Ushahidi 3.x, Conventions & Coding Style

PHP Coding Standards

Ushahidi consists of two parts: a core application in the `src/` directory and a delivery layer in the `application/ directory`. The delivery layer is currently built using the Kohana Framework and has slightly different coding standards than the rest of the application. The Kohana coding standards are documented in the Kohana user guide. The rest of the application uses PSR-2 code style, but uses tabs instead of spaces to prevent mixing tabs and spaces across the application.

Class Names and File Location

Class names in the core application are autoloaded using PSR-4 and all exist with within `Ushahidi` namespaces, and the files are all located within the `src/` directory.

Class names in Kohana follow a strict convention to facilitate autoloading. Class names should have uppercase first letters with underscores to separate words. Underscores are significant as they directly reflect the file location in the filesystem. Kohana does not use namespaces, and all delivery files are located within the the `application/` and `modules/` directories.

The following conventions apply:

1. CamelCased class names should be used when it is undesirable to create a new directory level.
2. All class file names and directory names must match the case of the class as per PSR-4.
3. All classes should be in the `classes` directory. This may be at any level in the cascading filesystem.

Basic Documentation Commenting

We use phpDocumentor, a tool for creating documentation directly from both PHP and external documentation, to keep track of all our code documentation. The documentation within the code is done using PHPDoc which is an adaptation of Javadoc for the PHP programming language.

Documenting code makes it easier to understand and lowers the barrier to entry for developers wishing to contribute to the codebase.

The first thing to take note of is that PHPDoc comments must be enclosed in DocBlocks. A DocBlock is a C-style comment that begins with a `/**` and with a leading asterisk `*` on each line. Any line within a DocBlock that doesn't begin with a `*` will be ignored. Example:

```
/**
 * Example use of DocBlocks in PHP
 */
```

Secondly, DocBlocks must precede the code you are adding comments to. For example, if you wanted to document the function `foo()`, you would proceed as follows:

```
/**
 * DocBlock comment for function "foo()"
 */
function foo()
{
}
```

Javascript

Javascript should follow Crockford code style with a few exceptions. Code should pass a style check using JSCS.

All JS should also pass in JSHint (with our .jshintrc).