MAKE-UP LECTURE

BAU
Bahçeşehir University

CET4104 – MOBILE PROGRAMMING

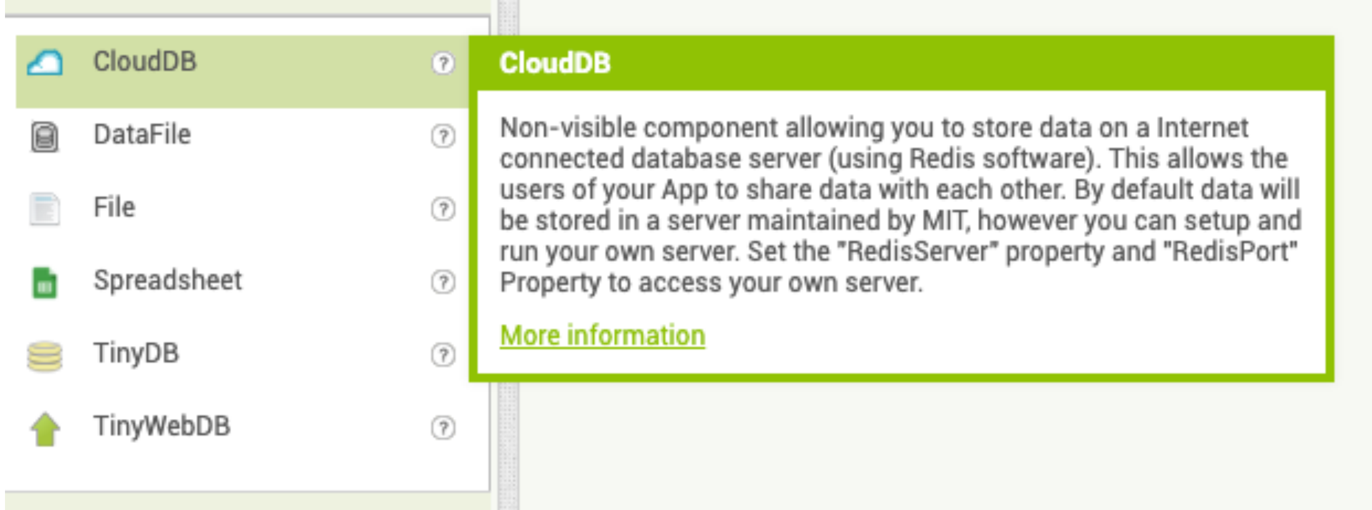
Week 11 – 19.05.2023

Dr. Mustafa COŞKUN
mustafa.coskun@ou.bau.edu.tr

OUR COURSE WILL START AT 12:20, PLEASE BE READY

I am waiting for the participants to join

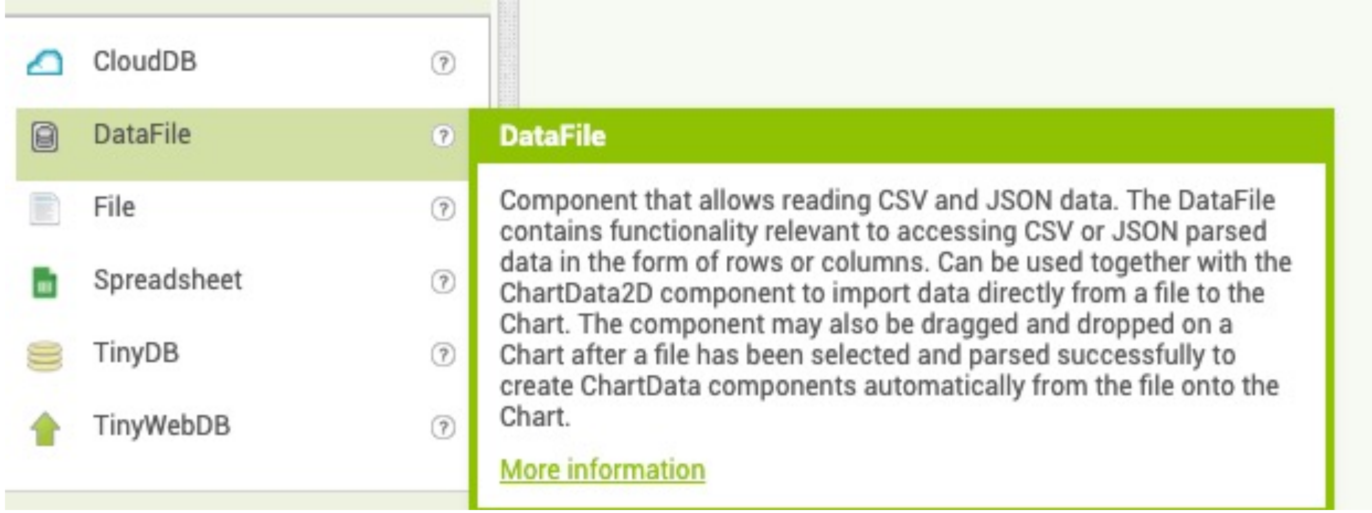
APP INVENTOR DATABASE AND FILE OPERATIONS



