



Beginning PowerApps

The Non-Developers Guide to Building
Business Mobile Applications

—
Tim Leung

Apress®

Beginning PowerApps

The Non-Developers Guide to Building
Business Mobile Applications



Tim Leung

Apress®

Beginning PowerApps

Tim Leung
Reading, United Kingdom

ISBN-13 (pbk): 978-1-4842-3002-2
<https://doi.org/10.1007/978-1-4842-3003-9>

ISBN-13 (electronic): 978-1-4842-3003-9

Library of Congress Control Number: 2017961125

Copyright © 2017 by Tim Leung

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Cover image by Freepik (www.freepik.com)

Managing Director: Welmoed Spahr
Editorial Director: Todd Green
Acquisitions Editor: Joan Murray
Development Editor: Laura Berendson
Technical Reviewer: Jenefer Monroe
Coordinating Editor: Jill Balzano
Copy Editor: Karen Jameson

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit <http://www.apress.com/rights-permissions>.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/9781484230022. For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

Contents

- About the Authorxv
- About the Technical Reviewerxvii
- Forewordxix
- Acknowledgmentsxxi
- Introductionxxiii
- Part I: Getting Started..... 1
- Chapter 1: Introducing PowerApps..... 3
 - What Is PowerApps? 3
 - Who Is the Typical Developer?..... 4
 - What Are the Typical Uses? 4
 - Asset Checkout App..... 4
 - Budget Tracker..... 5
 - Service Desk App..... 6
 - Site Inspection App 7
 - Other Sample Apps 8
 - How Do You Build a PowerApp? 8
 - How Do Users Run PowerApp Applications? 9
 - What Data Can a PowerApps Consume? 10
 - Connecting to On-Premises Data..... 11
 - Common Data Service 11
 - Connecting to Other Data Sources 12
 - How Do You Write Code? 12

How Much Does It Cost?	13
Summary.....	14
■ Chapter 2: Subscribing to PowerApps	15
How to Get PowerApps.....	15
Obtaining PowerApps via Office 365	15
What Is Office 365?	16
Consumer vs. Business Office 365	16
Understanding Work Accounts and Personal Accounts	16
What Office 365 Editions Support PowerApps?	18
Subscribing to a PowerApps Stand-Alone Plan.....	18
Registering for an Account	19
Obtaining a Suitable Email Address.....	20
Logging onto PowerApps	21
Configuring Environments.....	21
Installing PowerApps Studio	22
Summary.....	24
■ Chapter 3: Creating Your First App	25
Preparing an Excel Data Source	25
Creating an App.....	26
Creating an Auto-Generated App.....	27
Creating Apps from SharePoint	28
Exploring the Designer	29
Running Your App	30
Examining the Screens at Runtime	30
Understanding the Auto-Generated App.....	32
Adding and Removing Data Sources	32
Adding, Deleting, and Rearranging Screens	33
Understanding How Screens Are Connected	34
Understanding Forms and Cards	39

Setting Project Properties	49
Saving Your Project	51
Opening a Project.....	51
Summary	52
■ Chapter 4: Sharing Apps.....	53
Sharing Your App	53
Adding a User	55
Applying Permissions	57
Managing Environments.....	57
Moving an App to a New Environment.....	57
Updating a Data Source.....	58
How Data Connections Work	59
Versioning an App.....	60
Installing the Mobile Player	60
Running with Foreign Languages.....	61
Summary	64
■ Part II: Refining Your Application	65
■ Chapter 5: Exploring Data Sources.....	67
Using the Common Data Service.....	67
Getting Started	68
Exploring the Common Data Model	70
Exploring a Standard Entity	71
Creating a Relationship	76
Using Field Groups.....	77
Picklists	78
Deleting CDS Data	81
On-Premises Data Gateway.....	82
Installing the Gateway	82
Starting the Gateway Service	83

Connecting to a Data Source	84
Uninstalling a Gateway	86
Using a SharePoint Data Source	86
Creating a SharePoint List	87
Connecting to a SharePoint Data Source.....	88
Other Data Sources	89
Microsoft Translation Data Source.....	90
Microsoft MSN Weather Services	91
Summary	92
■ Chapter 6: Using Formulas	95
Writing Formulas	95
Using the Function Tool	96
Working with Variables.....	98
Setting Screen Variables.....	98
Setting Property Values from Code	101
Setting Global Variables.....	102
Manipulating Data	102
Working with Text	103
Working with Numbers.....	107
Working with Dates	110
Working with Tables of Data.....	113
Defining Hard-Coded Sets of Values.....	113
Understanding Collections.....	113
Defining Tables	116
Working with Columns.....	116
Table Functions.....	117
Counting Records	118
Performing Aggregate Calculations	119
Conditional Statements and Logical Operations	120
Specifying Colors	121

Navigation Functions.....	121
Launching Websites and Apps.....	123
Retrieving Startup Parameters.....	124
Exiting an App.....	125
Managing Errors.....	125
Summary.....	126
■ Chapter 7: Customizing Screens	129
Using Predefined Layouts.....	129
Laying Out Tablet Apps	134
Understanding Sample Screen Designs	135
Building a List and Details Screen.....	135
Updating the Details Section of a Screen	138
Displaying a Delete Confirmation Screen	139
Building a Tab Control Screen.....	140
Using Themes.....	142
Summary.....	143
■ Chapter 8: Using Controls.....	145
Overview of Controls	146
Building Data Structures to Support Controls	146
Action Controls	147
Displaying Geometric Shapes.....	148
Displaying Text with Label and HTML Text Controls	149
Using Simple Controls	150
Text Input Control.....	150
Radio Control	151
Toggle and Checkbox Controls	152
Slider Control.....	153
Rating Control	154

