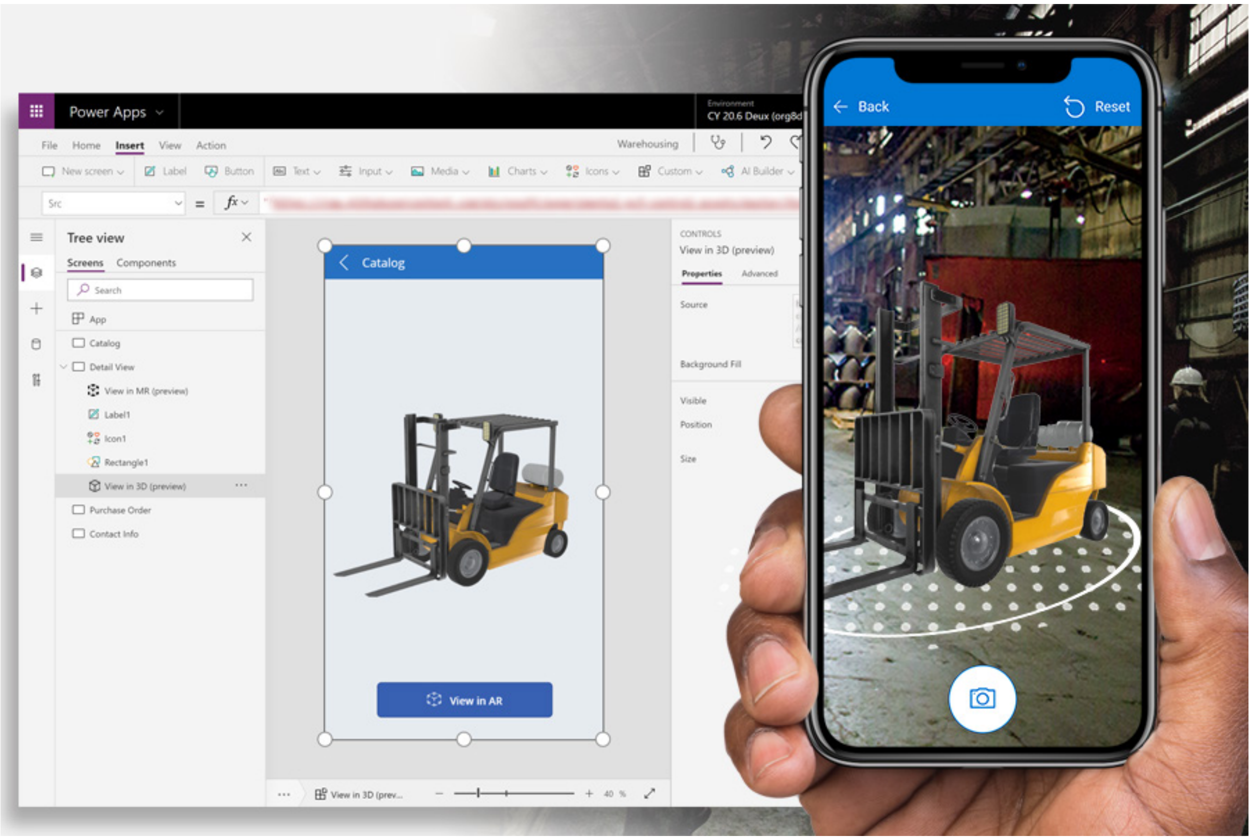
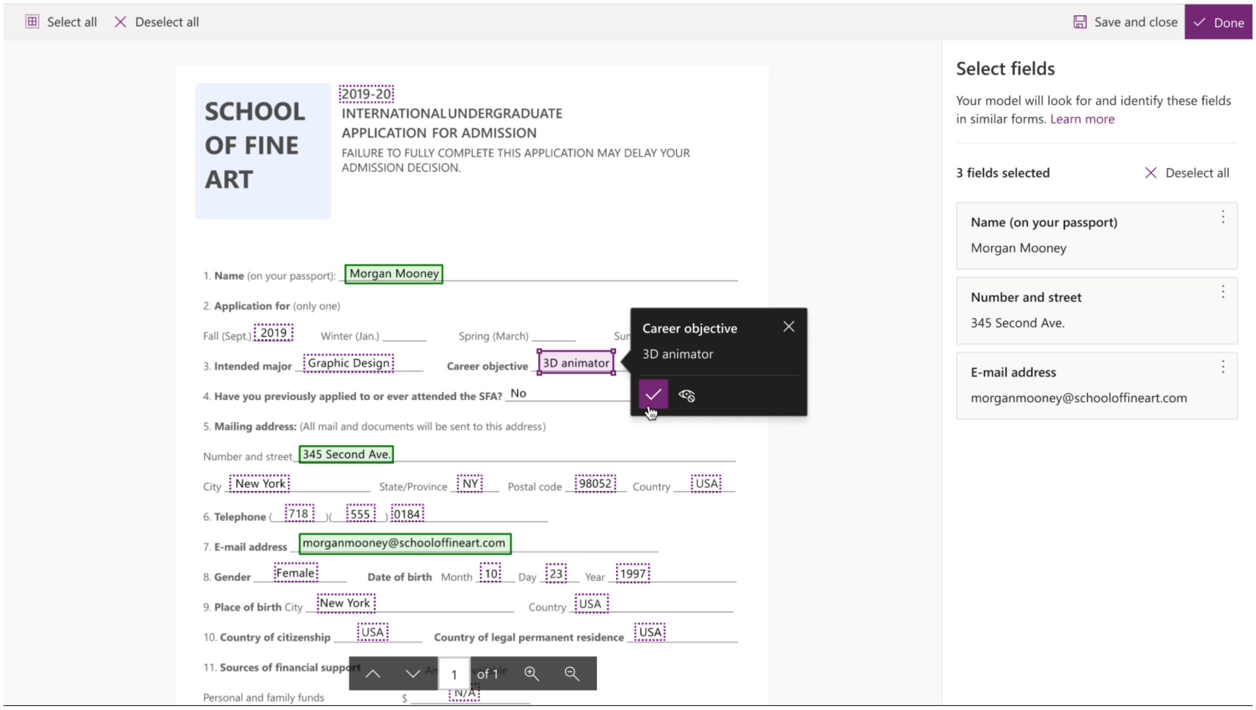


