



Drills - TinyWebDb

Drills - TinyWebDb

Coding exercises that focus on using the TinyDb and TinyWebDb components. The UI allows the user to input a tag/value pair and includes buttons to store and retrieve values from TinyDb and TinyWebDb.

CSP Learning Objectives:

- *The student can implement a correct program.*
- *The student can explain how a program implements algorithms.*

Setup

Click [here](#) to open App Inventor and import the [TinyWebDb Drills Template](#). Use the Save As button to rename your project *DrillsTinyWebDb* or something like that.



The UI

The UI for this app consists of *TextBoxs* for inputting data and *Buttons* for storing and retrieving the data that will be stored in the Database. Both the *TinyDb* and *TinyWebDb* components have been incorporated into the app. Notice that the *ServiceURL* for the TinyWebDb component is set to: <http://cpssc110-db.appspot.com>. This is a custom instance of TinyWebDb, not the default instance that comes with App Inventor.

The screenshot displays the App Inventor interface with three main panels: Viewer, Components, and Properties.

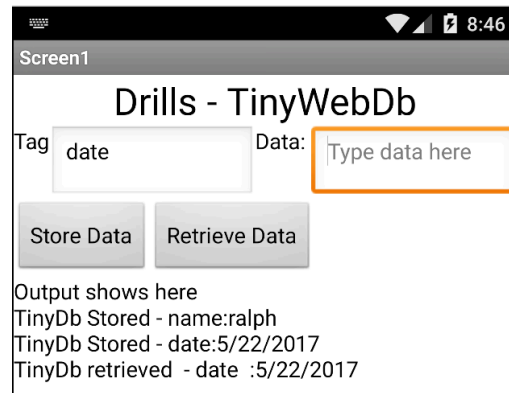
- Viewer Panel:** Shows a preview of the app's UI. At the top, there are checkboxes for "Display hidden components in Viewer" and "Check to see Preview on Tablet size.". The UI itself has a title bar "Drills - TinyWebDb". Below it, there are two input fields labeled "Tag" and "Data:". Under these fields are two buttons: "Store Data" and "Retrieve Data". A large text area below the buttons is labeled "Output shows here". At the bottom of the viewer, there is a "Non-visible components" section showing "TinyDB1" and "TinyWebDB1", with "TinyWebDB1" highlighted by a green box.
- Components Panel:** Lists the components used in the app. The hierarchy is: Screen1 (LabelBanner, HorizontalArrangement1 (LabelTag, TextBoxTag, LabelData, TextBoxData), HorizontalArrangement2 (ButtonStore, ButtonRetrieve, LabelOutput), TinyDB1, and TinyWebDB1). At the bottom of this panel are "Rename" and "Delete" buttons.
- Properties Panel:** Shows the properties for the selected "TinyWebDB1" component. The "ServiceURL" property is set to "http://cpssc110-db.appspot.com".
- Media Panel:** Located at the bottom right, it contains an "Upload File ..." button.



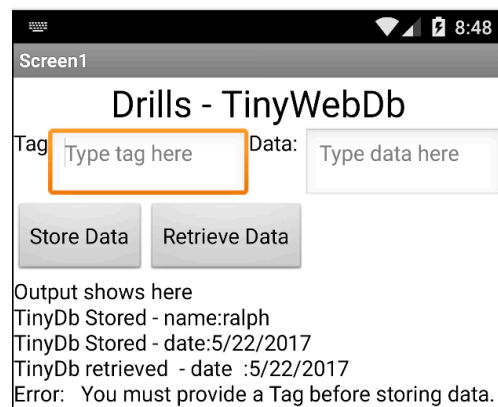
Drills

For each of the following problems, code the solution in the Blocks Editor. Then run and test your solution. Keep running and testing until you are certain that it is correct. If you have trouble or questions, ask someone for help. For each exercise, you might want to do 'SaveAs' to preserve your individual solutions.