Working with Dates	155
Date Picker Control	155
Setting Lookup Values	160
Limiting Input Values with a Drop-Down Control	160
Setting Lookup Values with a Drop-Down Control	161
Customizing the Drop-Down Display Value	162
Nesting Drop-Down Controls	163
List Box Control	164
Displaying Tables of Data	164
Using CDS-Specific Controls	165
Using Entity Form Controls	166
Drop-Down Controls with CDS	168
Working with Media Controls	170
Playing Videos	172
Playing Audio	173
Image Controls	173
Using the Timer Control	174
Resetting Form Controls	175
Summary	176
■ Chapter 9: Working with Data	179
Basic Behavior	179
Understanding Delegation	180
Searching Data	182
Basic Search Functions	183
Setting Multiple Sort Sequences	188
Additional Search Criteria Options	189
Joining Data	191
Joining Records - Showing Related Records	191
Checking for Nonexistence	192
Returning Distinct Records	193

Working with Data in Code	194
Retrieving Single Records	194
Updating Records	195
Creating a New Record	196
Deleting Data	200
Setting Default Screen Values	200
Validating Form Values	201
Checking Validation Rules	202
Sending Notifications	203
Summary	206
■ Chapter 10: Working with Images	209
Choosing Where to Store Images	209
Setting Up a Data Source	210
Storing Images in Excel	210
Storing Images in CDS and SQL Server Databases	210
Creating an Image App	211
Viewing Records in the CDS	212
Using the Add Picture Control	213
Using the Camera Control	213
Retrieving the Camera Data	215
Switching Between Cameras	215
Setting Brightness, Contrast, and Zoom Levels	216
Creating a Gallery	216
Using the Pen Control	221
Erasing Text	221
Converting Annotations to Text	222
Scanning Barcodes	223
Summary	224

■ Chapter 11: Mapping Data	227
Introduction to GPS	227
Retrieving Location Details	228
Introducing Signals	228
Freezing the Location Details	229
Obtaining Longitude and Latitude Readings	229
Saving Location Details	230
Switching the Signal On/Off	231
Other Signals	232
Displaying Maps	232
Using Google Maps	233
Bing Maps	234
Summary	235
■ Chapter 12: Using Charts	237
Introduction	237
Aggregating Data	239
Column Chart	240
Creating a Chart	242
Setting Legends	244
Setting Label Orientation	244
Applying Colors and Styles	245
Pie Chart	246
Line Charts	247
Grouping Data by Month	247
Showing Chart Data Dynamically	248
Showing Multiple Series	249
Summary	252

■ Chapter 13: Securing Apps	253
Authentication	253
Role-Based Security	255
Securing the CDS	255
Securing SharePoint Lists	257
Verifying Permissions Within an App	259
Configuring Record-Level Access	261
Setting Up the Data Source	261
Getting the Logged On User	262
Filtering Records by the Logged On User	263
Creating Data Loss Prevention Policies	264
Testing a DLP Policy	266
Summary	266
■ Part III: Maintaining Your Application	269
■ Chapter 14: Importing and Exporting Data	271
Using the Import and Export Controls	271
Exporting Data	272
Importing Data	273
Importing and Exporting Related Data	277
Importing and Exporting CDS Data	281
Importing Data	281
Exporting Data	283
Using the Excel Add-In	283
Importing Static Data	284
Summary	285

■ Chapter 15: Using Flow	287
What Is Flow.....	287
What Can We Do with Flow?	288
Creating a Flow	289
Copying Files	289
Calling Flows from Apps	292
Sending Notifications	293
Filtering Data	295
Working with Variables	297
Sending the Email.....	299
Manipulating Data	300
Constructing HTML Tables	300
Introducing Workflow Definition Language.....	301
Applying Workflow Definition Language	303
Using Output Values in Workflow Definition Commands.....	305
Referring to Items in Loops	306
Calling SQL Stored Procedures	306
Approving Actions	308
Managing Flows	309
Summary	311
■ Part IV: Extending Your Application	313
■ Chapter 16: Working Offline	315
What Happens Offline?.....	315
Techniques to Enable Offline Working	316
Saving and Retrieving Local Data.....	316
Making an App Available Offline	317
Setting the Data Source to a Local Collection	319
Handling Offline Deletions	321
Handling Offline Record Updates.....	322

Building the Synchronization Screen.....	325
Building the Conflict Resolution Screen	331
Summary	332
■ Chapter 17: Creating Custom Data Connectors	335
What Can We Accomplish with Web Services?	335
Overview of Steps	336
Understanding How the Web Works	336
Making HTTP Web Requests	337
Understanding HTTP Responses.....	337
Documenting a Web Service	338
Creating a Web Service Description	339
Using Postman.....	340
Using Swagger	340
Examining an Open API Document	342
Creating a Custom Connector.....	343
Using the Custom Connector.....	348
Summary	351
■ Appendix A: Data Models.....	353
Overview of CDS Entities.....	353
Foundation Entities.....	353
Person Entities.....	354
Group Entities	355
Sales Entities.....	355
Purchase Entities.....	358
Customer Service Entities	360
Human Resources Entities.....	361
CDS Field Names.....	362
Foundation Entities.....	362
Person Entities.....	362
Group Entities	364

Sales Entities 365

Customer Service Entities 368

Organization Entities..... 368

Purchase Entities..... 369

Human Resources Entities..... 370

Retail Entities..... 371

Data Structure of the Sample ‘Journey’ Application 371

Index..... 373

About the Author



Tim Leung is a software developer with vast experience in designing and building large-scale commercial applications. He is a recognized expert in the field of rapid application development and his previous publication with Apress (*Professional Visual Studio LightSwitch*, 2015) covered this topic in depth.

