

PRICE CHECKING ON AMAZON

Introduction

This tutorial will make a simple app that can retrieve prices of products from Amazon.com.

Design Editor Instructions

Drag a search box onto the screen.

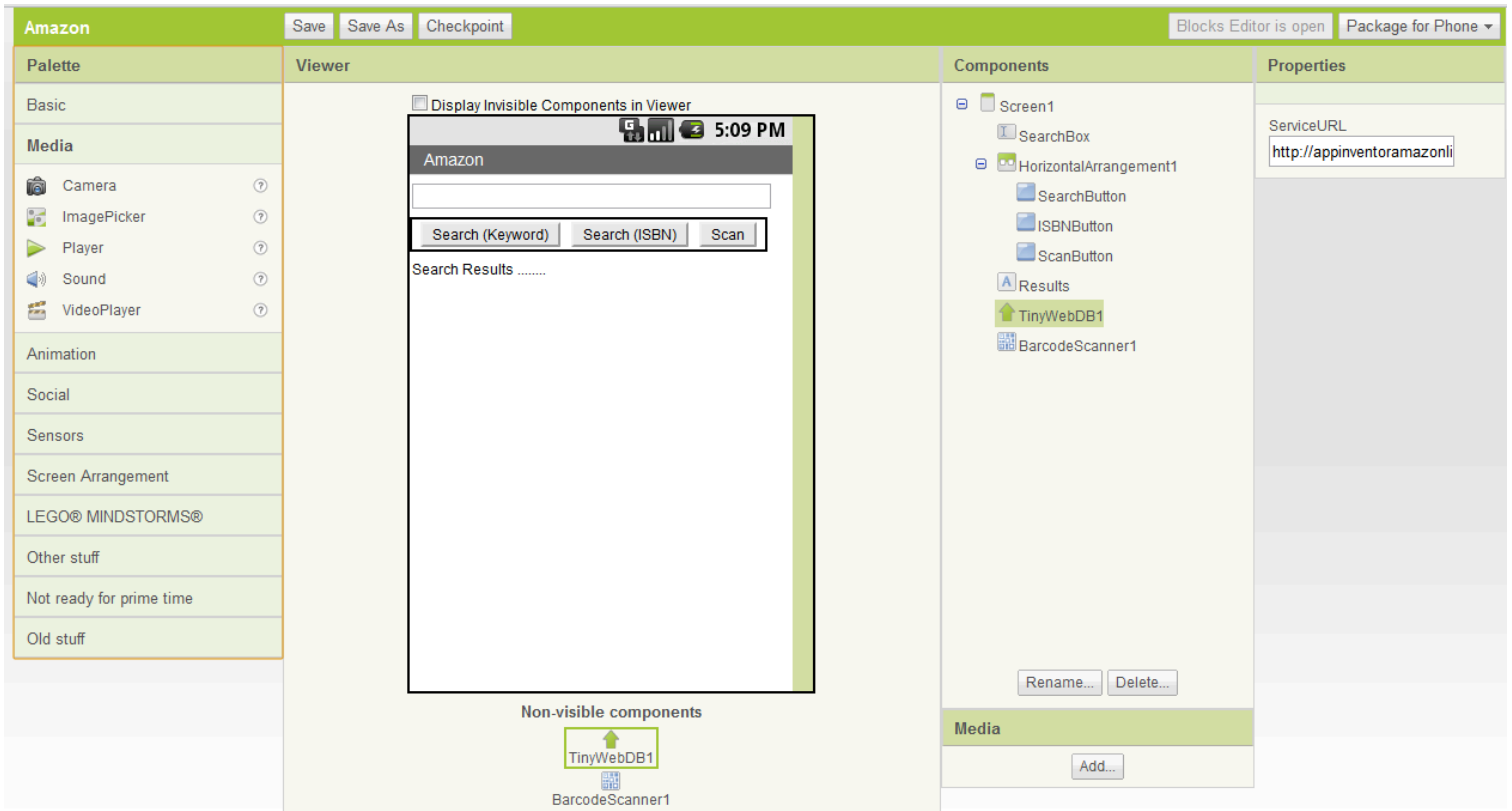
Pull out “HorizontalArrangement” component under “Screen Arrangement”. Inside that component, place three buttons. Change the names under components such as SearchButton, ISBNButton and ScanButton. Also Rename the text for the buttons in the properties section.