1. **Store and Retrieve in TinyDb.** Implement the *ButtonStore* and *ButtonRetrieve* buttons to store and retrieve data from **TinyDb**. When *ButtonStore* is clicked, the app should store the value typed in *TextBoxData* into the TinyDb under the tag *TextBoxTag*. Use *ButtonStore* to store some name:value pairs -- for example, name:ralph, date:5/22/2017. When the *ButtonRetrieve* is clicked the app should retrieve the value stored under the tag *TextBoxTag*. For example (in the example shown here), when the user types 'name' as the tag and presses ButtonRetrieve, the app displays whatever value was stored under that tag, in this case 'ralph'. It should display the test cases one test case per line, as shown here:



2. **Error Check.** Modify your previous solution to perform an error check to guard against the user storing or retrieving data when *TextBoxTag* is empty. To alert the user, print an error message in the *LabelOutput*. When storing data it is okay if *TextBoxData* is empty. In fact, that would be one way to clear that item in the Db. Notify the user with an error message in the *LabelOutput* if they've forgotten to type a value into the tag box





3. **Retrieve TinyWebDb.** Modify your solutions to the previous exercises to retrieve the data associated with *TextBoxTag* from the **TinyWebDb**. Test data has already been stored in <http://cpssc110-db.appspot.com> under the tag **tinywebdbdrill**. **NOTE:** If you correctly enter that tag and no data is returned, see below for how to enter your own data directly into TinyWebDb through its web interface.

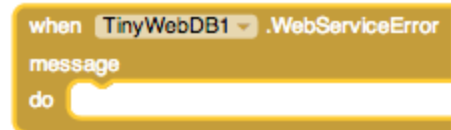
Screen1

Drills - TinyWebDb

Tag: Data:

Output shows here
Error: Please provide a tag to retrieve
TinyWebDb retrieved - tinywebdbdrill :this is a test

4. **Web Service Error.** It's a good idea when dealing with a Web database to use the *WebServiceError* block. It will report various kinds of errors, including garbled data: Add this block to your app and test it on the (case sensitive) tag **tinywebdbERR**. The data stored at that tag is the string *This should cause an error because it's missing and end quote*. Experiment with this to see whether you can discover other errors that cause "garbled data".



Screen1

Drills - TinyWebDb

Tag: Data:

Output shows here
The Web server returned a garbled value for the tag tinywebdbERR.



5. **Store and Retrieve Web Db.** Modify your solution to the previous exercise so that when the user inputs a tag/value pair, the data should be stored to and retrieved from both databases.

Screen1

Drills - TinyWebDb

Tag Data:

Output shows here
The Web server returned a garbled value for the tag tinywebdbERR.
TinyDb & WebDb Stored - test:testing
TinyDb Retrieved - test:testing
TinyWebDb retrieved - test :testing



Storing and Retrieving Directly from TinyWebDb's Web Interface

As you may know, TinyWebDb is implemented as a **web service** -- that is, it is a web application that interacts with a client, such as your mobile app or your browser. For our mobile apps we've been interacting with TinyWebDb through App Inventor code. However, it is possible to interact directly with TinyWebDb through a browser tab. Here's what the interface looks like, when you point your browser to : <http://cpsc110-db.appspot.com/>

App Inventor (TinyWebDB) Web Database Service



This web service stores and retrieves values for an [App Inventor for Android](#) app. App Inventor apps can access this service using the TinyWebDB component and setting the ServiceURL to the URL of this site.

Search database for a tag

Tag:

Returned as value to TinyWebDB component: **Test**

Store a tag-value pair in the database

Tag:

Value:

Key	
PIuWezGVfkjYbZ	Hd1UYU2O

It may not be so easy to see from this screenshot, but there are several input text fields. They support two operations:

- **Search database for tag.** In this case if you type a tag, the service will respond with its associated value.
- **Store a tag-value pair in the database.** In this case you can store data in the TinyWebDb by providing a tag/value pair.

One advantage of this is that you can use the web interface to experiment with data before implementing it in your code.