Tim is a Microsoft certified developer, a chartered member of the British Computer Society, and holds a degree in Information Technology.

About the Technical Reviewer



Jenefer Monroe is a Principle PM Manager at Microsoft, currently serving on the Customer Success Team for Power Apps, Microsoft Flow, and the Common Data Service. She previously served as Software Engineering Manager in the Excel team, and as a Software Engineer in SharePoint, Access, and Excel Online.

Foreword

Greetings readers! I am a program manager at Microsoft. I also served as the technical reviewer of Tim's *Beginning PowerApps* book. I would like to take a few minutes to share my insights on PowerApps, the role they play on meeting the business needs of Microsoft customers and partners, and what they can mean to you.

First, please allow me the opportunity to introduce myself. My role at Microsoft is that of a Customer Success program manager, which obviously means exactly that. I do everything in my power to make sure our customers succeed at meeting their unique business needs. Organizational-wise, I exist as a liaison between the customer and the product teams within engineering. Being here gives me the unusual perspective of being able to look from the outside in, as well as the inside out. My two primary objectives in this role are to work directly with customers to enroll them in the extensibility suite (otherwise known as the power suite), which refers to common data service, Microsoft Flow, PowerBI and PowerApps—the subject of this book. I hear firsthand about their challenges, be it the migration of InfoPath forms, the Access web app, or a brand new business still using paper. Sometimes I build pilots and proofs of concept... anything I can do to make their life easier. Wearing my other hat, I advocate for our customers and deliver stories to the internal team, sharing general and specific feedback including pain points, features that our customers love or hate, and anything in between.

I come from an engineering and development background. Prior to this role, I spent ten years (also with Microsoft) with Excel and five years prior on teams in Office, SharePoint, and Access. My most recent years with Excel were spent building the collaboration platform for Excel Desktop, which allows for the ability to have the rich desktop application edit a file at the same time as other rich desktop applications. Whew, talk about the hardest technical challenge of my life! So I stepped into this role in a similar fashion to other people on my team. We are very able to be technically knowledgeable about what is impeding customer success while keeping on top of what is getting rolled by the product team. It's an amazing role. I love it. I absolutely love it!

However, while I was head-down in collaboration, I looked up and I immediately recognized I had been so deeply entrenched in this singular technical problem for so long that I had some massive industry transformation. You know the cloud? Yes, that cloud. Well, I had somehow convinced myself that it was just for storing documents because that is how my product used it. Really? That was my wake up call. I began searching for the Holy Grail and found PowerApps.

I believe in PowerApps as an extensibility platform for business users. When I began scratching the surface of it, I realized that the myriad of opportunities it offers citizen developers (also referred to in many circles as the power user) is nothing short of mind blowing. It offers an opportunity to utilize existing skill sets to do something entirely new. Just as an example, say you have the tools to develop professional looking PowerPoint decks. Then, like magic, you can apply those exact same skills to PowerApps to develop beautiful professional-looking business applications. Or, if you are adept at making formulas in Excel, or mingling data with a look up, or excel at doing logic flows with if statements, or doing mathematical operations on data to summarize... all these skills transform to PowerApps and can be used to develop powerful logical flows within your business application. Giving power users the resources to create an application that can show up on Android, and iOS, and Windows and even in a browser, now that's life changing.

Let's talk about accessing data, and more specifically, connectors (160 available at time of writing) that allow power users to connect to things like SQL in SharePoint, and external applications like Twitter and Facebook as well. You cannot do that in PowerPoint and Excel. Now I can go into Active Directory and see who you are, and I can pull your picture through the Office 365 connector simply by learning a couple of interfaces that are readily available. You can build a really slick UI for your team that shows pictures and other information. These connectors are a key new tool for the power user. And, wait, that's not all! While we're doing all of this cool stuff with connectors and cross platform development in PowerApps, we are actually keeping our IT and admin colleagues in mind. Think about it. Massive user bases withstanding, the ability for IT Pros and admins to create governance models around the usage of Excel and Access never happened. Admins and IT departments hated Access and Excel because, in spite of them being user and business friendly, it left the governance of those IT needs completely out of the hands of IT. Not so with PowerApps! Right out of the starting gate, we are ensuring that IT and admins feel comfortable with data-loss policies and monitoring features. In fact, I would even go so far as to call it a pillar in PowerApps. You have to admit, the ability to see where the viral usage of this product is happening, and saying, "I have control over it," is pretty darn compelling.