CloudDB

Non-visible component allowing you to store data on a Internet connected database server (using Redis software). This allows the users of your App to share data with each other. By default data will be stored in a server maintained by MIT, however you can setup and run your own server. Set the "RedisServer" property and "RedisPort" Property to access your own server.

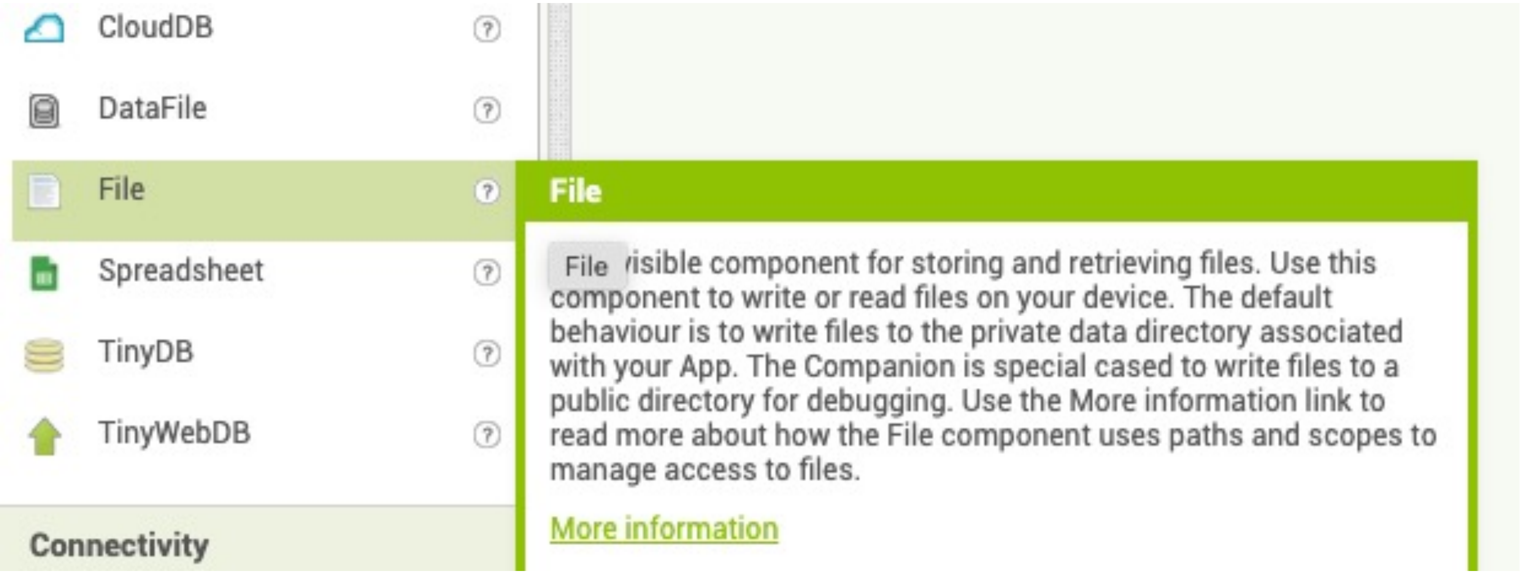
More information



DataFile

Component that allows reading CSV and JSON data. The DataFile contains functionality relevant to accessing CSV or JSON parsed data in the form of rows or columns. Can be used together with the ChartData2D component to import data directly from a file to the Chart. The component may also be dragged and dropped on a Chart after a file has been selected and parsed successfully to create ChartData components automatically from the file onto the Chart.

