Reviewing the Blog Module - tutorials

Throughout the tutorial, we have created a fully functional CRUD module using a blog as an example. While doing so, we've made use of several different design patterns and best-practices. Now it's time to reiterate and take a look at some of the code samples we've written. This is going to be done in a Q&A fashion.

Do we always need all the layers and interfaces?

Short answer: no.

Long answer: The importance of interfaces increases the bigger your application becomes. If you can foresee that your application will be used by other people or should be extendable, then you should strongly consider creating interfaces and coding to them. This is a very common best-practice that is not tied to Zend Framework specifically, but rather more general object oriented programming.

The main role of the multiple layers that we have introduced are to provide a strict separation of concerns for our application.

It is tempting to include your database access directly in your controllers. We recommend splitting it out to other objects, and providing interfaces for the interactions whenever you can. Doing so helps decouple your controllers from the implementation, allowing you to swap out the implementation later without changing the controllers. Using interfaces also simplifies testing, as you can provide mock implementations easily.

Why are there so many controllers?

With the exception of our `ListController`, we created a controller for each route we added.

We could have combined these into a single controller. In practice, we have observed the following when doing

so:

- Controllers grow in complexity, making maintenance and additions more difficult.
- The number of dependencies grows with the number of responsibilities. Many actions may need only a subset of the dependencies, leading to needless performance and resource overhead.
- Testing becomes more difficult.
- Re-use becomes more difficult.

The primary problem is that such controllers quickly break the Single Responsibility Principle, and inherit all the problems that principle attempts to combat.

We recommend a single action per controller whenever possible.

Do you have more questions? PR them!

If there's anything you feel that's missing in this FAQ, please create an issue or send a pull request with your question!