PowerApps fits into the Microsoft business intelligence space as part of the new extensibility platform. We also sometimes refer to it as the power suite, and in true Microsoft style, we continue to "evolve" naming conventions. Greatly simplified, I ask you to think of it in three parts: Measure, Act, and Automate. PowerBI is the Measure. It is how we expect people to measure their data. For example, it works for small transactional style data (e.g. someone wants to see machine orders), but it also works for giant flows of data, like from IoT. Power Apps is the Act. So now that I know this data that I have pulled together, what is the human decision-maker going to do? If we refer back to the example of the hardware order app, it would be something like, "which machine shall I query?" Flow is the Automate part of the platform. It is how you go and take some of those processes for immediate action. Some don't actually require human intelligence; they're just automatic—like if you trigger an approval every time a new lead is found. Related, Common Data Services is a topic on its own. Underneath it all, that has to do with the extensibility of data. So it's essentially the data service, SQL for the Power User, and the focal point of data.

PowerApps has been evolving for the last couple of years. Many of you probably remember when PowerApps used "Sienna." Well Tim has been working with PowerApps since way back when. I believe he has more knowledge and context about this product than many of us on the team. He was one of the first people I met in my new role. I can say with confidence that the book you are holding in your hands is a phenomenal one. Tim, by nature, is such a detail-oriented and inquisitive person. But don't just take it from me; the proof is in pudding. Go ahead and take a quick look at the offline chapter—the one on how to utilize caching for offline scenarios. If you think about Power apps as a mobile interface for data, the ability to do small periods of offline reporting is really important. Let's say, for example, that you are inspecting oil rigs. You might not have constant access to the Internet. In order to sync the data in real time, you may need to cache it locally for small periods of time to submit later. Now this is not something Microsoft natively supports, but Tim's book offers a workaround. While a somewhat complex task, Tim has laid out the process in an intuitive and engaging manner. In fact, when I shared it with the team, they were notably impressed. Tim's book does not just impress in that one chapter. It also offers readers who are interested in learning PowerApps a means to do so from A to Z. He offers a comprehensive learning path for PowerApps not seen in depth or breadth in any articles, documentation, or sites anywhere, not even in the coursework that is available online. To me, *Beginning PowerApps* hands down has the best content available to get yourself to the 200 level. Read it, and by the time you are done, hopefully he will have a 300 level book in the queue ready for publication. Enjoy!

Jenefer Monroe
Microsoft
November 2017

Acknowledgments

First, I'd like to thank Apress for the opportunity to publish this work. I especially want to thank all the members of my editorial team for their hard work. This includes Todd Green, Joan Murray, Jill Balzano, and Laura Berendson.

Equally, I'd like to thank Jenefer Monroe, Darshan Desei, and all the other team members at Microsoft who have worked on my project behind the scenes.

Finally, I'd like to thank the PowerApps team for developing a great product and to acknowledge all of those who have contributed to the PowerApps community.

Introduction

Welcome, and thank you for choosing this book. Within its contents, you'll learn how to build functional apps and how to avoid the common mistakes that beginners often make. The contents of this book are organized into three logical sections:

- The first four chapters describe how to build and publish an app. This book assumes no prior knowledge and therefore, we'll start at the very beginning. This includes how to subscribe and to install the requisite software. Within these chapters, we'll cover elementary topics such as how to retrieve and update data from a data source, how to build screens, and how to configure the navigation linkages between screens.
- The main section of this book describes how to embellish our apps. We'll look at how to use a cloud-based data store called the common data service. We'll also cover practical tasks including how to use screen controls, how to capture photos, how to retrieve location details with GPS, and how to display data with charts. This section of the book also contains a reference guide of formulas. These are very important because they provide the means for us to program PowerApps.
- In the final section of this book, we'll cover some advanced techniques. This includes how to configure security, how to import and export data, and how to build apps that can work offline. We'll also examine how to use Flow, a workflow service that can carry out tasks outside of PowerApps. The final chapter describes how to expand the capabilities of PowerApps by calling third-party web services.

To provide some context to the concepts in this book, we'll refer frequently to demonstration app that records journey details. This typifies the sort of application that businesses use to record mileage expense claims. In technical terms, this app enables us to explore a wide range of data types. The data structure that supports this app stores user and vehicle details, and we can use this structure to explore techniques such as how to use drop-down controls, and how to assign lookup values.

Figure 1 illustrates the type of screen that we can build with this data structure.