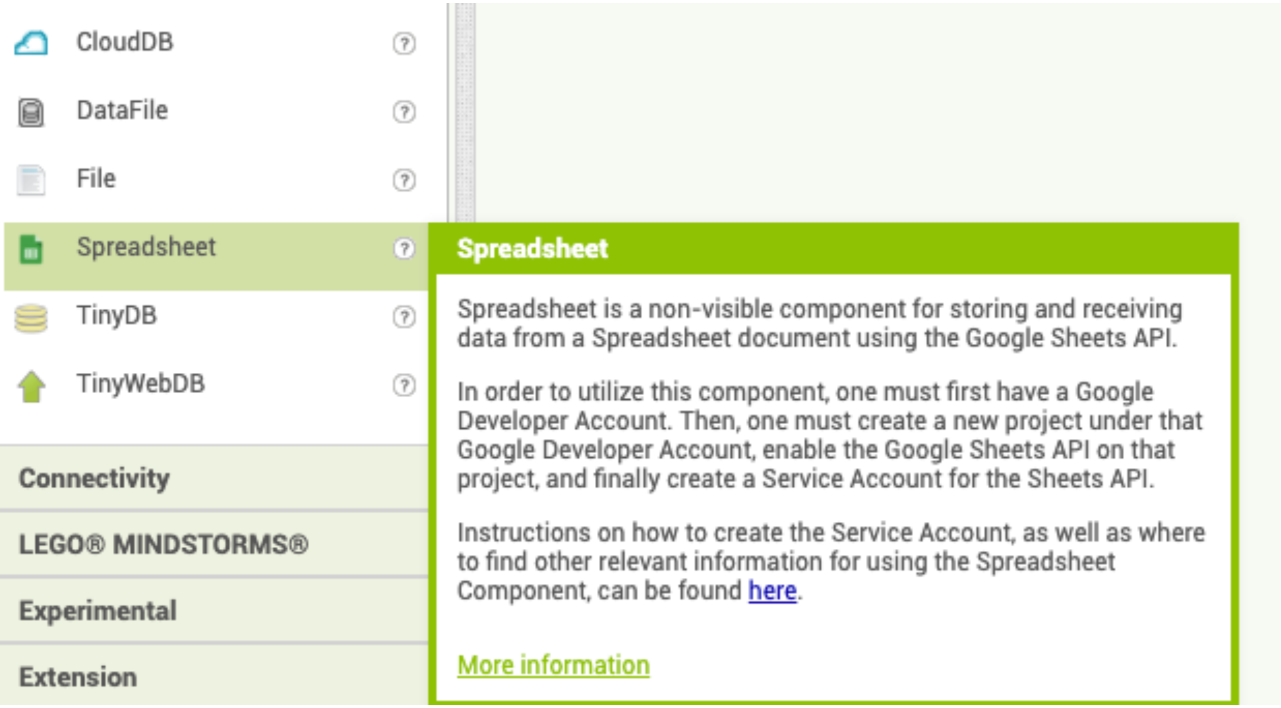
More information



File

File visible component for storing and retrieving files. Use this component to write or read files on your device. The default behaviour is to write files to the private data directory associated with your App. The Companion is special cased to write files to a public directory for debugging. Use the More information link to read more about how the File component uses paths and scopes to manage access to files.

