

1 - Introduction

These App Inventor 2 Guides introduce the fast and simple way to develop Android apps using MIT App Inventor, the software development system that runs in your browser, for creating apps that run on your smart phone and tablet.

Volume 3 is focused on using App Inventor with databases: *TinyDB*, *TinyWebDB*, *Fusion Tables* and *Data Files*.

This e-book series is targeted at adult learners (high school or secondary school and up) who are interested in producing real world applications (rather than games). Hence, these books focus on features that are suitable for creating business, productivity and educational applications.

Volume 3 assumes the reader is familiar with the basics of App Inventor programming; start with Volume 1 if you are brand new to App Inventor.

Each example provides code snippets you can adapt or modify for use in your own programs so that you can begin using *TinyDB*, *TinyWebDB* or *Fusion Tables* and data files.

TinyDB

This volume repeats the introductory *TinyDB* section of Volume 2 (in case you do not have Volume 2) and then expands that original app to storing and working with lists in *TinyDB*, and then shows how to simulate the use of multiple tags in a *TinyDB* database. In *TinyDB*, a tag is like an index keyword in the back of a book – you can find information quickly by searching the index. But *TinyDB* supports only one tag – what if you need multiple index keywords to locate information in your database? In this volume, I show you how to do this.

Each device has only one *TinyDB* database. But there are scenarios where having more than one *TinyDB* would be helpful. I show you how to simulate having several *TinyDB* databases.

TinyWebDB

TinyWebDB is similar to *TinyDB* but stores the database at a web server in the Internet cloud. This opens up many possibilities, such as updating app data without having to install a new app version. Apps simply read their data from the Internet database.

Multiple users may access data in a *TinyWebDB*. This means a group of users may simultaneously use shared data (with some limitations) – imagine a group of 30 students taking a quiz using their Android tablet. The quiz is stored in the cloud in a *TinyWebDB*. After taking the quiz, the student's score is stored in the cloud database.

TinyWebDB has limitations: Unlike *TinyDB*, where apps can fetch a list of all tags used, there is no similar feature in *TinyWebDB*. But I show how you to simulate the tags list with app code.

TinyWebDB is a database engine that runs in the cloud, on a server. I show you how to set up and manage your own *TinyWebDB* server using free services of Google Cloud services.

Fusion Tables

Fusion Tables are a cloud-based, high performance, high capability database for use by a wide variety of applications, not just App Inventor apps. Fusion Table databases are provided by Google Cloud services and run on Google Servers. Fortunately, small databases may be created and used, for free. Larger databases with many users can also be set up, for a subscription fee.

Apps that use Fusion Tables must use Google Cloud service management features to create and define their database, and to obtain special codes (such as “API keys” and “Table IDs”) to programmatically access the tables. The steps to do this are moderately complex – the examples in this text walk through the process in step-by-step fashion.

Reading and writing data to and from a Fusion Table relies on a special language called Structured Query Language, which is abbreviated as “SQL”, and in turn, is pronounced as “sequel” and rarely as “S – Q – L”. Basic operations using SQL are easy to learn and understand. But SQL has powerful capabilities for searching, retrieving and doing calculations on the data in the database. This Guide only presents basic operations for reading, searching and writing of data in the SQL database.

Which database component to use?

In terms of ease of use, *TinyDB* is the simplest. However, *TinyDB* saves data on the device in which the app is running and there is no way to share data with other app users.

TinyWebDB has most of the simplicity of *TinyDB* – but stores its data in the cloud. That means *TinyWebDB* apps may use centralized data (such as a standard price list or inventory list) or share data with each other by reading and writing data in the *TinyWebDB* database. On the downside, *TinyWebDB* access is more complex than *TinyWebDB* and requires setting up a *TinyWebDB* database in the cloud. The setup can be done for free, for modest applications, using Google Cloud service but this too is bit more complex than using *TinyDB*.

Fusion Tables are the most powerful database feature available to App Inventor programmers. Fusion Tables are a cloud-based, Structured Query Language (SQL) based information system that provide search and retrieval features not available in either *TinyDB* or *TinyWebDB*. But this comes at the cost of a moderately complex procedure to set up a cloud-based Fusion Table (or tables) for your application. Further, you need to learn a little about SQL in order to create, read, update and delete records in the database.

Basically, there is much more to learn, but Fusion Tables are the real deal in database technology and provide the power to implement some complex database functionality.