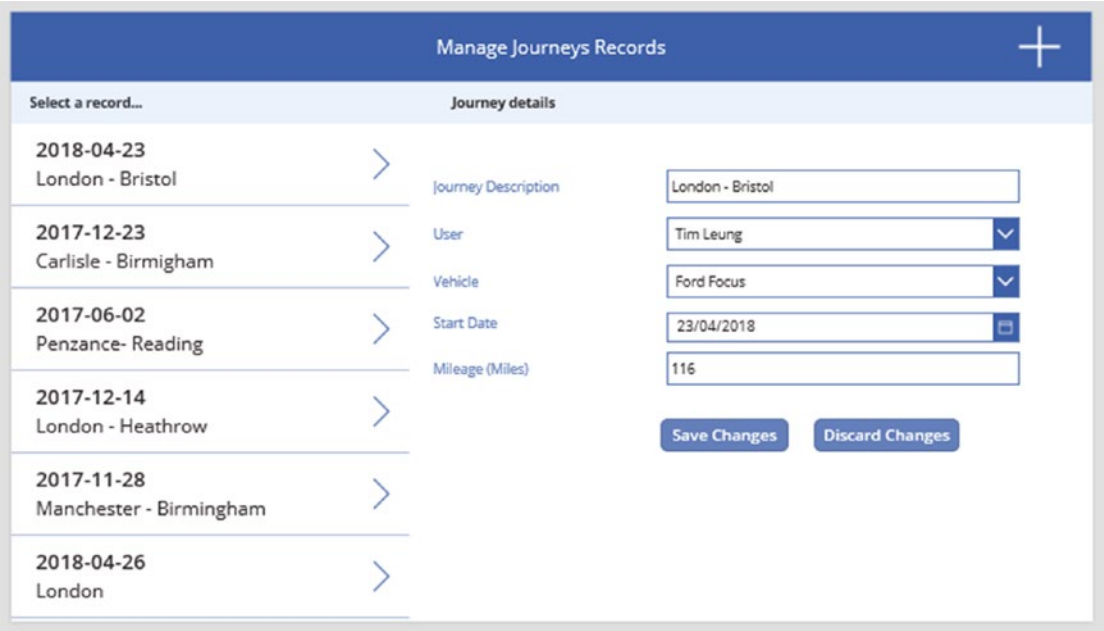


Figure 1. Screen from example app

Versioning

PowerApps benefits from frequent updates. Microsoft usually updates PowerApps on a three-week cycle. These updates include new features and bug fixes. Because of these frequent updates, menu item locations can change over time, and the screenshots in this book may quickly appear out of date. Please don't be too alarmed by this! The core concepts and principles in this book are likely to remain the same, despite any additions and cosmetic changes that take place afterward.

Erata

Although we take great care to verify the contents of this book, there is a slight chance that it might contain some errors. Therefore, the product page for this book contains an up-to-date list of any problems that we identify.

www.apress.com/9781484230022

Finally

We hope you find this book useful, and we wish you much success in building some great apps with PowerApps.

PART I



Getting Started

CHAPTER 1



Introducing PowerApps

Even if you are completely new to PowerApps, you probably know that PowerApps is a tool for building mobile device applications. Additionally, you'll understand that PowerApps is simple to use and allows you to build applications very quickly. But beyond these basic concepts, you might not fully understand the specific capabilities of PowerApps, or how, exactly, a PowerApps application looks or feels. In this chapter, we'll clarify the things that we can do with PowerApps by looking at the sample apps that Microsoft provides. The purpose of this is to clarify whether PowerApps can solve a specific business problem. Some of the basic topics that we'll explore in this first chapter will include the following:

- What we can accomplish with PowerApps
- Examples of applications
- How to get started with building apps

What Is PowerApps?

From a high-level perspective, PowerApps is a subscription-based service for building applications. The apps that we build can run through a web browser, and are therefore compatible with a wide range of devices. However, these apps work best on a mobile or tablet device. A typical app will connect to a data source and include screens to view and edit data. A typical app also includes functionality to help facilitate a business process.

Historically, PowerApps evolved from a Microsoft project called Siena. Microsoft developed Siena during the Windows 8 era. Siena provided a platform for programmers to build new style 'metro' apps that connected to data.

Microsoft provides sample applications you can use to learn PowerApps, and we'll look at these later in this chapter. These apps include service desk, budget tracking, and site inspection applications. One thing you quickly realize is that PowerApps is a business tool. It isn't a tool for building consumer grade applications – for example, apps or games that you want to sell via app stores.

A key characteristic of PowerApps is the way in which it embraces the cloud. Specifically, apps can access data from OneDrive, Dropbox, Salesforce, Dynamics 365, and many other cloud providers. Another characteristic is the product's ease of use. You don't need to be developer to use PowerApps. In fact, the product feels highly inspired by Microsoft Office. It includes a visual designer and we use Excel like formulas to build functionality. In terms of deployment, the concept is that we share PowerApps in the same way that we would share an Office document.

Who Is the Typical Developer?

Microsoft designed PowerApps for non-developers – users who are not professionally trained software programmers. The target demographic includes managers or office workers who work regularly with Microsoft Office. Market research shows that demand for mobile applications will outstrip the availability of qualified programmers. Therefore, the goal of PowerApps is to address this problem by making it easier for users with basic IT literacy to build mobile apps.

At this early stage, the product seems to attract a large number of professional users from the Microsoft Office community. This includes users with Office 365, Access and Access Web Apps, and SharePoint experience. In part, this is because these users have easy access via SharePoint and Office 365.

The product is likely to attract Microsoft InfoPath users – a product that has been discontinued by Microsoft. InfoPath is a tool for adding forms over data. Personally, I chose PowerApps as a LightSwitch alternative – this was a Microsoft product for rapid application development.

For more complex applications, there may be tasks that require the skill of an experienced developer. Such tasks include connecting apps to data sources that are not natively supported, configuring application security, and building flows. A flow is a set of tasks that runs, following an event in an app. For example, you can use flows to send emails, copy files, or to add items to a SharePoint list.

What Are the Typical Uses?

What applications can you build with PowerApps and what tasks can those applications carry out? To help us better understand what PowerApps can do, let's look at some of the sample apps that Microsoft provides. These include the Asset Checkout, Budget Tracker, Cost Estimator, and Inventory Management apps.

Asset Checkout App

The Asset Checkout App enables users to check out items. However, this isn't the e-commerce tool that the 'checkout' part of the name suggests. It's more like a system that enables you to borrow hardware items from a library. That being said, the app demonstrates some useful features. The most noteworthy is the integration of product images into the application (Figure 1-1). The first screen contains UI features such as search, a horizontal scroll control, and a tab control. The product image strip on the home screen shows the most recently checked-out items. Therefore, logic exists in the application to perform this type of query.