Insert a label at the bottom with text “Searching Results ...”. Rename it to “Results” in the components section.

Lastly, include the “BarcodeScanner” from “Other stuff” and TinyWebDB from “Not ready for prime time”.

Click on the TinyWebDB component. In its “Properties”, in the “ServiceURL” box, type <http://appinventoramazonlist.appspot.com> - This is a service that communicates with Amazon and retrieves the results of a search. When the user makes a search from our application, the TinyWebDB will ask the above the service to fetch the results for what product user wants. The service will get results and deposit them into the TinyWebDB. So, it is this service that is actually communicating with Amazon’s database.

Your screen should look like the following (the pictures includes the names of the components – these are the names that will be used to refer to these components later).



Note that in the application, the user will have three ways of searching for something on Amazon: he/she can search by keyword, ISBN number, or first scan the ISBN number of the item they want to search for.

Now we need to talk about how to use the scan button. The correct functionality of the “Scan” button is that when it is pressed, the camera should start automatically, and the user should be able to capture the barcode of some product. In order for this behavior to occur, the phone must have ZXing barcode scanner app installed. You can install it at:

<http://handheld.softpedia.com/progDownload/ZXing-Barcode-Scanner-Android-Download-89620.html>

Blocks Editor Instructions:

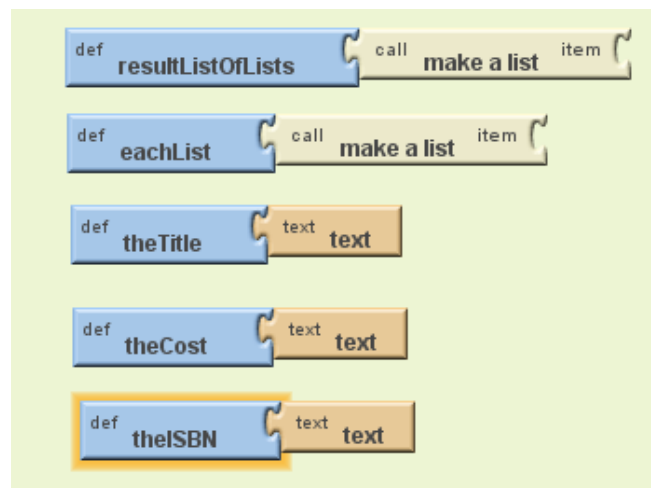
First of all, the data in TinyWebDB – what kind of format is it in? It is in the format of a list of lists. And the content of the lists will be of type String. For example, if you searched for “don quixote”, you would get the following back:

- ["VALUE", "don quixote", [{"\"Don Quixote\"\", \"\$6.91\", \"0060934344\"}, {\"\"Don Quixote (Penguin Classics)\"\", \"\$7.29\", \"0142437239\"}], [\"DON

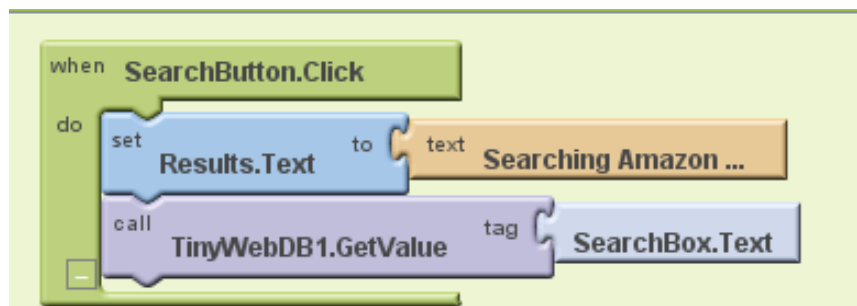
```
QUIXOTE Complete\\"", 0.0, "B003PPDH52"], [\"Don Quixote de La Mancha (Volume 1)\\"", \"$12.16\", \"0217828264\"], [\"Don Quixote (Penguin Classics)\\"", \"$8.64\", \"0140449094\"]]]
```