Figure 9 How Mixed Reality can provide enhanced information to the user about their environment



SCHOOL OF FINE ART

2019-20
INTERNATIONALUNDERGRADUATE
APPLICATION FOR ADMISSION
FAILURE TO FULLY COMPLETE THIS APPLICATION MAY DELAY YOUR ADMISSION DECISION.

1. **Name** (on your passport): Morgan Mooney

2. **Application for** (only one)

Fall (Sept.) 2019 Winter (Jan.) ______ Spring (March) ______ Sur

3. **Intended major** Graphic Design **Career objective** 3D animator

4. **Have you previously applied to or ever attended the SFA?** No

5. **Mailing address:** (All mail and documents will be sent to this address)

Number and street 345 Second Ave.

City New York State/Province NY Postal code 98052 Country USA

6. **Telephone** (718)(555) 0184

7. **E-mail address** morganmooney@schooloffineart.com

8. **Gender** Female **Date of birth** Month 10 Day 23 Year 1997

9. **Place of birth** City New York Country USA

10. **Country of citizenship** USA **Country of legal permanent residence** USA

11. **Sources of financial support**

Personal and family funds $ N/A

Career objective
3D animator

Select fields

Your model will look for and identify these fields in similar forms. Learn more

3 fields selected Deselect all

Name (on your passport)
Morgan Mooney

Number and street
345 Second Ave.

E-mail address
morganmooney@schooloffineart.com

Figure 10 You can choose which fields to extract from a scanned form

How your application becomes a world citizen through integration with Azure

One of the most significant trends in programming in recent decades applies increasing rigor to the environment in which a program is developed, tested, deployed, and monitored. Although activities such as version control and unit testing started primarily as ways to create higher-quality software, the result has been much more far-reaching.

Today, mastering the development life cycle is understood to be every bit as important as mastering the skill of programming. Continuous Integration, Continuous Deployment, and Continuous Delivery (often abbreviated as CI/CD) ensure that software is rigorously tested at every stage of development and then deployed in a consistent, safe manner. Many shops now deploy software hundreds or thousands of times daily because each change can go from check-in to rollout with the click of a button.

Increasingly, the development process is being expanded to include the active participation of business decision-makers and other project stakeholders. Power Apps facilitates this process by democratizing development. Business users can contribute both to the specification of the software and to its development.

