Testing your own project

The GemsTracker library will try to implement more and more testing to help provide more stable code and easier refactoring. To implement testing at project level, you will need some extra's on top of the normal GemsTracker installation.

Requirements

A working PHPUnit, for example installed via PEAR. Additional GemsTracker library files from the test folder. Please make sure that library and test files have the same version number.

Setup

Add a folder `\test` to your project root and copy the additional files you downloaded from the test folder. Your project root will look something like this:

```
\application
\htdocs
\library
\test
\var
```

And more specific, the test folder will look like this where the library could be configured as external:

```
\test\classes\ProjectName
\test\library\Gems
\test\library\Zend
\test\bootstrap.php
```

To be able to run the tests, you will need to tell PHPunit to use a bootstrap file that will setup the include path and register autoloaders. A generic helper file is present in the `\test\library` folder by the name `project-bootstrap.php`. For default setups this will be enough, and the only thing left to do is create your own bootstrap file in the test folder:

bootstrap.php

```php
define('GEMS_PROJECT_NAME', 'ProjectName');
require_once 'library/project-bootstrap.php';
```

Now you can run the tests from your test folder with the command

```
phpunit --bootstrap bootstrap.php ./
```

Available helpers

[Survey Completed Event (survey score script)](#)

From:
https://gemstracker.org/wiki/ - **GemsTracker**

Permanent link:
https://gemstracker.org/wiki/doku.php?id=devzone:testing:project&rev=1370432647

Last update: **2020/03/12 12:08**