At index 2, we have our list of lists. Inside the big list, there will be 5 sub lists – every sub list containing three things – the specific title, the cost, and its ISBN number.

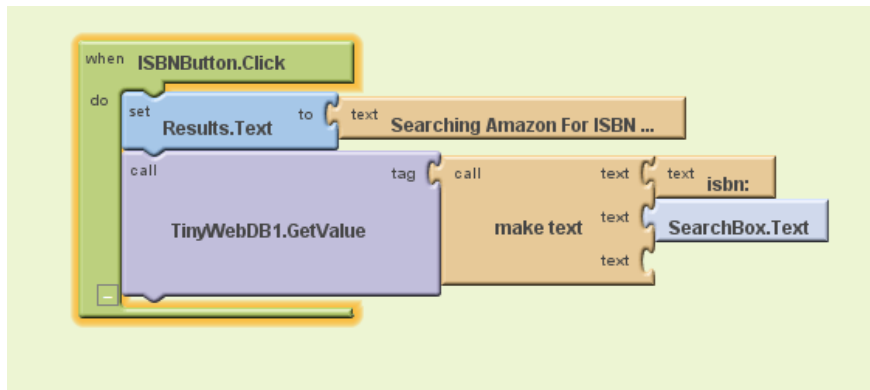
First create 5 variables inside the block editor like below. The first variable will hold the list of lists from the search. We will be iterating through this list of lists – during that iteration, the second variable will hold individual lists inside the big list. The other three variables will be assigned to the title, the cost, and the ISBN number of each list.



When a user presses the search button, we will set our “Results” label to say “Searching Amazon ...”. Then we will make a method call on TinyWebDB1 – we will call “GetValue” with the text that user typed in the search box. This will populate the TinyWebDB1 with the big list.