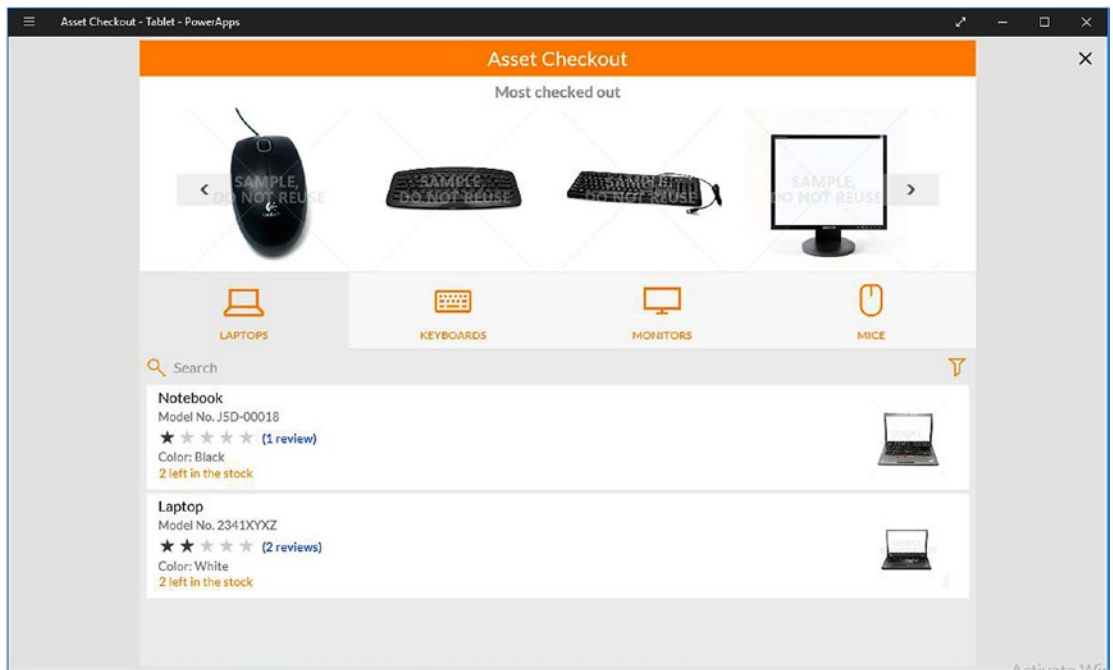


Figure 1-1. Asset Checkout App

Budget Tracker

The Budget Tracker App highlights the mathematical tasks that you can carry out in a PowerApps application. The home screen shows calculations that are conditionally formatted with red and green fonts. It also features a pie chart that illustrates a breakdown of expenses (Figure 1-2). With this app, users can add and delete expense records. Therefore, we can learn how to program data tasks such as adding or deleting records through this app.

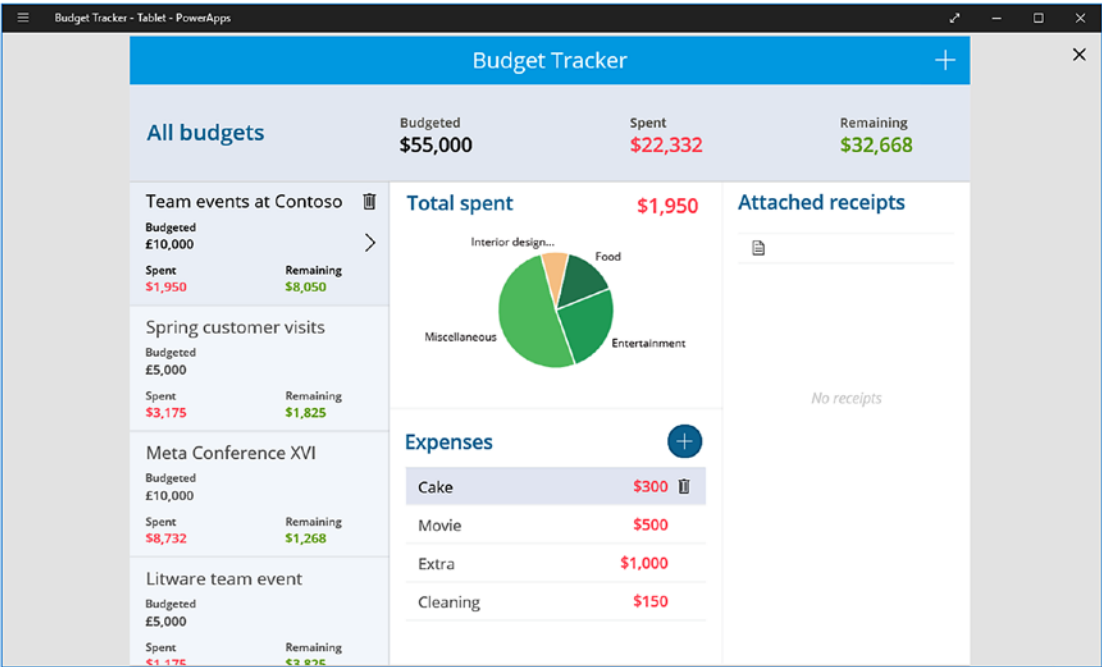


Figure 1-2. Budget Tracker App

A highlight of the budget tracker app is the ability for users to capture pictures of receipts, and to assign those receipts to an expense claim. This feature demonstrates how PowerApps can integrate natively with cameras that are built into mobile devices.

