

My Project

Md.Mehedi hassan
2131801642

Deprecated List

Global [InstalledVersions::getRawData](#) ()

Use [getAllRawData\(\)](#) instead which returns all datasets for all autoloaders present in the process.
[getRawData](#) only returns the first dataset loaded, which may not be what you expect.

Namespace Index

Namespace List

Here is a list of all namespaces with brief descriptions:

Composer	3
Composer\Autoload	3

Data Structure Index

Data Structures

Here are the data structures with brief descriptions:

ClassLoader	3
ComposerAutoloaderInit4b55c7ef33235a0353c1b6b079f86964	8
ComposerStaticInit4b55c7ef33235a0353c1b6b079f86964	9
InstalledVersions	10

File Index

File List

Here is a list of all files with brief descriptions:

livesearch.php	14
order.php	15
search.php	18
vendor/autoload.php	21
vendor/composer/autoload_classmap.php	22
vendor/composer/autoload_files.php	23
vendor/composer/autoload_namespaces.php	24
vendor/composer/autoload_psr4.php	24
vendor/composer/autoload_real.php	25
vendor/composer/autoload_static.php	26
vendor/composer/ClassLoader.php	27
vendor/composer/installed.php	33
vendor/composer/InstalledVersions.php	35
vendor/composer/platform_check.php	38

Namespace Documentation

Composer Namespace Reference

Namespaces

- namespace [Autoload](#)

Data Structures

class [InstalledVersions](#)

Composer\Autoload Namespace Reference

Data Structures

- class [ClassLoader](#)class [ComposerStaticInit4b55c7ef33235a0353c1b6b079f86964](#)
-

Data Structure Documentation

ClassLoader Class Reference

Public Member Functions

- [__construct](#) (\$vendorDir=null)
- [getPrefixes](#) ()
- [getPrefixesPsr4](#) ()
- [getFallbackDirs](#) ()
- [getFallbackDirsPsr4](#) ()
- [getClassMap](#) ()
- [addClassMap](#) (array \$classMap)
- [add](#) (\$prefix, \$paths, \$prepend=false)
- [addPsr4](#) (\$prefix, \$paths, \$prepend=false)
- [set](#) (\$prefix, \$paths)
- [setPsr4](#) (\$prefix, \$paths)
- [setUseIncludePath](#) (\$useIncludePath)
- [getUseIncludePath](#) ()
- [setClassMapAuthoritative](#) (\$classMapAuthoritative)
- [isClassMapAuthoritative](#) ()
- [setApcuPrefix](#) (\$apcuPrefix)
- [getApcuPrefix](#) ()
- [register](#) (\$prepend=false)
- [unregister](#) ()
- [loadClass](#) (\$class)
- [findFile](#) (\$class)

Static Public Member Functions

- static [getRegisteredLoaders](#) ()
-

Detailed Description

ClassLoader implements a PSR-0, PSR-4 and classmap class loader.

```
$loader = new \Composer\Autoload\ClassLoader();
```

```
// register classes with namespaces
$loader->add('Symfony\Component', __DIR__.'/component');
$loader->add('Symfony',           __DIR__.'/framework');

// activate the autoloader
$loader->register();

// to enable searching the include path (eg. for PEAR packages)
$loader->setUseIncludePath(true);
```

In this example, if you try to use a class in the `Symfony\Component` namespace or one of its children (`Symfony\Component\Console` for instance), the autoloader will first look for the class under the `component/` directory, and it will then fallback to the `framework/` directory if not found before giving up.

This class is loosely based on the `Symfony UniversalClassLoader`.

Author

Fabien Potencier fabien@symfony.com

Jordi Boggiano j.boggiano@seld.be

See also

<https://www.php-fig.org/psr/psr-0/>

<https://www.php-fig.org/psr/psr-4/>

Definition at line 43 of file [ClassLoader.php](#).

Constructor & Destructor Documentation

__construct (\$vendorDir = null)

Parameters

string null	<i>\$vendorDir</i>	
---------------	--------------------	--

Definition at line 106 of file [ClassLoader.php](#).

Member Function Documentation

add (\$prefix, \$paths, \$prepend = false)

Registers a set of PSR-0 directories for a given prefix, either appending or prepending to the ones previously set for this prefix.

Parameters

string	<i>\$prefix</i>	The prefix
	<i>list<string>/string</i>	<i>\$paths</i> The PSR-0 root directories
bool	<i>\$prepend</i>	Whether to prepend the directories

Returns

void

Definition at line 180 of file [ClassLoader.php](#).

addClassMap (array \$classMap)

Parameters

<i>array<string,string></i>	<i>\$classMap</i> Class to filename map
-----------------------------------	---

<i>g</i> >	
------------	--

Returns

void

Definition at line [161](#) of file [ClassLoader.php](#).

addPsr4 (\$prefix, \$paths, \$prepend = false)

Registers a set of PSR-4 directories for a given namespace, either appending or prepending to the ones previously set for this namespace.

