

Using the Network

Contact information

Dr Martin Bateman
Room CM116
C&T Building
mbateman@uclan.ac.uk
<http://www.star.uclan.ac.uk/~mb/>
@martinbateman use #collll



2

Networks

What is a network?

3

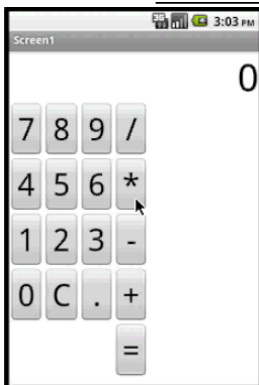
Communicating with App Inventor

- App Inventor gives us many methods for communicating for example
 - Web Service
 - tinywebdb
 - twitter
 - bluetooth
 - texting
 - email
- Each is provided by a different Component

Concentrate here for today

4

What we are going to build...

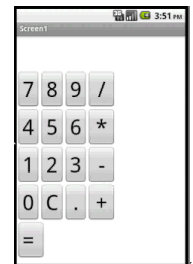
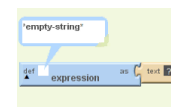
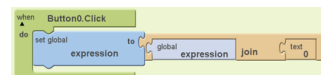


- Simple calculator
- We'll use networked services where we can

5

Building the expression

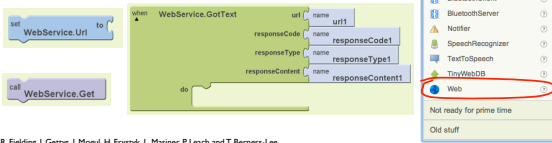
- Use a variable to store the expression
- An event handler per Button (0 to 9, +, -, /, * and .) which appends the symbol to expression
- When ButtonEquals.Click
 - send expression to Web Service



6

Web Service

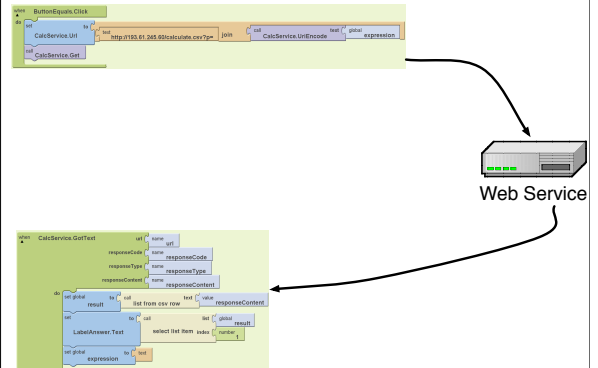
- None visible component
- Allows us to talk to a Web Service
- Web Services allow us to extend functionality using remote computers
 - they use HTTP[*] to communicate
- Three blocks are of interest



[*] R. Fielding, J. Gettys, J. Mogil, H. Frystyk, L. Masner, P. Leach and T. Berners-Lee, "Hypertext Transfer Protocol – HTTP/1.1", RFC 2616, June 1999