Finally, the PaaS model blurs the traditional lines between development and operations. Developers can use Power Apps to create applications that are deployed immediately without additional infrastructure management. Azure takes care of low-level details such as security, networking, and load balancing so that you can concentrate on how you want your app to function.

The integration of Power Apps into Azure gives you access to the tools Azure provides in this area, such as

- Scalability and high availability
- Version control and CI/CD
- Rapid development (DevOps)
- Rich telemetry and alerts
- Security and regulatory conformance

We'll discuss these tools in the following sections.

Scalability and high availability

As a production-scale cloud service, Azure takes care of the automatic scaling and fault tolerance that operators expect from the cloud. You can specify the control parameters in your function, so you can set a limit to scaling, how long to wait between health checks, and other parameters that are common in scalability and high availability.

Version control and CI/CD

For these essential features of modern development, Power Apps takes advantage of Azure's access to GitHub, the world's most popular development platform and now part of Microsoft, with an estimated 56 million developers. You can configure Power Apps to check your function automatically into your GitHub repository whenever you save the source code and to run tests at regular intervals. Write the actions you want

GitHub to take in a workflow definition file. There are a couple dozen events, such as pushing a new version of code into the repository or building a release, where you can automatically trigger workflows.

Integration goes even further than the development and test stages. Through GitHub Actions or Azure DevOps, you can automate workflows and specify events that trigger a workflow. GitHub and Azure DevOps perform many of the same tasks, but developers are encouraged to use GitHub Actions because they offer richer possibilities.

Modern development processes recognize programming as a team effort—one that is often carried out by people collaborating around the world. Therefore, integration with GitHub is essential to the Power Apps vision of rapidly incorporating many tools and features into applications.

DevOps

As software became more complex, many companies assumed that their products required a slow, heavyweight development process and clear separation of responsibilities within organizations. Large projects suffered from teams that operated in silos with poor communication and competing priorities. The result was slower development and stifled innovation.

The DevOps movement automates testing and deployment, providing a more consistent and secure development environment and potentially lowering the technical bar for participation in the process. Many organizations see that the rapid iteration and rollout of promised features make DevOps a good approach for at least some of their software. Further, DevOps encourages cross-functional communication and less-strict definition of operating boundaries. Developers may take on tasks that were once reserved for infrastructure engineers, for example, and business managers may participate in software-team decisions.

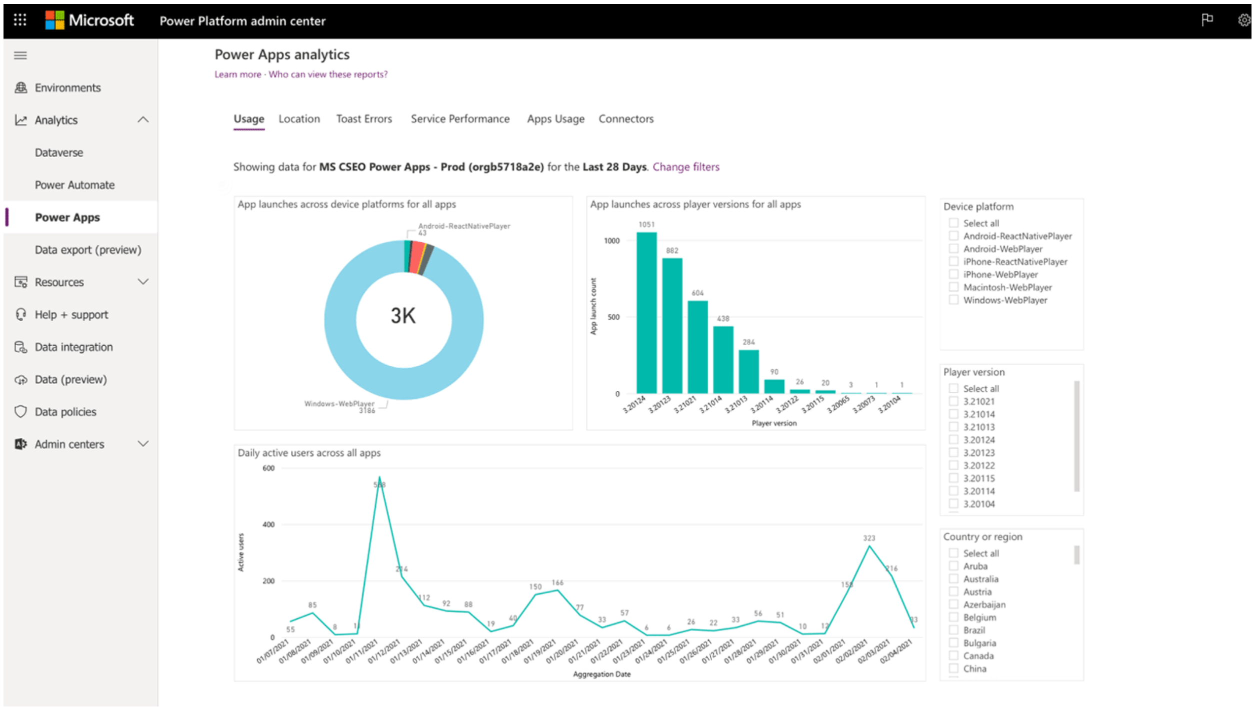
As an interesting consequence of its success (the term having been invented a little more than a decade ago), DevOps has become so popular that the word is starting to lose its precise meaning. Whatever you call it, however, the Power Platform, of which Power Apps is a part, offers teams the tools they need to automate and manage their application life cycles.