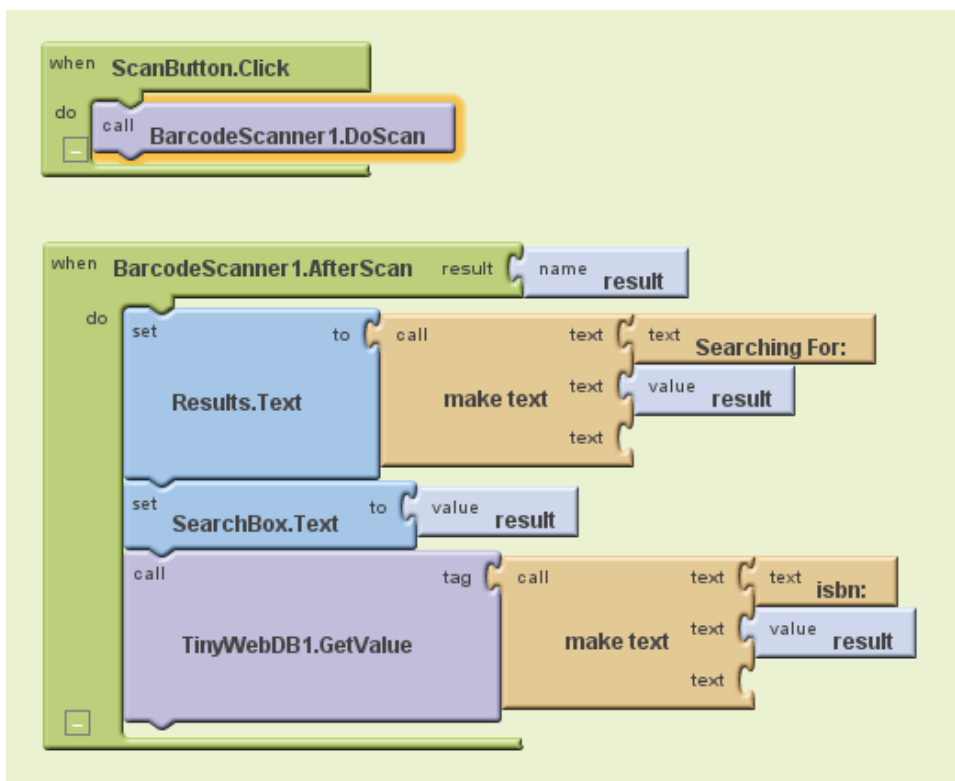


If the user clicks on the ISBN button instead,



then we assume that the user typed in some ISBN number. So in the search that we send to TinyWebDB1, we add the string “isbn:” to what user typed.

If user presses the “Scan” button, we do the following:

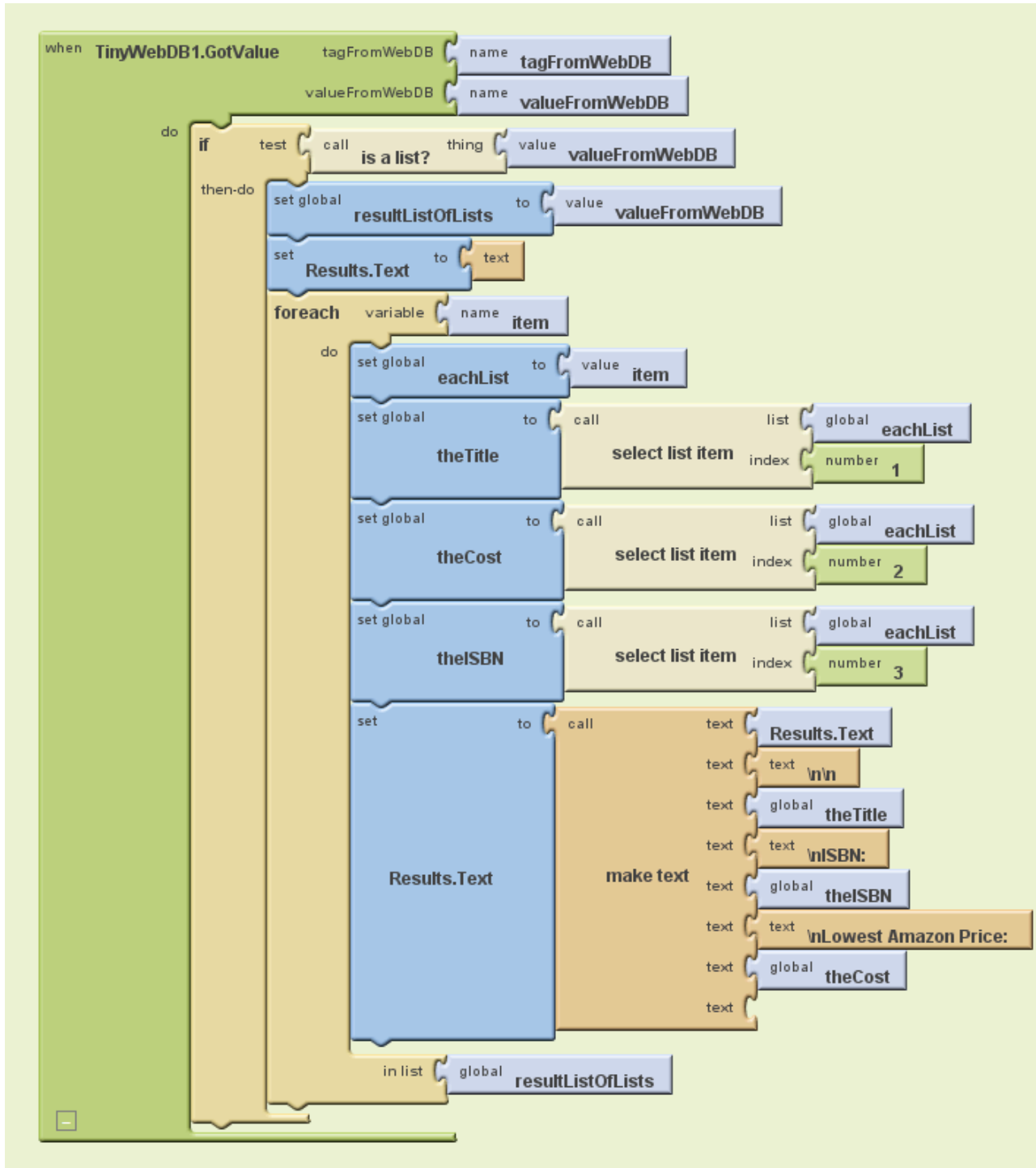


First we call the “DoScan” method of BarcodeScanner1 – that should launch the camera.

Then when the user is done scanning the barcode (i.e. done taking the picture of the barcode), the resulting ISBN number is available in “result” parameter. We

set the values of the “Results” label and the text of “SearchBox” to be the number inside “result” parameter. Again, we make a search to TinyWebDB1 by appending “isbn:” to the font.

Now, what is left to do is iterate through our results and display them on the screen:



What's happening above is that when we the TinyWebDB1 first receives the value from the search, "TinyDB1.GetValue" event will be triggered. When that event happens, we will receive "valueFromDB" as our parameter. That parameter holds the result.

First we check if it is a list or not. Then we assign it to our variable "resultListOfLists".

Initially, what we want to display starts with an empty string.

Then, in a "for each" loop that loops through "resultListOfLists", we set variable "eachList" to be the next list in "resultListOfLists". Then from "eachList", we retrieve the title, the cost, and the ISBN number from indices 1, 2, and 3 respectively.

Now we format this content a little and then add it to our "Results" label.

Now the "Results" label will contain information that was in all 5 lists, and this information will be available for user to see on the screen.

This is what the final Blocks Editor Diagram looks like:

```

def resultListOfLists call make a list item
def theTitle text text
def theISBN text text
def eachList call make a list item
def theCost text text

```

```

when TinyWebDB1.GetValue tagFromWebDB name tagFromWebDB
valueFromWebDB name valueFromWebDB
do
  if test call is a list? thing value valueFromWebDB
  then-do
    set global resultListOfLists to value valueFromWebDB
    set Results.Text to text
    foreach variable name item
    do
      set global eachList to value item
      set global theTitle to call select list item list global eachList index number 1
      set global theCost to call select list item list global eachList index number 2
      set global theISBN to call select list item list global eachList index number 3
      set Results.Text to call make text text Results.Text
      text text '\n\n'
      text global theTitle
      text text '\nISBN:'
      text global theISBN
      text text '\nLowest Amazon Price:'
      text global theCost
    in list global resultListOfLists
  end
end

```

```

when SearchButton.Click
do
  set Results.Text to text Searching Amazon ...
  call TinyWebDB1.GetValue tag SearchBox.Text
end

```

```

when ISBNButton.Click
do
  set Results.Text to text Searching Amazon For ISBN ...
  call TinyWebDB1.GetValue tag call make text text isbn: SearchBox.Text
end

```

```

when ScanButton.Click
do
  call BarcodeScanner1.DoScan
end

```

```

when BarcodeScanner1.AfterScan result name result
do
  set Results.Text to call make text text Searching For: result
  set SearchBox.Text to value result
  call TinyWebDB1.GetValue tag call make text text isbn: result
end

```