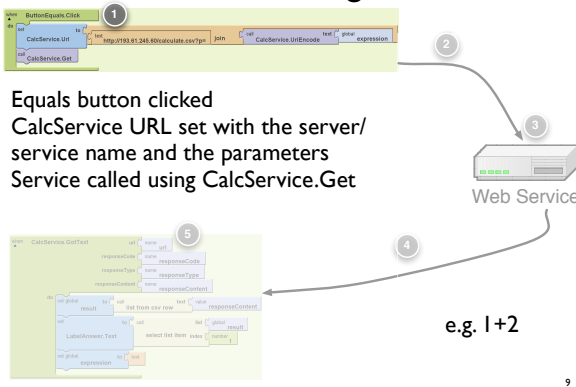
7

Using a Web Service



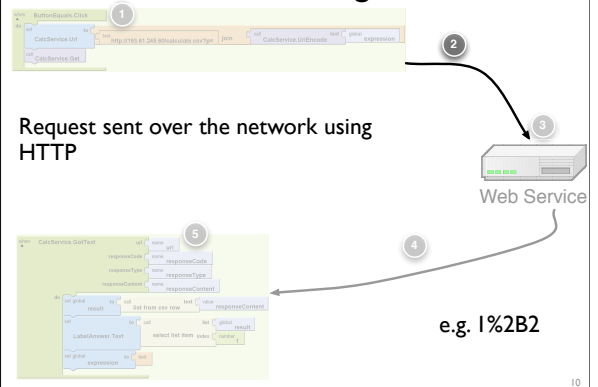
8

Using a Web Service



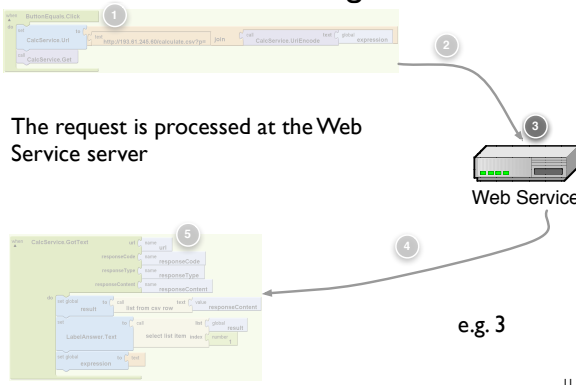
9

Using a Web Service



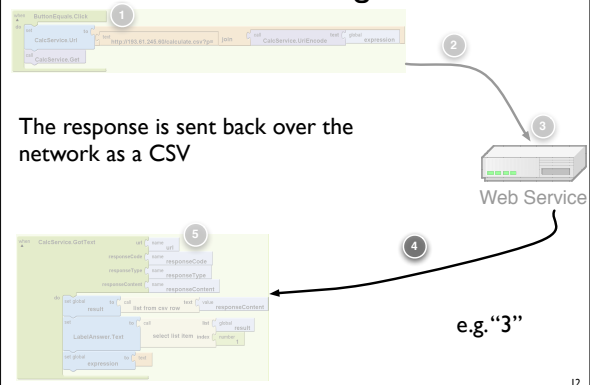
10

Using a Web Service



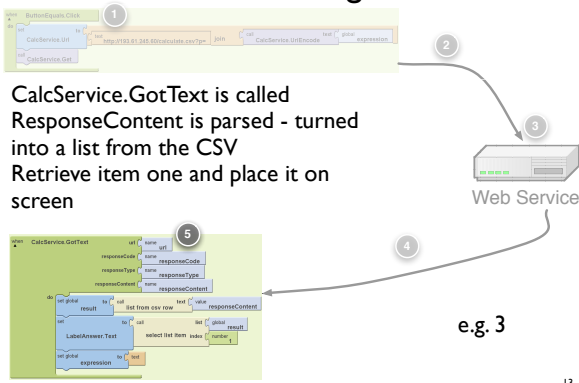
11

Using a Web Service



12

Using a Web Service



13

Passing parameters to Web Services

- We pass the parameters to our Web Service as part of the URI such as

`http://193.61.245.60:80/calculate.csv?parameter=value`

[*]T. Berners-Lee, R. Fielding and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", RFC 3986, January 2005

14

Passing parameters to Web Services

- We pass the parameters to our Web Service as part of the URI such as

`http://193.61.245.60:80/calculate.csv?parameter=value`

- is
 - the protocol we are using to communicate
 - in this case the *Hyper Text Transfer Protocol* - the same protocol as the World Wide Web

[*]T. Berners-Lee, R. Fielding and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", RFC 3986, January 2005

15

Passing parameters to Web Services

- We pass the parameters to our Web Service as part of the URI such as

`http://193.61.245.60:80/calculate.csv?parameter=value`

- is
 - the computer we are talking to
 - normally we'd use a *hostname* here I'm using an IP address

Why is this a bad thing?

[*]T. Berners-Lee, R. Fielding and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", RFC 3986, January 2005

16

Passing parameters to Web Services

- We pass the parameters to our Web Service as part of the URI such as

`http://193.61.245.60:80/calculate.csv?parameter=value`

- is
 - the port on the computer - used to identify a piece of software
 - HTTP by default uses port 80 so we normally omit it

[*]T. Berners-Lee, R. Fielding and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", RFC 3986, January 2005

17

Passing parameters to Web Services

- We pass the parameters to our Web Service as part of the URI such as

`http://193.61.245.60:80/calculate.csv?parameter=value`

- is
 - the name of the Web Service
 - here calculate.csv for a calculator service
 - the file extension (csv) is an artefact of our Web Service hosting environment

[*]T. Berners-Lee, R. Fielding and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", RFC 3986, January 2005