Choosing Which Database to Use

Which method you use depends on your app requirements, your ability, and your time. Chances are good that you can set up *TinyDB* for your app quickly and with limited App Inventor experience. But as the database features expand, so does the complexity of programming and the knowledge required to put the features to use.

TinyDB

Use *TinyDB* to store data between executions of the app. Normally, when an App Inventor app is no longer running, its internal data is erased. But data stored in *TinyDB* is retained for future use.

Use *TinyDB* when data is only needed on the device, and is not shared with others.

Use *TinyDB* to share data between “screens” within App Inventor applications.

TinyDB is easy to use.

TinyWebDB

Use *TinyWebDB* when your app must use or share data that is stored off the device, in the Internet cloud.

TinyWebDB is harder to use because the cloud service must be setup and configured, and programming access to the database is slightly more complex than with *TinyDB*.

TinyWebDB requires the device to have an Internet connection.

Fusion Tables

Use Fusion Tables when your app needs to use complex search criteria.

Use Fusion Tables when your app needs or can use cloud-based database services.

Use Fusion Tables to leverage the powerful capabilities of the SQL query language.

Use Fusion Tables to for data that will be exported to CSV format files for later use in spreadsheets, or for importing spreadsheet data.

Use Fusion Tables for applications requiring large, fast database services. Google sells inexpensive Fusion Table capacity services for business customers; prices for data storage and CPU time used are inexpensive.

Fusion Tables are more difficult than *TinyWebDB* to setup and configure.

An Internet connection is required to access a Fusion Table.

Text Files

Each Android device has a hidden file system similar to the way files are stored on Windows, Mac OS X or Linux. App Inventor provides for reading and writing text files. Data that is stored in text files may be read by other apps, or shared in the comma separated values (CSV) format with spreadsheet apps.

Files may be transferred from your device to a computer. For example, an app could record observations and data in CSV file format. Later, the CSV file may be transferred to a personal computer and read in to a spreadsheet program for analysis.

Conventions

What do we call an Android device? Sometimes this text will say “Android device” or “smart phone” or “phone” or “tablet”. In each case, they mean the same thing: a device running Android which permits the installation of Android apps designed for small devices such as phones and tablets.

What do we call an app? The term *app* is commonly used for a program running on a smart phone or tablet. The terms “app”, “application”, “program” and “software” are used interchangeably in this guide.

What is a browser? App Inventor is an Internet-based app development system that is accessed using an Internet browser such as Firefox, Google Chrome, Internet Explorer, Safari or Opera. In this book, “browser” refers generically to your choice of Internet browser.

What does cloud-based mean? The term *cloud* comes from the use of a drawing of a *cloud* (as in clouds in the sky) in illustrations about how computer networks work. When a computer in one city connects through the Internet to a web server in a distant city, the Internet connection is often illustrated using a picture of a cloud.

A computer connects to the cloud at one end and a web server connects to the cloud at the other end. The simple cloud drawing intentionally hides the details of potentially many intermediate network connections, routes and network equipment that may be used to transfer data from one end of the link to another.

When *a service is cloud-based*, it means the service is located on one or more servers *somewhere in the Internet*. The term *cloud-based* describes many kinds of services – Google Docs, Microsoft Office365, Gmail, YouTube videos, and MIT’s App Inventor Android software development system are just some of many examples.

Method and event handlers are sometimes written in a form similar to *btnAdd.Click*, which means the *Click* event handler of the button named *btnAdd*. Calls to methods are sometimes written in a similar way, such as *TinyWebDB1.StoreValue*, meaning to call the

method named *StoreValue* defined for the *TinyWebDB1* component. In this text, a method may sometimes be referred to as either a method or a procedure.

References to components in the Designer and the Blocks editor are sometimes shortened. For example, when a *TinyDB* control is added to the Designer, this appears as “TinyDB1”. In the text, the “1” will usually be omitted when it is apparent which control this refers to.

About the Sample Apps

The sample apps presented in this text are designed to illustrate programming concepts and methods. The user interfaces are kept simple – the focus is on the blocks code that works with the database software. Creating fancy user interfaces adds complexity and clutter. By keeping the user interfaces simple, you can focus on the database programming and learn tools to add database features to your own apps. Ultimately, you can design whatever user interface is best for your own applications – and incorporate the database programming features that you need to make your apps work for you.

Sample source code for most apps in this book are available online. The name of the source file containing the example code is located in each chapter.