Rich telemetry and alerts

Azure collects a wide variety of statistics about the deployment and performance of processes running in the cloud; then it exposes these statistics and lets you view trends through dashboards that you create through Power Apps (figure 11). Bold operators and programmers can even link these statistics to AI tools to achieve high-level observability.

Security and regulatory conformance

All the tools described so far in this report are built with corporate security in mind. In addition, the Power Platform recognizes data protection regimes such as the European Union’s GDPR, so you can ensure compliance as data flows from one user to another. The Power Platform controls access according to many criteria that you can call upon to protect your application, including the following:



Showing data for **MS CSEO Power Apps - Prod (orgb5718a2e)** for the **Last 28 Days**. Change filters

Device platform
- □ Select all
- □ Android-ReactNativePlayer
- □ Android-WebPlayer
- □ iPhone-ReactNativePlayer
- □ iPhone-WebPlayer
- □ Macintosh-WebPlayer
- □ Windows-WebPlayer

Player version
- □ Select all
- □ 3.21021
- □ 3.21014
- □ 3.21013
- □ 3.20124
- □ 3.20123
- □ 3.20122
- □ 3.20115
- □ 3.20114
- □ 3.20104

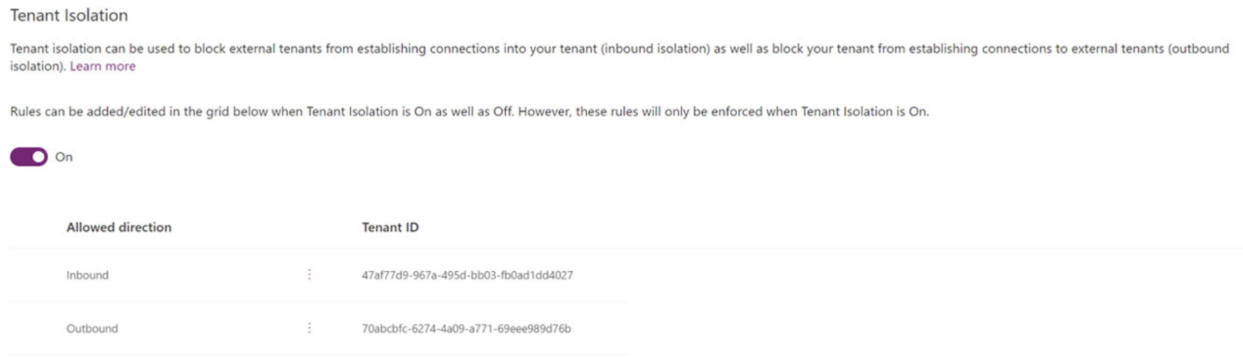
Country or region
- □ Select all
- □ Aruba
- □ Australia
- □ Austria
- □ Azerbaijan
- □ Belgium
- □ Brazil
- □ Bulgaria
- □ Canada
- □ China

Figure 11 Sample dashboard produced by Azure for a running application

- You can control access at the level of the application, the flow, the user, and so on.
- In a cloud environment, where you serve many companies or divisions of your own company, you can configure firewall policies for each tenant, such as to control inbound or outbound connections (figure 12).
- You can cut off nonbusiness connectors—the kinds of popular sites that people use for fun or for personal projects.

