Server Export Function

Server

The server has been updated to export a list of UID's. You can program your application to retrieve this list, and then arrange the content as you see fit.

Note: The implementation of this feature is in no way recommended practice. This feature has been implemented in a way which would be considered bad design from an access point interface (API) perspective.

Instructions

The server will now accept an export command, that will provide a list of the user identification details (UID) which have checked in to the lecture.

The command can be run with the following code:

https://engg1500.newcastle.edu.au/ENGG1500DemoServer/API?command=export&group=GROUPN

where N is the number associated with your group.

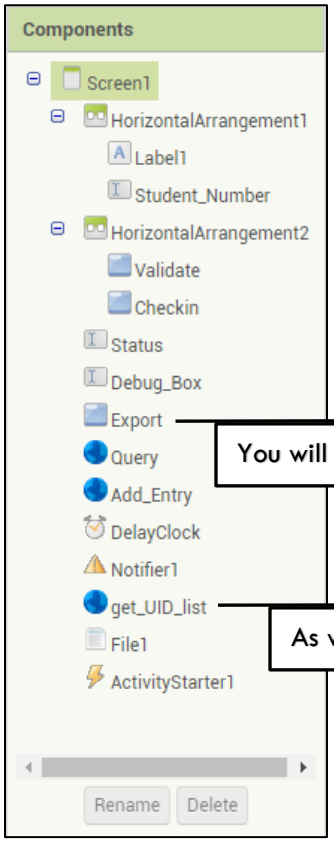
This will return the details:

```
{
   "apiResponse": {
      "code": "SUCCESS"
      "response": "[c8475738, c8474738, c17274849, c18237365]"
   }
}
```

Where a list of UIDs will be imbedded into the response code.

Solution Code

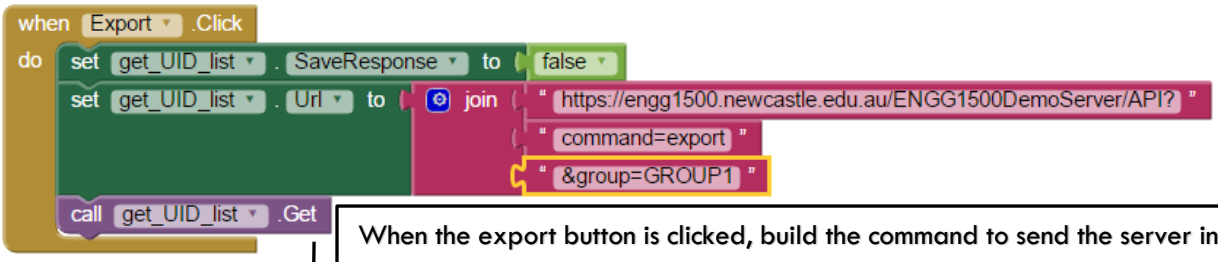
Designer



You will need to add a new button "Export" to the screen

As well as a new web component to run the communication with the server

Blocks



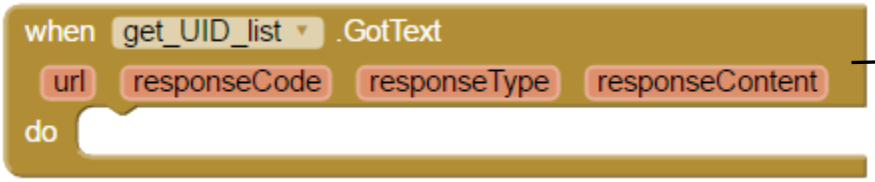
When the export button is clicked, build the command to send the server in the usual way. Then send the request to the web using the .Get method. Ensure that the web components SaveResponse property is set to false, so that you can trigger the web components GotText event.