More information



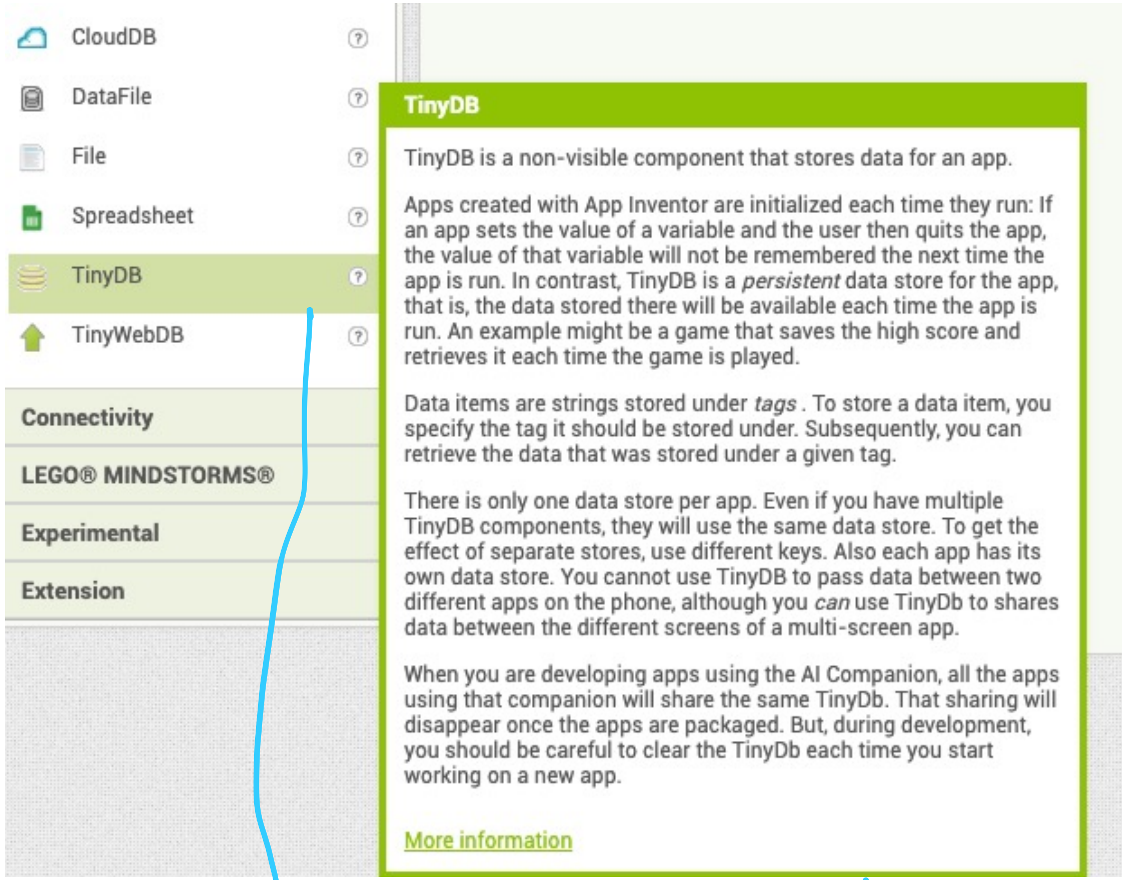
Spreadsheet

Spreadsheet is a non-visible component for storing and receiving data from a Spreadsheet document using the Google Sheets API.

In order to utilize this component, one must first have a Google Developer Account. Then, one must create a new project under that Google Developer Account, enable the Google Sheets API on that project, and finally create a Service Account for the Sheets API.

Instructions on how to create the Service Account, as well as where to find other relevant information for using the Spreadsheet Component, can be found here.

More information



TinyDB

TinyDB is a non-visible component that stores data for an app.

Apps created with App Inventor are initialized each time they run: If an app sets the value of a variable and the user then quits the app, the value of that variable will not be remembered the next time the app is run. In contrast, TinyDB is a *persistent* data store for the app, that is, the data stored there will be available each time the app is run. An example might be a game that saves the high score and retrieves it each time the game is played.

Data items are strings stored under *tags* . To store a data item, you specify the tag it should be stored under. Subsequently, you can retrieve the data that was stored under a given tag.

There is only one data store per app. Even if you have multiple TinyDB components, they will use the same data store. To get the effect of separate stores, use different keys. Also each app has its own data store. You cannot use TinyDB to pass data between two different apps on the phone, although you *can* use TinyDb to shares data between the different screens of a multi-screen app.

When you are developing apps using the AI Companion, all the apps using that companion will share the same TinyDb. That sharing will disappear once the apps are packaged. But, during development, you should be careful to clear the TinyDb each time you start working on a new app.

More information

Array = a[i]   index → num

a(0) a[1]

list = a[i]

dictionary → a['meyve']

a['ahmet']

std['name']

std['surname']