Widespread deployment

Originally, programs were distributed as standalone tools; then the Internet made them easier to use by allowing companies to deliver them through a browser



Tenant Isolation

Tenant isolation can be used to block external tenants from establishing connections into your tenant (inbound isolation) as well as block your tenant from establishing connections to external tenants (outbound isolation). Learn more

Rules can be added/edited in the grid below when Tenant Isolation is On as well as Off. However, these rules will only be enforced when Tenant Isolation is On.

● On

Allowed direction	Tenant ID
Inbound	47af77d9-967a-495d-bb03-fb0ad1dd4027
Outbound	70abcbfc-6274-4a09-a771-69eee989d76b

Figure 12 Configuring isolation rules for different tenants sharing a private cloud

(Software as a Service). But these types of programs are no longer sufficient to reach all the users you need to reach. In fact, the promise of Power Apps to allow massive integration extends to the environments in which you can embed your functions.

In this section, we'll examine the following channels to reach users:

- Websites
- Mobile apps
- Power BI
- Power Automation and Microsoft Flows
- Microsoft Teams and Microsoft Office 360

Websites

Suppose that you're working for a college and have developed an application that displays a map, customizing it based on what you know about the user. For a chemistry major, for example, you highlight on the map all the buildings that are relevant to that discipline. Now you want people to use this app.

In addition to building your own web application, you may want many other websites run by the university—or even by the surrounding city and other institutions—to show your maps. You can do this by creating an HTML canvas element, an amazingly powerful feature that is thoroughly standard and robust. Azure exposes your function as a canvas that any website with the proper permissions can embed. Websites that display Power Apps functions are responsive by design (figure 13).

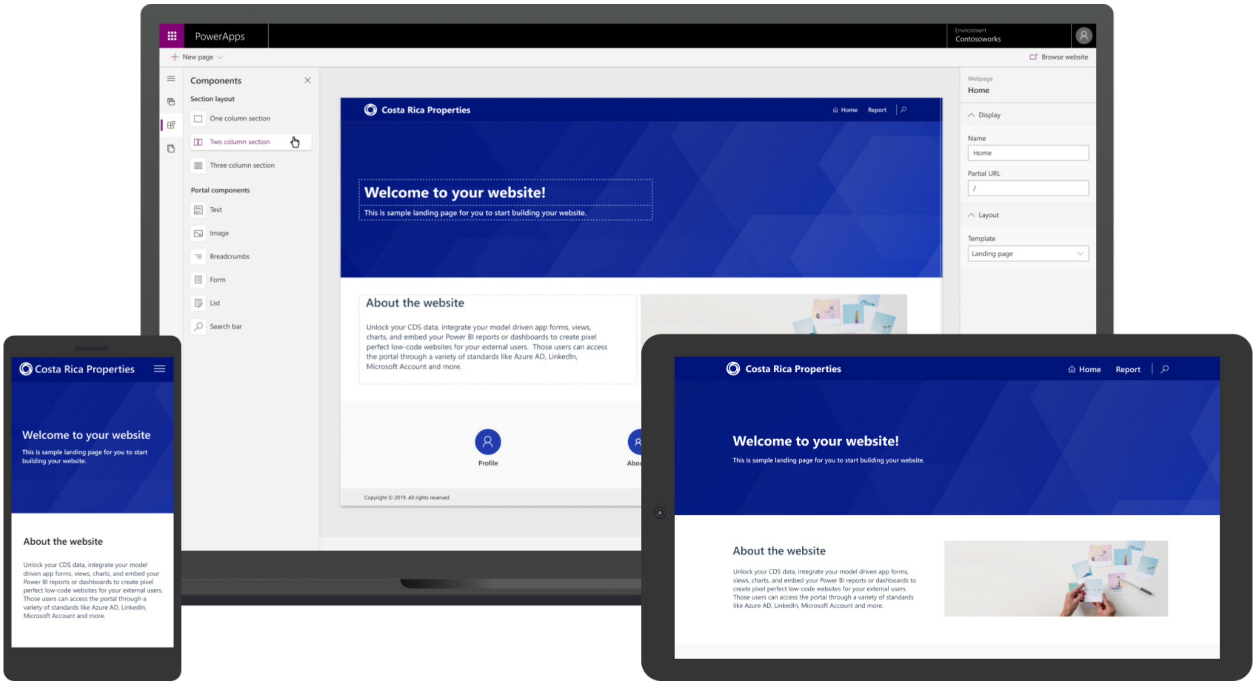
One simple, low-code way to create a website uses Power Apps portals. These portals are based on Power Fx, a descriptive language that lets you create a website and integrate it with a database by specifying a few parameters. For example, Ernst & Young used a Power Apps portal to create an application that helped banks distribute funds from the U.S. Paycheck Protection Program quickly in response to the hardships caused by the COVID-19 pandemic.