Remember, after sending this command, the responseContent from this server will look something like this:

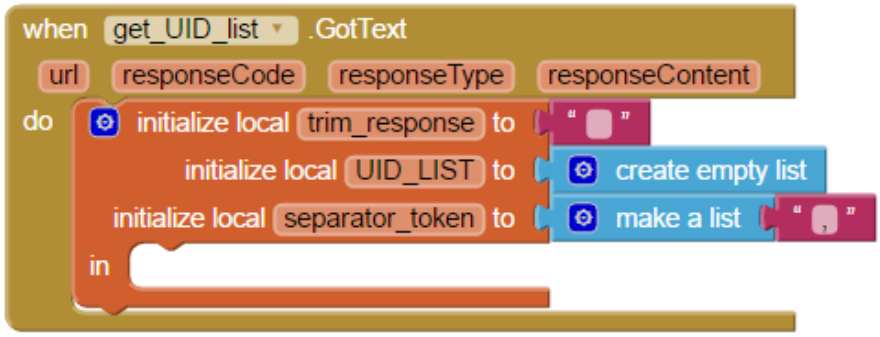
```
{
   "apiResponse": {
      "code": "SUCCESS"
      "response": "[c8475738, c8474738, c17274849, c18237365]"
   }
}
```

The aim is to cut up this response, extract the UIDs, and store them somewhere so that they can be used/saved.

There is a few things going on here, so lets break it up.



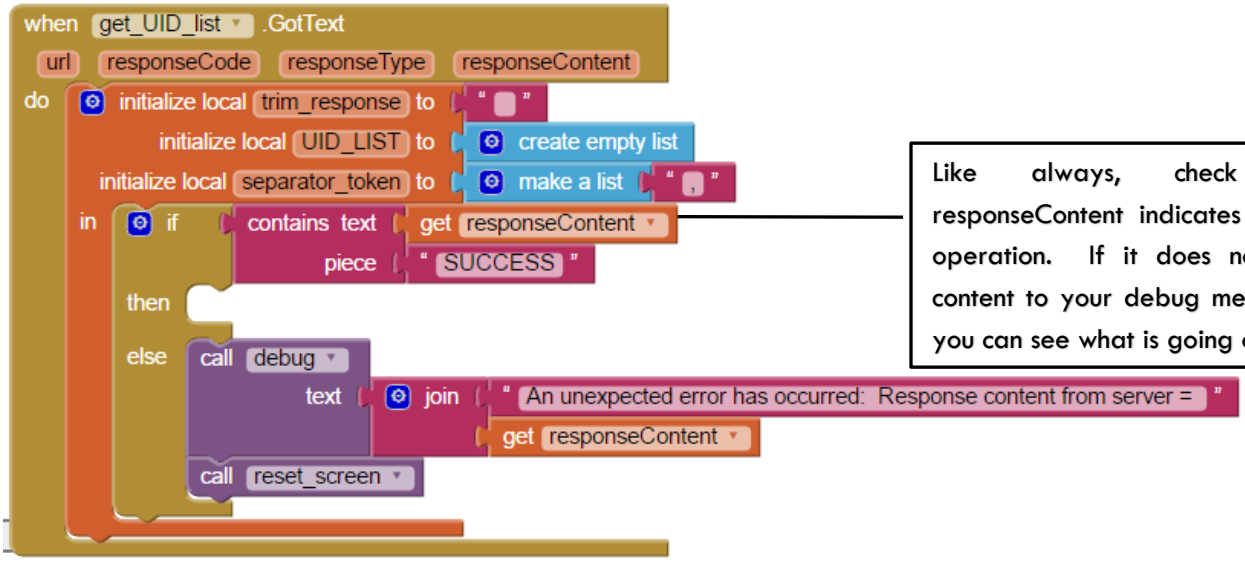
Using the GotText event from the web component.