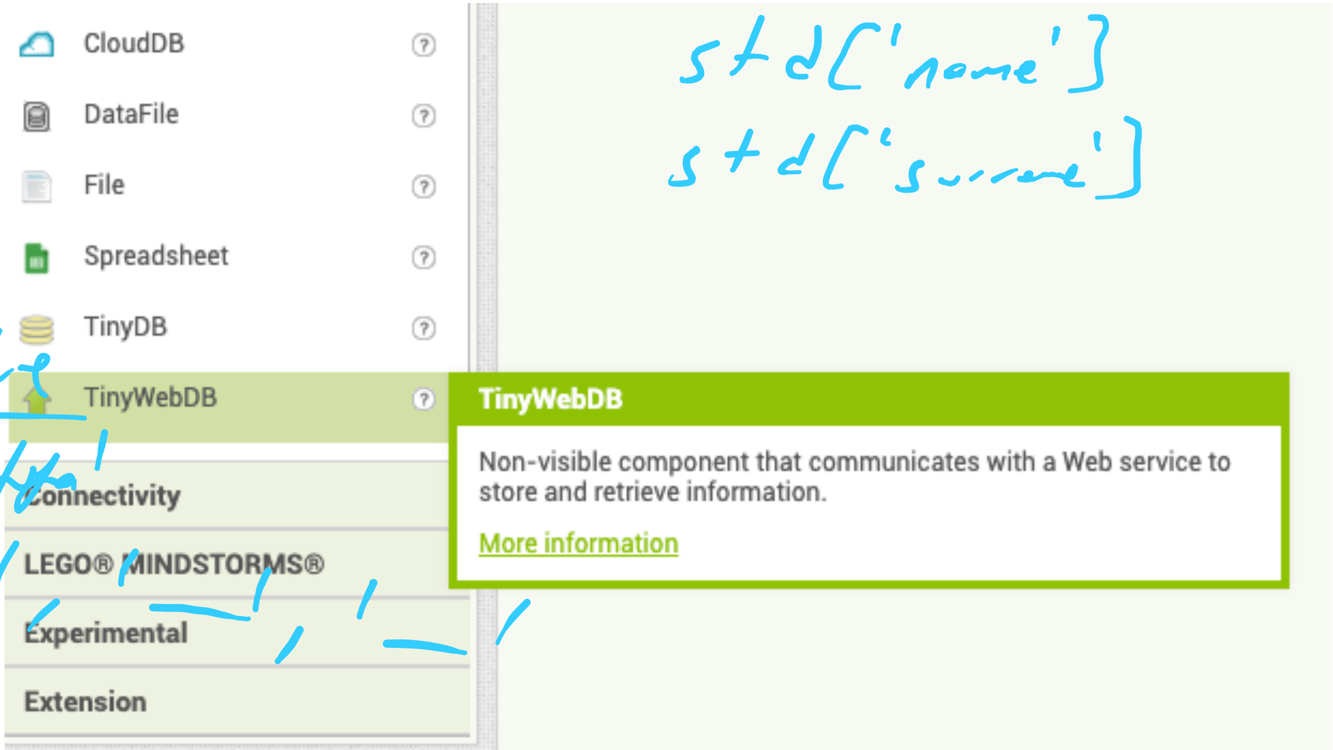
only one file

tag   value

ad   'mustafa'

tag   tasks   '—'   '—'   '—'