18

Passing parameters to Web Services

- We pass the parameters to our Web Service as part of the URI such as

`http://193.61.245.60:80/calculate.csv?parameter=value`

- is
 - a list of parameters to pass to the Web Service - this is similar to the parameters passed to a procedure
 - the question mark (?) shows the start of the parameter list
 - value is passed to the Web Service with the name parameter
 - use ampersand (&) to separate parameters, e.g.
 - `[...]?parameter1=value1¶meter2=value2[...]`

[*]T. Berners-Lee, R. Fielding and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", RFC 3986, January 2005

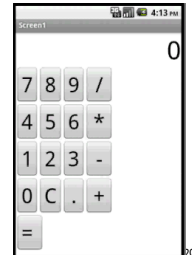
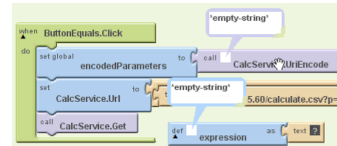
19

Passing Parameters Problem

- You cannot pass some characters in this way because they have special meanings e.g.

`http://193.61.245.60:80/calculate.csv?p=1+2`

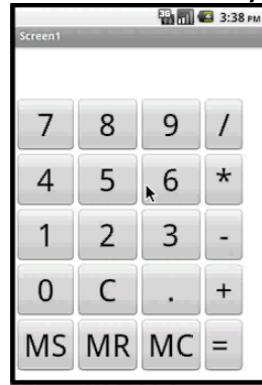
- fails because + has a special meaning it is used as a space
- + instead is represented by %2B
- called percent encoding



20

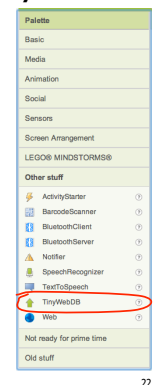
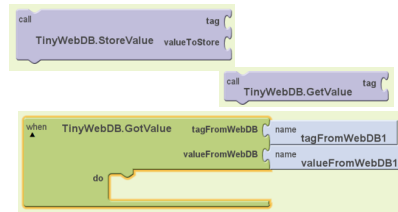
And now for some memory

- Calculators have memory
 - single value memory
- Let's use a network accessible database called TinyWebDB
- Extends the TinyDB idea
- Database is hosted in the network rather than on the phone



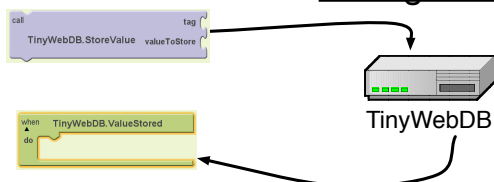
TinyWebDB

- None visible component
- Talks to a specific Web Service
 - which implements a simple database
- You must set the ServiceURL property
 - in the GUI builder
- Three blocks are of interest



22

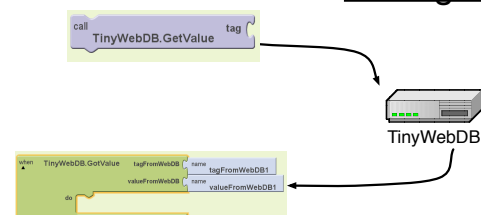
Storing a value



- The value valueToStore is stored in the TinyWebDB at the key tag
- Don't need the TinyWebDB.ValueStored

Do you think TinyWebDB is a Web Service - why?

Getting a value



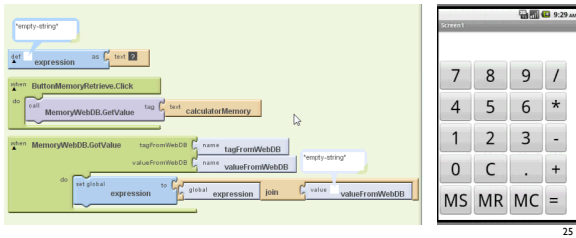
- Same pattern as the Web Service request
- Different from a Procedure call

Why do you think the Web Service call doesn't follow the same pattern as the Procedure Call?

24

TinyWebDB

- In our calculator here's how it works



This week's challenge

- Build a quiz application
 - you've done one with a local list of questions
 - now use remote services for
 - questions
 - storing scores
- Use the Quiz Web Service (see today's Lab sheet for details)
- After Brendan's lectures
 - designed and polished feel to the user interface

26

Questions?

27