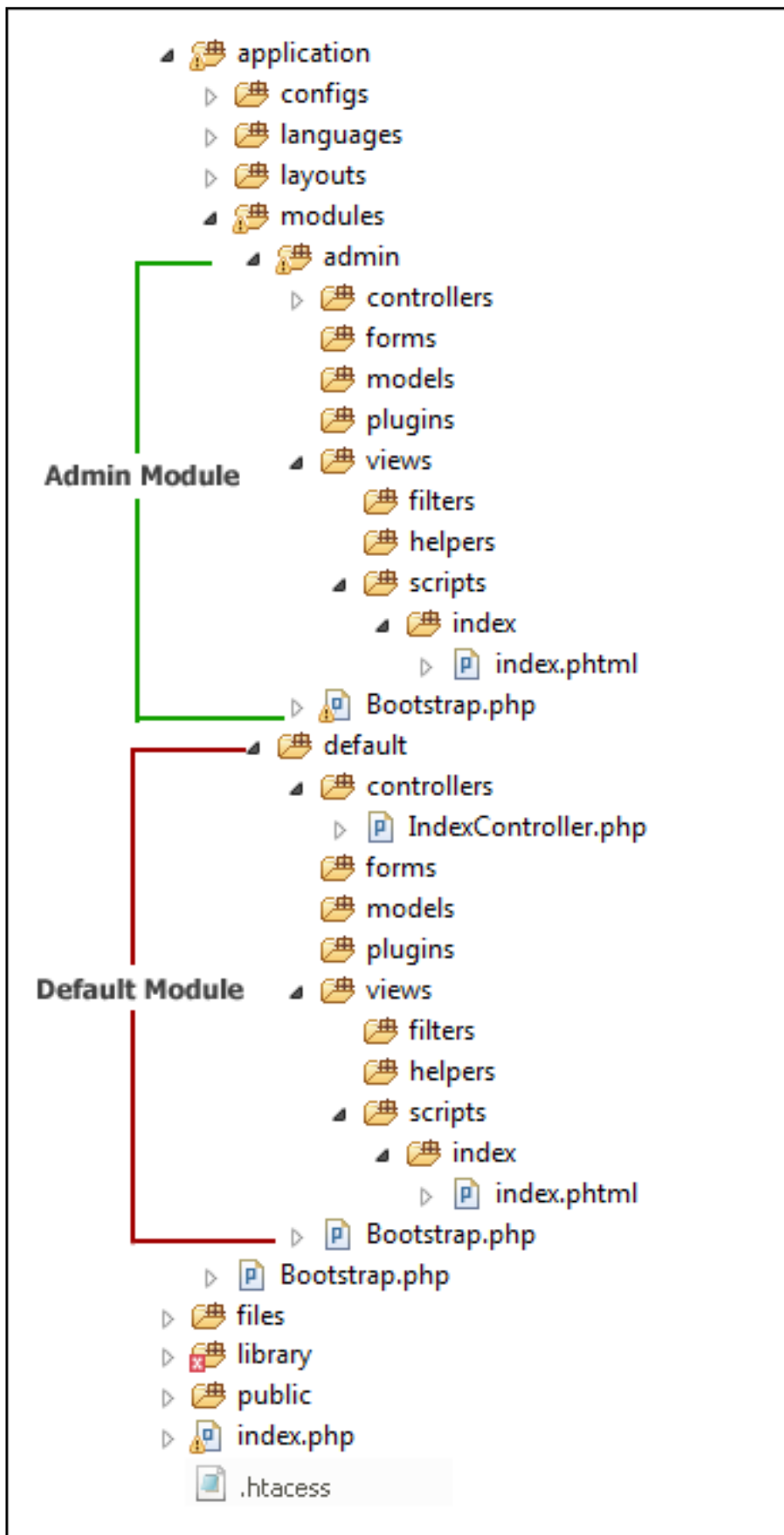


## Introduction of Modules Architecture.

When the application becomes bigger and bigger the controller, models and views/scripts directories contain more and more files. That's a bit odd, because it becomes difficult to maintain, and then the modules come in hand. As our application is based on Zend App there are tons of articles in the web, but let me show you a simple directory layout and ... sample code that sets up the framework.

You can visit [Zend Module Structure](#) for more details. Here is a short description to clear your understanding how our application modules are working.



Each module has its group “controllers, models e views” independent, ie, everything is organized into folders and Zend does the rest (story!). Now Within each module you must add a file Bootstrap.php for initialization of your module, But note that the header is a little different:

### **BOOTSTRAP PRINCIPAL:**

```
class Bootstrap extends Zend_Application_Bootstrap_Bootstrap {  
}
```

### **BOOTSTRAP FOR A MODULE:**

```
class Admin_Bootstrap extends Zend_Application_Module_Bootstrap {  
}
```

Note that in front of the class name Bootstrap you should use the name of your module, in the above case I am adding the bootstrap module admin, then the name should be Admin\_Bootstrap and this class should be to Extent Zend\_Application\_Module\_Bootstrap, because so Zend knows that this is a bootstrap module!

---

Article ID: 53

Last updated: 09 Jul, 2011

Revision: 3

Modules Wise Documentation -> Modules / Extension -> Introduction of Modules Architecture.

<http://www.httpsdoc.com/entry/53/>