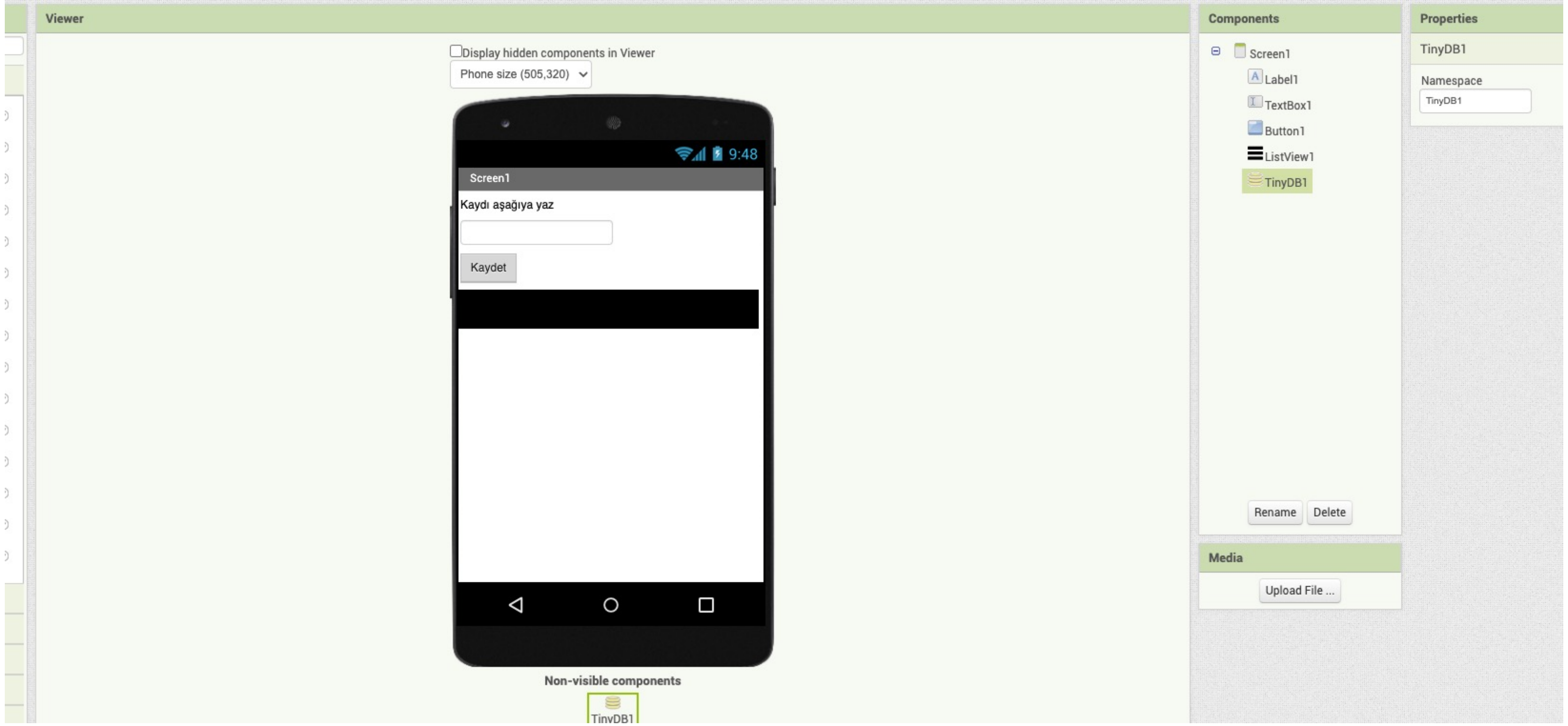
stored data → tag
value



TinyWebDB

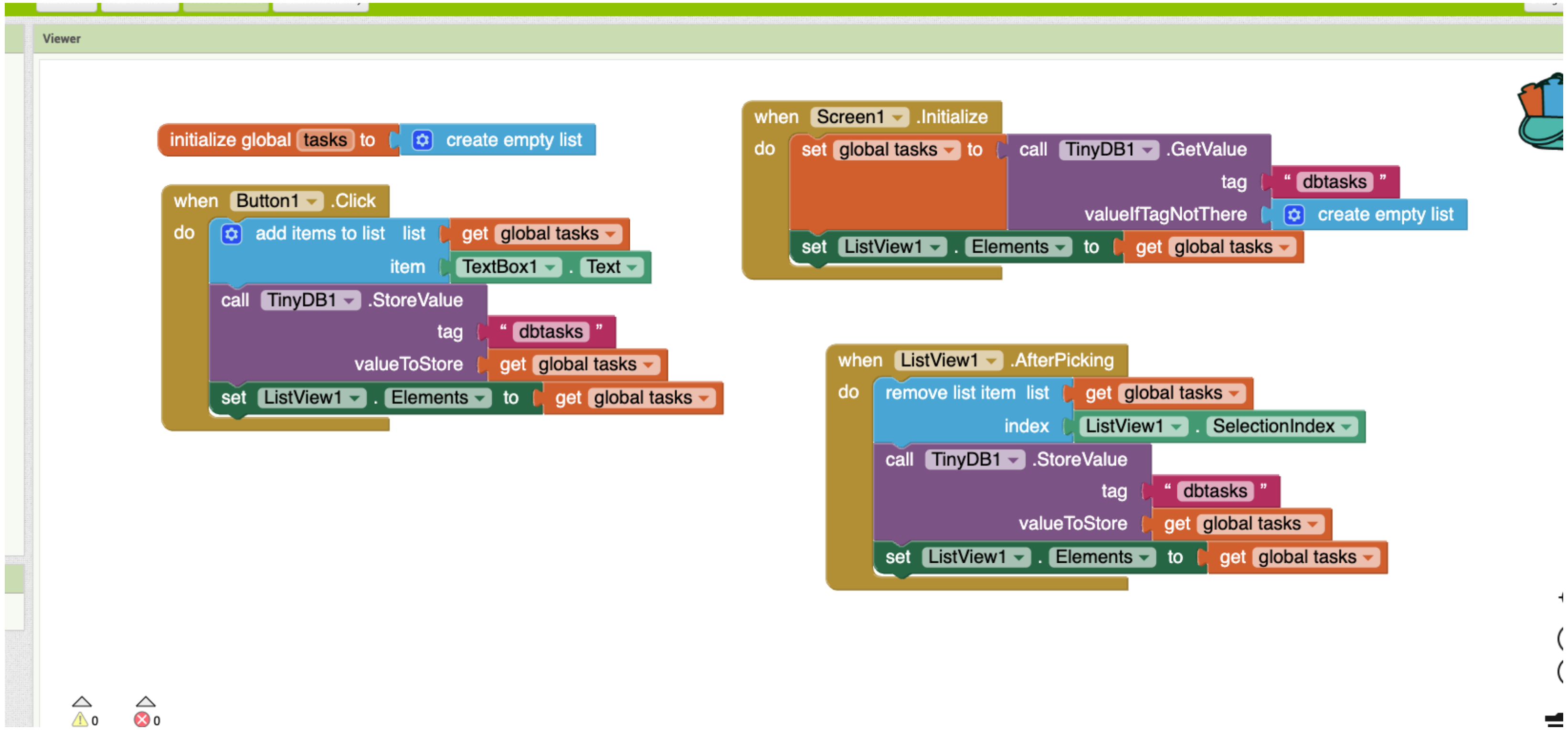
Non-visible component that communicates with a Web service to store and retrieve information.

