

After 13 years of PHP programming I decided to share some experience . In my Github profile <https://github.com/Gelembjuk> i uploaded couple my classes/packages which can be useful for other developers.

For now I have shared 4 packages. All are also available with the Composer (<https://packagist.org/>) and are very easy to install (as any package available in composer listings)

Logger

[gelembjuk/logger](https://github.com/Gelembjuk/logger) in composer, <https://github.com/Gelembjuk/logger> in Github.

The package contains classes for logging, error catching and display.

ErrorScreen class helps to catch different types of errors, PHP Exceptions, Warnings and Fatal errors, log error information and display correct error screen to a user depending on expected formats (html, json, xml, http).

FileLogger class based on Psr/Log helps to filter what to log. You can set On/Off different log groups. The main idea of this class is to help to debug big applications. But it would also work as a normal "everyday" logger to a file. It is used in my other packages/classes but can be replaced with any other Psr/Log logger.

ApplicationLogger trait helps to add logging functionality in different classes very easy. You don't need to care about such methods like log(), debug(), setLogger(), getLogger() etc. In your classes when this trait is used. Is really useful big big applications with many classes .

Auth

[gelembjuk/auth](https://github.com/Gelembjuk/auth) in Composer. <https://github.com/Gelembjuk/auth> in Github.

Adds Social Login to a website. Now works with Facebook, Twitter, Google and LinkedIn. It unifies social login functionality for different social networks APIs and uses single interface for all of them. Usage is very simple, a developer only needs to register API keys for each social platform..

The class allows to register/login users on a website using their social network accounts.

Templating

[gelembjuk/templating](https://github.com/Gelembjuk/templating) in Composer. <https://github.com/Gelembjuk/templating> in Github.

It is unified interface for different template engines in PHP. Now it includes wrapper for Smarty and Twig - 2 most popular engines. I plan to add other later.

The idea of this package is to be able to change a template engine of your application without modification of a PHP code.

However, you still have to modify your templates when you change an engine. For example, Smarty and Twig uses different format of templates.

Mail

[gelembjuk/mail](https://github.com/Gelembjuk/mail) in Composer. <https://github.com/Gelembjuk/mail> in Github.

The package helps to format email message, work with localizations and send an email with unified interface supporting popular mailer (PHPMailer etc).

The package contains 2 components - Email Formatter and Email Sender. The EmailFormat allows to generate email message and subject from prepared templates and data. Additionally, it supports internationalization. You can have same templates translated with different languages and based on a specified locale a template will be selected .

Email Sender class is an interface for different PHP mail senders. For now it supports only PHPMailer and mail() function, but can be extended with more classes/libraries . Again, if you use this then you don't need to modify your code when want to change an email sender class/library. Only change a name of the class.

Summary

All that packages are attempt to create some unified or abstract model for "traditional" components of PHP web applications. I

think, the final result of this must be some "abstract PHP framework" where each component can be easy replaced by other class/library doing same work.</p>