Validating Inputs in PHP

Using Valitron: A Validation Library

This is a sample code that showcases the usage of Valitron, a data validation library that does not require any dependency.

For more information on how to use this library, please refer to its documentation that is available on its GitHub repository's Webpage.

NOTE: The following helper functions have been added to the initial version of the `Validator` class:

- `errorsToString()` - Serializes the produced errors into a string.
- `errorsToJson()` - Serializes the produced errors into a *JSON* document.

Using the Native PHP Functions

A helper class named `Input` is also provided in this code sample. This class contains standalone *static* functions that can be used to validate single values. You can add implementation of your own validation functions.

If you wish to add your own validation functions, you need to use some (or a combination) of the following native PHP functions:

- Variable handling Functions
- Character type checking
- Filter Functions

Deploying the Provided Sample on Apache

1. Create a folder named `validation` within `htdocs`
2. Extract the provided `validation.zip` file and copy its content into `htdocs/validation`

NOTE: Ensure that the files are indeed in `htdocs/validation` and not in a sub-subfolder.

Running the Test Functions

1. Using a Web browser window or Thunder Client, initiate a GET request to `http://localhost/validation/`
2. Open `index.php` and uncomment the `function calls` that are included starting from `line #14`.

Adding this Sample into your Slim App

To integrate the `Valitron` library and the helper class into your existing Slim app, you need to copy the `validation` folder to `src/`

NOTE: The Validator and Input classes belong to the `Vanier\Api\Validations` namespace.

Now you can use this code in the route callbacks to validate received values in the:

1. Resource URI
2. Request query string parameter
3. Request body