Mobile apps

Many applications are mobile-first, prioritizing the experience that a user has on a cellphone or tablet. Power Apps provides tools that help you rapidly build mobile-friendly versions of your software. You can quickly generate iOS and Android apps from your functions. Windows devices use the canvas element described in the Websites section.

Power BI

Nonprogramming knowledge workers are accustomed to using BI tools to investigate organizational data through charts and reports. You can provide functions to Microsoft's Power BI to greatly expand the analytics that it offers users. After a Power BI user selects your function from a menu, Power BI displays a table with the data fields output by the function; then the user can select what they want to see. The result: customizable integration of your function into Power BI without the need for the user to do any programming. Schlumberger used Power Apps to create a wellness app for its



Welcome to your website!

This is sample landing page for you to start building your website.

About the website

Unlock your CDS data, integrate your model driven app forms, views, charts, and embed your Power BI reports or dashboards to create pixel perfect low-code websites for your external users. Those users can access the portal through a variety of standards like Azure AD, LinkedIn, Microsoft Account and more.

Copyright © 2019. All rights reserved

Welcome to your website

This is sample landing page for you to start building your website.

About the website

Unlock your CDS data, integrate your model driven app forms, views, charts, and embed your Power BI reports or dashboards to create pixel perfect low-code websites for your external users. Those users can access the portal through a variety of standards like Azure AD, LinkedIn, Microsoft Account and more.

Welcome to your website!

This is sample landing page for you to start building your website.

About the website

Unlock your CDS data, integrate your model driven app forms, views, charts, and embed your Power BI reports or dashboards to create pixel perfect low-code websites for your external users. Those users can access the portal through a variety of standards like Azure AD, LinkedIn, Microsoft Account and more.

Figure 13 Responsive web sites are a built-in features of Power Apps

employees and then integrated the app with Power BI so that it could track the use of the app and determine whether it was having a effect.

Power Automation and Microsoft Flows

Workers in Microsoft environments such as 360 and Power BI have been using these no-code tools to follow their workflows and automate them; now they can call Power Apps functions within these workflows. At the point where a decision has to be made about the recipient of an email or chat message, for example, you can write a function to analyze the task and make the right choice without involving the user.

Microsoft Teams and Microsoft Office 360

Millions of users spend a majority of their working day in Office 360 and Teams. One of the most powerful features of Power Apps is seamless integration with these important platforms. Suppose that a business analyst using Microsoft Excel in Office 360 regularly needs to perform a complex set of calculations to generate a report. In Power Apps, you could create a custom function to build the report from the Excel data and embed it directly in the worksheet.

As we've previously discussed, Power Apps is designed for both professional developers and citizen developers such as the business analyst described in the preceding paragraph. With a little practice, a motivated analyst could create such custom functions without the assistance of a development team.

Sample scenarios

Previous sections of this report introduced various applications and services, suggesting how each one could enhance an application. Now we'll walk through some projects that start small and add more functionality to meet the growing needs of the organizations. We'll also see how Power Apps can promote rapid development.

A workflow management tool

Company A wants an application that helps people share tasks and track their progress. This application should integrate with the company's ERP. A developer uses Power Apps to create an application that offers automatic access to the ERP through the single sign-on feature provided in Azure. People who need to explore another team to find the right person to handle a task can use AD, which also can be integrated easily with Power Apps.

Although employees say that the application is useful, they are not consulting it as often as they could be because they are working in Teams. A simple configuration change exposes the application as a Teams personal app. A calendar can be provided through a canvas app.

Most of the company's employees have been working from home since COVID-19 hit. But sometimes, they need to come to a company facility to do things like pick up equipment. To help employees find the facility closest to their homes, the developer adds geolocation, consulting a database that contains the address of every facility and every employee's home.