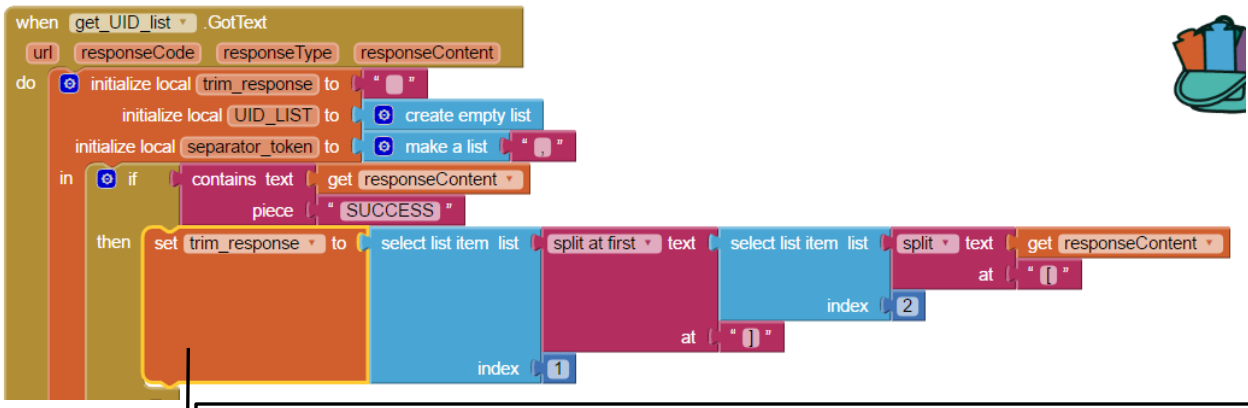
Now let's initialise some local variables. You will want a string which will hold the trimmed response: that is, the segment of the response between the square brackets. For example: `[c8475738, c8474738,c17274849,c18237365]`

You will also want a list which will hold each individuation UID in a list format.

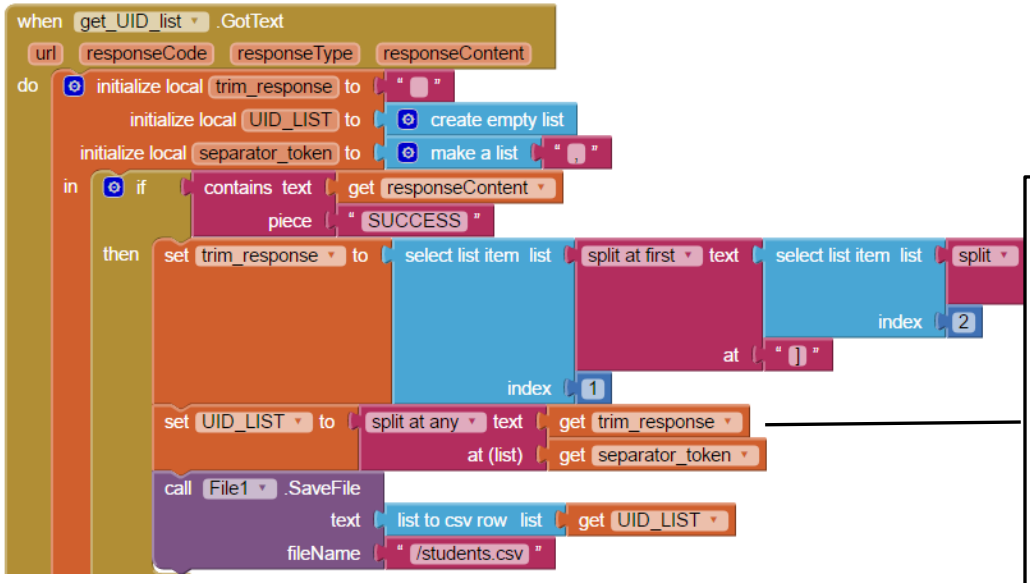
Finally, so that we can split the string into the UIDs, you will also need another list, which defines the separator token.



Like always, check if the responseContent indicates a successful operation. If it does not, send the content to your debug method, so that you can see what is going on.



If the responseContent indicates a successful operation, then it will need to be trimmed down to the pertinent content. This code will allow the variable trim_response to hold the details between the first open square bracket, and the final close square bracket.



Once the response has been trimmed, you can extract the UIDs by splitting the response string using list which contains the defined separator token.

Once the list is created, the application can work with the UID list in any desired form.