

## Mobile Computing

### Lab 6

## 1 Web Services

Today's lab is focused on creating a web service that will allow your app to get data from a remote database. Web Services use standard data interchange formats to send data to the app, and we will be using the JSON format for this purpose. Alternatives include SOAP XML. For an example of how JSON works, look at <http://www.elated.com/articles/json-basics>. We will be using a web server that is hosted at `lamp.ms.wits.ac.za`. You may have some experience of this server from the DBF course, but I have outlined the steps to connect in case you have not used it before.

### 1.1 Logging in

- In linux, open a terminal and log in to the server by typing (with some modification):

```
ssh username@lamp.ms.wits.ac.za
```

- Your username is your student number preceded by the letter s, and your password is the same. For example, if your student number is 9700884, your login will be s9700884.
- Once you have logged in, you should change your password by typing

```
passwd
```

- Next, create a folder called `public_html` in your home directory by typing the following:

```
mkdir public_html
```

### 1.2 Mysql

- You might already know this section because you are already using this account in DBF.
- Your username for mysql is your student number preceded by an s.
- In the terminal, you can connect to mysql by typing a command of the form:

```
mysql -uUsername -p
```

eg.

```
mysql -us9700884 -p
```

- In mysql, you have a database created for you. Its name is your student number preceded by a d.

- You can switch to this database by typing:

```
\uDatabase
```

eg.

```
\ud9700884
```

- In this database, create a table called CARS, which has the following parameters:
  1. CAR\_ID - primary key, auto increment
  2. OWNER - stores the name of the owner
  3. AGE - stores how old the car is
  4. BRAND - stores which make of car it is
  5. NUMBER - stores the license plate number of the car
- Add three cars to this database. Use any data you like.
- Quit mysql by typing exit.

### 1.3 Web

- Next, we must create the server side application that will read the data from the database and provide it to the android app.
- Go to the `public_html` folder that you created on the server.
- Edit a new file called `cars.php` by typing the following:

```
nano cars.php
```

- Paste in the following code. Change the username and password variables so that they match your mysql account.

```
<?php
$username = "s9700884";
$password = "s9700884";
$database = "d9700884";
$link = mysqli_connect("127.0.0.1", $username, $password, $database);
$output=array();
/* Select queries return a resultset */
if ($result = mysqli_query($link, "SELECT * from CARS")) {
    while ($row=$result->fetch_assoc()) {
        $output[]=$row;
    }
}
mysqli_close($link);
echo json_encode($output);
?>
```

- You can save and exit by hitting Ctrl-X
- Note that in PHP, all variable names begin with a \$, and variables do not have to be declared.
- Next, test that this is working by going to  
<https://lamp.ms.wits.ac.za/home/username/cars.php>  
eg.  
<https://lamp.ms.wits.ac.za/home/s9700884/cars.php>

## 1.4 Android

- Create an android app that connects to the server, getting the json data from the server and displaying it on the screen.
- Submit this app to the marking system.

## 1.5 Request variables

- In your previous lab, you looked at how to send request variables as name value pairs from android. Your current PHP code doesnt respond to these variables.
- In PHP, the request variables are stored in a variable called \$\_REQUEST.
- A PHP script which responds to a request variable called brand is shown below. You can save this file as cars2.php

---

```
<?php
$username = "s9700884";
$password = "s9700884";
$databse = "d9700884";
$link = mysqli_connect("127.0.0.1", $username, $password, $databse);

$brand = $_REQUEST["brand"];
$output=array();
/* Select queries return a resultset */
if ($r = mysqli_query($link, "SELECT * from cars where
    brand='$brand'")) {
    while ($row=$r->fetch_assoc()){
        $output[]=$row;
    }
}
mysqli_close($link);
echo json_encode($output);
?>
```

---

- You can test this code by going to  
<https://lamp.ms.wits.ac.za/home/username/cars2.php?brand=Toyota>  
eg.  
<https://lamp.ms.wits.ac.za/home/s9700884/cars2.php?brand=Toyota>

- Note that in a production system, you should use a prepared statement to bind the brand variable to the query rather than putting it in the string directly. This prevents hackers from using SQL injection, and is more efficient. I will put up a video illustrating how SQL injection works and how a hacker can exploit this vulnerability.
- Hand in your PHP file for this part of the code on moodle.