More information



Namespace: TinyDB1

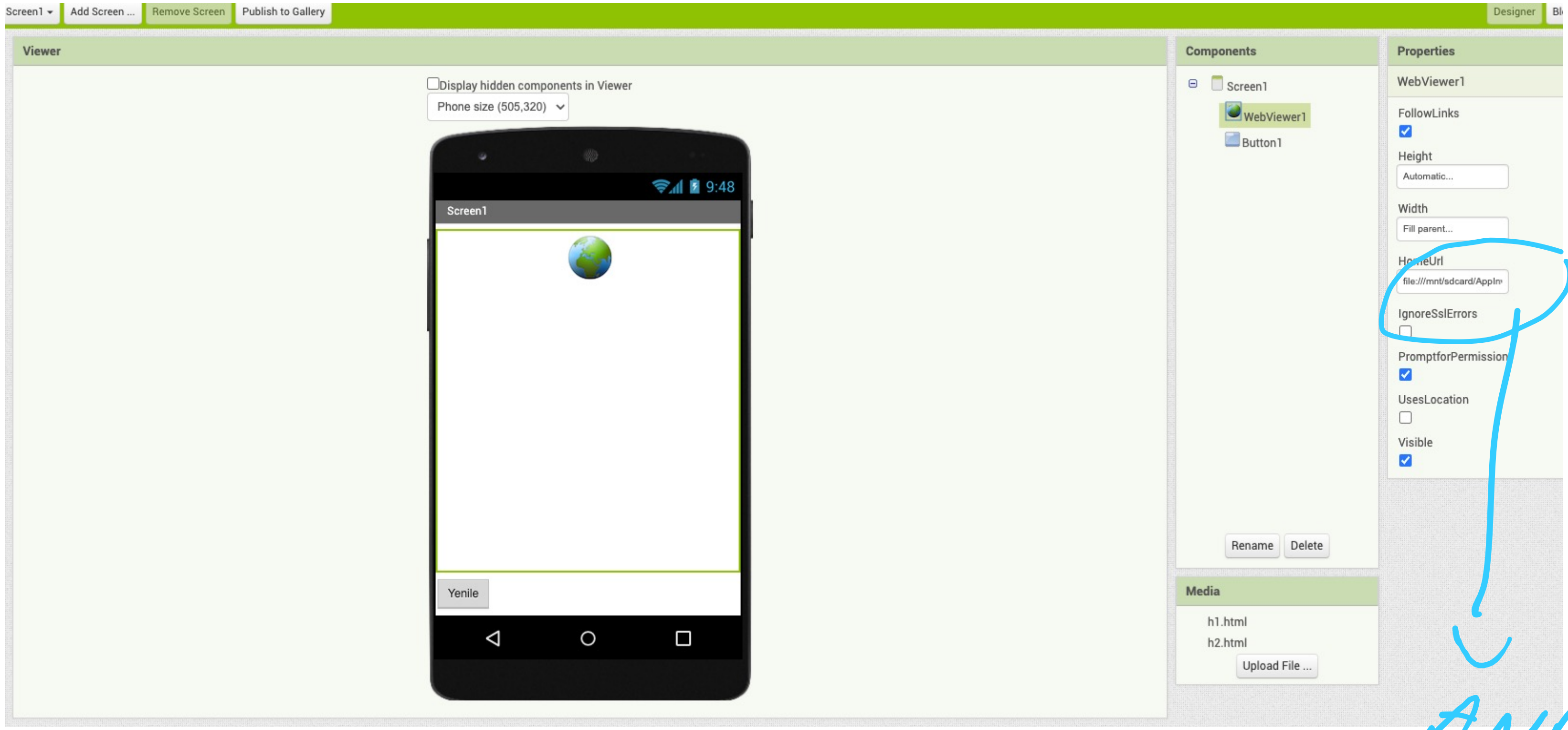
TINYDB EXAMPLE



```
initialize global tasks to   create empty list

when Button1 .Click
do   add items to list  list   get global tasks
                        item   TextBox1 . Text
     call TinyDB1 .StoreValue
                  tag   " dbtasks "
         valueToStore   get global tasks
     set ListView1 . Elements to   get global tasks

when Screen1 .Initialize
do   set global tasks to   call TinyDB1 .GetValue
                                      tag   " dbtasks "
                        valueIfTagNotThere   create empty list
     set ListView1 . Elements to   get global tasks

when ListView1 .AfterPicking
do   remove list item  list   get global tasks
                      index   ListView1 . SelectionIndex
     call TinyDB1 .StoreValue
                  tag   " dbtasks "
         valueToStore   get global tasks
     set ListView1 . Elements to   get global tasks
```

