Database access

There are several ways to access the database from a Ushahidi 2.x controller:

- Calling functions in Kohana models - this is preferred (we're working in an Model-View-Controller framework)
- Calling the Kohana ORM - occasionally (e.g. for simple queries or a rare database access across multiple database tables). ORM calls are also preferred in the model functions called above.
- Using mysql code - bad and naughty, but sometimes necessary in Kohana 2.3.1, which doesn't handle things like joins and groups well (Kohana 3 is better).

Model calls

Examples of accessing database tables using their models:

In the reports controller:

```
$total_reports = Incident_Model::get_total_reports(TRUE);
```

In the incident model:

```
public static function get_total_reports($approved = FALSE)
{
return ($approved)
? ORM::factory('incident')->where('incident_active', '1')->count_all()
: ORM::factory('incident')->count_all();
}
```

Note that the incident model uses an ORM call to access the incident table in the database.

ORM calls

If you look at the Ushahidi v2.x code, you'll see a lot of ORM calls. For instance:

To return a row from the table when you know the row id:

```
$report = ORM::factory('incident', $category_id);
```

To return all rows from a table:

```
$reports = ORM::factory('incident')->find_all();
```

To return the number of rows from a table:

```
$reportcount = ORM::factory('incident')->count_all();
```

These are made more useful by adding filters, ordering etc to the ORM calls, e.g.

```
$reports = ORM::factory('incident')

->where('incident_verified', '1')

->where('incident_active', '1')

->find_all();

$reports = ORM::factory('incident')

->orderby(array('incident_date'=>'ASC'))

->find_all();
```

The ORM library is powerful and has many more things it can do - see the Kohana ORM Library pages for more information on these.

MySQL Calls

If you have to do this, here's how:

- create a database object to call:

```
$db = new Database();
```

- Create your convoluted sql statement:
- Call the database with your sql statement:

```
$db->query($sql);
```