The app collects information on what devices are used to gain access. Analytics show that 95% of users are on mobile devices, so the application is exposed as a mobile app. Eventually, to make access even easier for mobile users, a chatbot is added that can use built-in speech recognition to answer questions.

A factory management application

A factory needs to match resources with orders, comparing information about resources, incoming materials, output goals, and so on. All this information would be available in a relational database, which is straightforward to code, but the developers create the application in Power Apps to make future growth easy. Right away, they find many AI tools that are useful for making predictions about what they need. Over time, the factory's processes get more efficient as AI tools find patterns in data on ordering and production. Because these AI tools work best with a nonrelational database that's good for big data, one of those databases is easily linked in.

The factory then takes a major step forward in data collection by installing sensors on its machine. These sensors capture data that's useful for predicting system failures or the introduction of defects into products. The IoT interfaces in Power Apps make integrating these devices a quick task.

Finally, business analysts are asking for access to the applications' insights through Power BI, which they use every day. That integration is also simple, and analysts can choose exactly what they want to see.

A comprehensive vision for integrated applications

There's too much wonderful software out there to tolerate barriers to its use. During the 1990s, people got used to consuming applications through their browsers. In the following decade, it was the turn of mobile devices to offer applications, followed by third-party sites such as social media. The current trend is toward offering functions developed by many organizations within any interface the user prefers.

Thus, programmers need to tie multiple databases into their applications quickly, along with more highly processed data from ERPs and other services, and even with sensors in the field. Programmers need to integrate complex AI analytics, and they need to reach users wherever they are, on whatever devices and sites they use.

Power Apps responds to these multiple demands for integration by using standard, familiar mechanisms such as RESTful API calls. Programmers and operators need to expand their skills to encompass this kind of integration to create applications that keep up with user expectations. The rewards for everyone are impressive: faster release cycles, more powerful tools, greater convenience, and greater democratization of data and computing.

Conclusion

Programmers are increasingly willing to get help with programming. Just as they rarely write assembly code, having long ago adopted advanced modern languages, programmers are turning from plain text editors and command-line tools to IDEs because the advantages of such tools are too impressive to ignore.

Some programmers, out of pride, want to do all their own coding. Although they recognize the necessity of bringing in libraries and modules from third parties, they may feel that it is a sign of immaturity to use low-code tools and pick functions from a GUI. But luckily, the sign of a professional programmer is no longer coding virtuosity. You don't show your expertise by proving that you can write functions to process double-linked lists from scratch. Instead, the modern programmer is valued for higher-level skills, such as

- Talking to end users and understanding their business requirements
- Determining the effects of architecture choices on performance
- Designing for scale and for easy upgrades
- Communicating well with teams and in comments inserted during code check-ins
- Representing their organization's needs in standards bodies and open source communities

An environment such as Power Apps, in the context of the Azure platform, takes on the low-level tasks and frees you to exercise these high-level skills. You can still do all the programming you used to do with Power Apps, but faster and with less time spent on fixing bugs. It's even better if you exploit the boost that Power Apps gives your projects to boost your own career, taking on new tasks that you never had the cycles for before.