file:///mnt/sdcard/AppInventor/assets/h1.html

IOS



□ Display hidden components in Viewer

Screen1
- WebViewer1
- Button1

WebViewer1

FollowLinks ☑

Height: Automatic...

Width: Fill parent...

HomeUrl: file:///mnt/sdcard/AppIn

IgnoreSslErrors □

PromptforPermission ☑

UsesLocation □

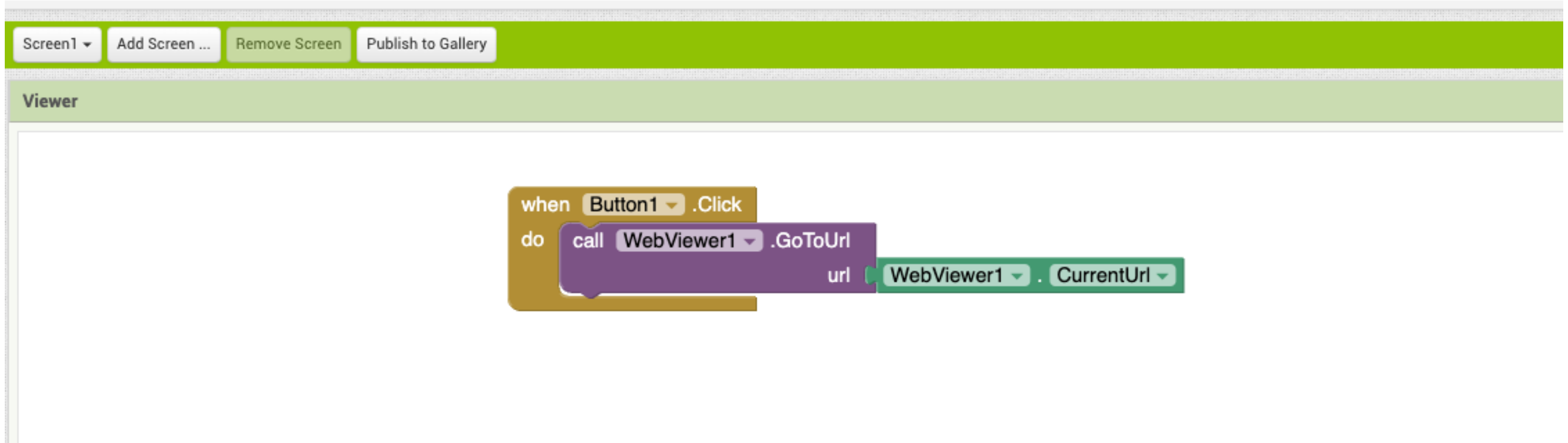
Visible ☑

h1.html
h2.html

ANDROID

WEBVIEWER AND HTML LOCAL FILE

http://localhost/h1.html



```
when Button1 .Click
do  call WebViewer1 .GoToUrl
                         url  WebViewer1 . CurrentUrl
```