Service Desk App

The Service Desk App is a nice example of a data entry application. The purpose of this app is to manage the support tickets that arise through a help desk department. I like this application because it demonstrates the data structures that developers typically expect to see in database applications. For example, the application allows users to assign a priority rating, and an area (or department) to each support ticket, therefore illustrating the concept of one-to-many data structures. See Figure 1-3.

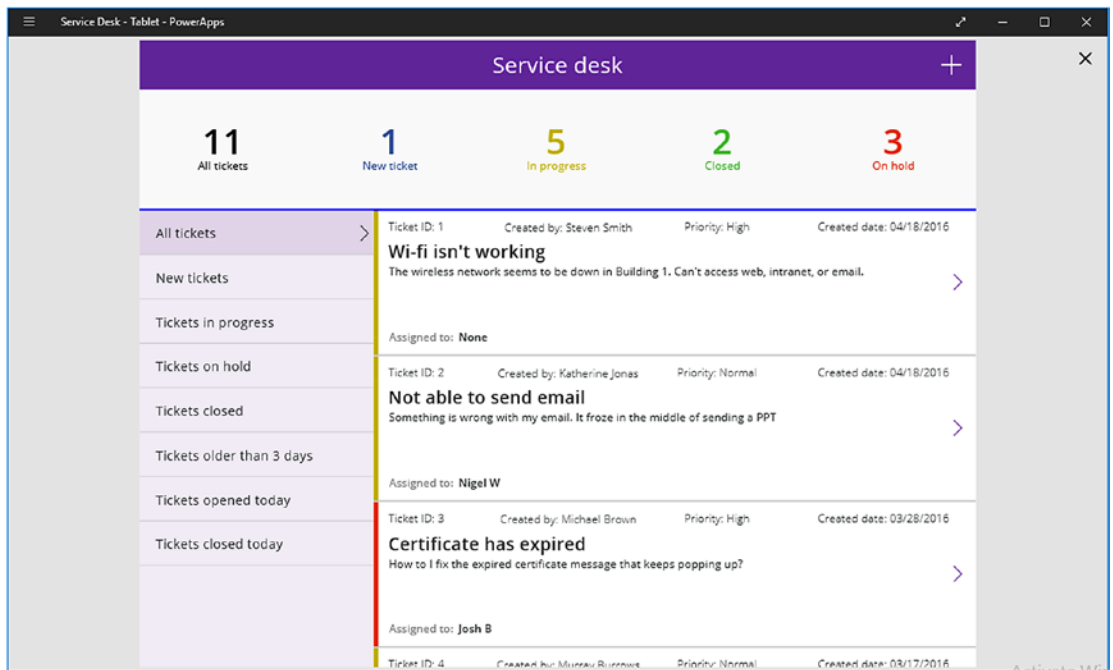


Figure 1-3. Service Desk App

Site Inspection App

The Site Inspection App demonstrates integration with location and mapping services. With each site inspection record, the user can use a device camera to capture multiple images. This one-to-many relationship of image data can be useful in the apps that you build. The Site Inspection app also enables users to record the current location using the GPS on the mobile device. For each record, the app can also integrate a map that shows the location. See Figure 1-4.

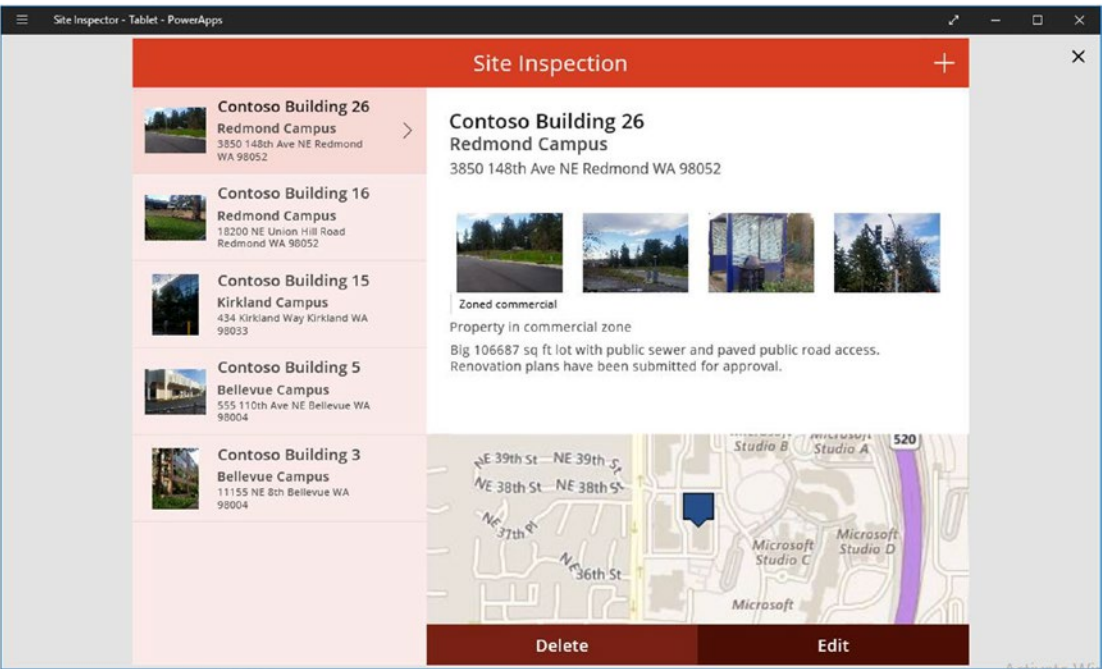


Figure 1-4. Site Inspection App

Other Sample Apps

The remaining sample applications are characterized by data features with a similar theme, namely, the ability to select a record from a list, and to view and edit the selected record. Other notable apps include the following:

