Project Name Documentation

Release 0.1

Feb 27, 2018

Contents

CHAPTER 1

Introduction

PHP-DataURI is a library which handles URI scheme and provide a way to include data in-line in web pages as if they were external ressources.

This feature is mainly used with HTML5 file API.

This is supposed 100% RFC 2397

CHAPTER 2

Installation

We rely on composer to use this library. If you do no still use composer for your project, you can start with this `composer.json` at the root of your project:

```
{
    "require": {
        "data-uri/data-uri": "dev-master"
    }
}
```

Install composer :

```
# Install composer
curl -s http://getcomposer.org/installer | php
# Upgrade your install
php composer.phar install
```

You now just have to autoload the library to use it :

```
<?php
require 'vendor/autoload.php';
```

This is a very short intro to composer. If you ever experience an issue or want to know more about composer, you will find help on their website http://getcomposer.org/.

CHAPTER 3

PHP dataURI

The PHP-dataURI library is very simple and consists of three main classes :

- One represents a data URI scheme as a PHP object
- One parses the data URI string
- One dumps the data URI PHP object

3.1 Parser

```php
<?php
use DataURI;

$dataString = "data:text/plain;charset=utf-8,%23%24%25";

// Parse one data URI scheme and return a Data object
$dataObject = DataURI\Parser::parse($dataString);

echo $dataObject->getMimeType();
// Output text/plain
echo $dataObject->getData();
// Output #$%
var_dump($dataObject->getParameters());
// Output an array of parameters array('charset' => 'utf-8')
```

3.2 Dumper

```php
<?php
use DataURI;

// Instance a Data object
```

```php
$dataObject = new DataURI\Data("#$%");
// Add some parameters
$dataObject->addParameters('charset' => 'utf-8');

echo DataURI\Dumper::dump($dataObject);
// Output data:text/plain;charset=utf-8,%23%24%25
```

3.3 Dump URI from file

```php
<?php
use DataURI;

$dataObject = DataURI\Data::buildFromFile("/path/to/my/image.png");
echo DataURI\Dumper::dump($dataObject);
// Output ...+S/EAAAAASUVORK5CYII=
```

3.4 Dump URI from url

```php
<?php
use DataURI;

$dataObject = DataURI\Data::buildFromUrl("http://www.example.org/path/to/my/image.png
↪");
echo DataURI\Dumper::dump($dataObject);
// Output ...+S/EAAAAASUVORK5CYII=
```

CHAPTER 4

Handling Exceptions

PHP-dataURI throws 4 different types of exception :

- `\DataURI\Exception\FileNotFoundException` is thrown when an invalid pathfile is supplied or when we don't get a valid response from URL
- `\DataURI\Exception\InvalidDataException` is thrown when raw data could not be decoded
- `\DataURI\Exception\TooLongDataException` is thrown when provided data is too long according to the RFC 2397
- `\DataURI\Exception\InvalidArgumentException` is thrown when provided data URI scheme could not be parsed extends SPL InvalidArgumentException

All these Exception implements `DataURI\Exception\Exception` so you can catch any of these exceptions by catching this exception interface.

CHAPTER 5

Report a bug

If you experience an issue, please report it in our issue tracker. Before reporting an issue, please be sure that it is not already reported by browsing open issues.

CHAPTER 6

Contribute

You find a bug and resolved it ? You added a feature and want to share ? You found a typo in this doc and fixed it ? Feel free to send a Pull Request on GitHub, we will be glad to merge your code.

CHAPTER 7

Run tests

PHP-dataURI relies on PHPUnit for unit tests. To run tests on your system, ensure you have PHPUnit installed, and, at the root of PHP-dataURI, execute it :

```
phpunit
```

CHAPTER 8

About

PHP-dataURI has been written by Nicolas Le Goff @ Alchemy for Phraseanet, our DAM software. Try it, it's awesome !

CHAPTER 9

License

PHP-dataURI is licensed under the MIT License