Microsoft

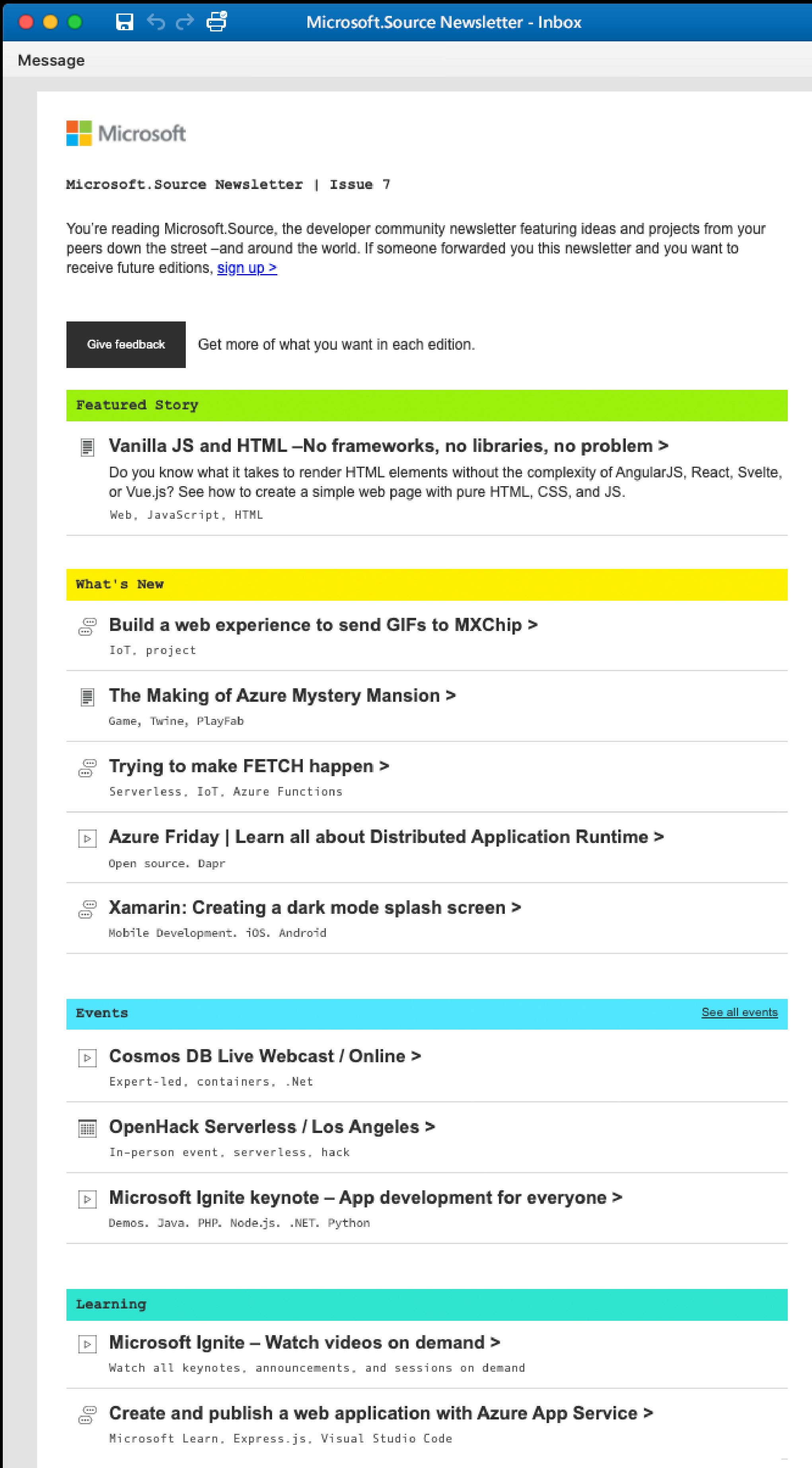
By developers, for developers

Microsoft.Source newsletter

Get technical articles, sample code, and information on upcoming events in Microsoft.Source, the curated monthly developer community newsletter.

- Keep up on the latest technologies
- Connect with your peers at community events
- Learn with hands-on resources

→ Sign up



Microsoft

Microsoft.Source Newsletter | Issue 7

You're reading Microsoft.Source, the developer community newsletter featuring ideas and projects from your peers down the street –and around the world. If someone forwarded you this newsletter and you want to receive future editions, sign up >

Get more of what you want in each edition.

Featured Story

Vanilla JS and HTML –No frameworks, no libraries, no problem >
Do you know what it takes to render HTML elements without the complexity of AngularJS, React, Svelte, or Vue.js? See how to create a simple web page with pure HTML, CSS, and JS.
Web, JavaScript, HTML

What's New

Build a web experience to send GIFs to MXChip >
IoT, project

The Making of Azure Mystery Mansion >
Game, Twine, PlayFab

Trying to make FETCH happen >
Serverless, IoT, Azure Functions

Azure Friday | Learn all about Distributed Application Runtime >
Open source, Dapr

Xamarin: Creating a dark mode splash screen >
Mobile Development, iOS, Android

Events See all events

Cosmos DB Live Webcast / Online >
Expert-led, containers, .Net

OpenHack Serverless / Los Angeles >
In-person event, serverless, hack

Microsoft Ignite keynote – App development for everyone >
Demos, Java, PHP, Node.js, .NET, Python

Learning

Microsoft Ignite – Watch videos on demand >
Watch all keynotes, announcements, and sessions on demand

Create and publish a web application with Azure App Service >
Microsoft Learn, Express.js, Visual Studio Code