- Product showcase app. – This is a well-presented app that looks great and demonstrates how to display videos in apps.
- PDF Reader App – This app allows users to view PDF documents from within the app.
- Suggestion App – This app implements some basic role maintenance. You can add users to an administrator role, and those users can carry out additional tasks in the application.
- Case Management App – A feature of this app is that it stores data using the Common Data Service (CDS). I’ll describe this in more detail later in this book.

How Do You Build a PowerApp?

PowerApps Studio is the tool for building apps. There are two versions available – a version that runs on Microsoft Windows, and a version that runs through a web browser. The Windows version is a store app and requires Windows 8.1 or Windows 10. The web-based version is slightly more limited than the Windows version. Note that the Windows App doesn’t support Windows 7. Therefore, Windows 7 users will need to use the web version of PowerApps Studio. There are currently three supported browsers – Microsoft IE11, Google Chrome, and Microsoft Edge. Firefox and Safari are not supported.

Figure 1-5 shows a screenshot of the Windows version of PowerApps Studio. This image highlights how we can design applications using a visual designer. At the top of the designer, there is a ribbon menu bar that looks like the ribbon bar that exists in Microsoft Office. Within the designer, we can see a visual representation of the screens and controls as we build our app.

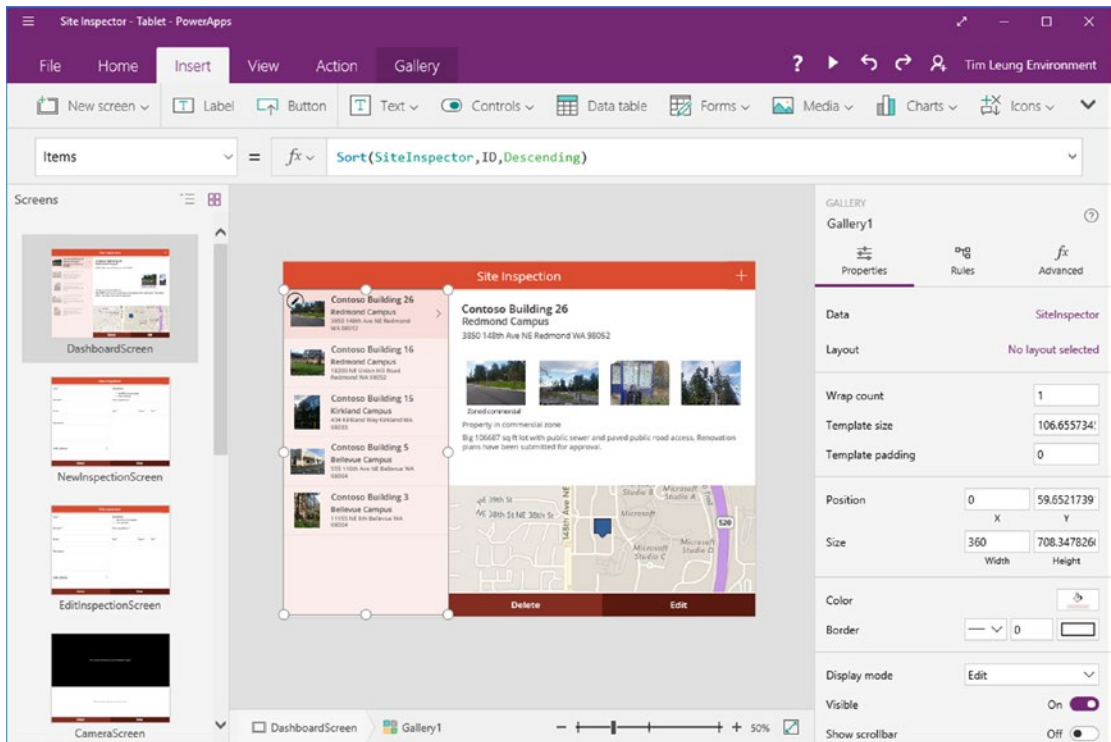


Figure 1-5. PowerApps Studio

Tip The Windows version of PowerApps Studio is a Store App and therefore, we can only run one instance of the app at a time. To work around this limitation, we can use PowerApps Studio for Windows for our main project, and open an additional project in the web version of PowerApps Studio. While we're learning, we can use this technique to open a sample project in a separate window, and we can easily switch to the sample project to refer to formulas and layout.

How Do Users Run PowerApp Applications?

Once we build an app in PowerApps Studio, we can grant users access to our app. Users use a PowerApps 'runtime' app to run apps on mobile devices. These runtime apps are called 'players', and players are available for Android, iOS, and Windows Mobile devices. Figure 1-6 shows the section on the PowerApps website where we can download the PowerApps software.