Parameters

string	<i>\$prefix</i>	The prefix/namespace, with trailing '\'
	<i>list<string>/string</i>	\$paths The PSR-4 base directories
bool	<i>\$prepend</i>	Whether to prepend the directories

Exceptions

--	--

InvalidArgumentException

Returns

void

Definition at line [230](#) of file [ClassLoader.php](#).

findFile (\$class)

Finds the path to the file where the class is defined.

Parameters

string	<i>\$class</i>	The name of the class
--------	----------------	-----------------------

Returns

string|false The path if found, false otherwise

Definition at line [442](#) of file [ClassLoader.php](#).

getApcuPrefix ()

The APCu prefix in use, or null if APCu caching is not enabled.

Returns

string|null

Definition at line [375](#) of file [ClassLoader.php](#).

getClassMap ()

Returns

array<string, string> Array of classname => path

Definition at line [151](#) of file [ClassLoader.php](#).

getFallbackDirs ()

Returns

list<string>

Definition at line [135](#) of file [ClassLoader.php](#).

getFallbackDirsPsr4 ()

Returns

list<string>

Definition at line [143](#) of file [ClassLoader.php](#).

getPrefixes ()**Returns**

array<string, list<string>>

Definition at line [115](#) of file [ClassLoader.php](#).

getPrefixesPsr4 ()**Returns**

array<string, list<string>>

Definition at line [127](#) of file [ClassLoader.php](#).

static getRegisteredLoaders () [static]

Returns the currently registered loaders keyed by their corresponding vendor directories.

Returns

array<string, self>

Definition at line [482](#) of file [ClassLoader.php](#).

getUseIncludePath ()

Can be used to check if the autoloader uses the include path to check for classes.

Returns

bool

Definition at line [330](#) of file [ClassLoader.php](#).

isClassMapAuthoritative ()

Should class lookup fail if not found in the current class map?

Returns

bool

Definition at line [353](#) of file [ClassLoader.php](#).

loadClass (\$class)

Loads the given class or interface.

Parameters

string	<i>\$class</i>	The name of the class
--------	----------------	-----------------------

Returns

true|null True if loaded, null otherwise

Definition at line [423](#) of file [ClassLoader.php](#).

register (\$prepend = false)

Registers this instance as an autoloader.

Parameters

bool	<i>\$prepend</i>	Whether to prepend the autoloader or not
------	------------------	--

Returns

void

Definition at line [387](#) of file [ClassLoader.php](#).**set (\$prefix, \$paths)**

Registers a set of PSR-0 directories for a given prefix, replacing any others previously set for this prefix.

Parameters

string	<i>\$prefix</i>	The prefix
	<i>list<string>/string</i>	\$paths The PSR-0 base directories

Returns

void

Definition at line [278](#) of file [ClassLoader.php](#).**setApcuPrefix (\$apcuPrefix)**

APCu prefix to use to cache found/not-found classes, if the extension is enabled.

Parameters

string null	<i>\$apcuPrefix</i>	
---------------	---------------------	--

Returns

void

Definition at line [365](#) of file [ClassLoader.php](#).**setClassMapAuthoritative (\$classMapAuthoritative)**

Turns off searching the prefix and fallback directories for classes that have not been registered with the class map.

Parameters

bool	<i>\$classMapAuthoritative</i>	
------	--------------------------------	--

Returns

void

Definition at line [343](#) of file [ClassLoader.php](#).**setPsr4 (\$prefix, \$paths)**

Registers a set of PSR-4 directories for a given namespace, replacing any others previously set for this namespace.

Parameters

string	<i>\$prefix</i>	The prefix/namespace, with trailing '\'
	<i>list<string>/string</i>	\$paths The PSR-4 base directories

Exceptions

--	--

InvalidArgumentException

Returns

void

Definition at line [298](#) of file [ClassLoader.php](#).**setUseIncludePath (\$useIncludePath)**

Turns on searching the include path for class files.

Parameters

bool	<i>\$useIncludePath</i>	
------	-------------------------	--

Returns

void

Definition at line [319](#) of file [ClassLoader.php](#).

unregister ()

Unregisters this instance as an autoloader.

Returns

void

Definition at line [408](#) of file [ClassLoader.php](#).

The documentation for this class was generated from the following file:

- vendor/composer/[ClassLoader.php](#)

ComposerAutoloaderInit4b55c7ef33235a0353c1b6b079f86964 Class Reference

Static Public Member Functions

- static [loadClassLoader](#) (\$class)
- static [getLoader](#) ()

Detailed Description

Definition at line [5](#) of file [autoload_real.php](#).

Member Function Documentation

static getLoader () [static]**Returns**

\Composer\Autoload\ClassLoader

Definition at line [19](#) of file [autoload_real.php](#).

static loadClassLoader (\$class) [static]

Definition at line [9](#) of file [autoload_real.php](#).

The documentation for this class was generated from the following file:

- vendor/composer/[autoload_real.php](#)
-

ComposerStaticInit4b55c7ef33235a0353c1b6b079f86964 Class Reference

Static Public Member Functions

- static [getInitializer](#) ([ClassLoader](#) \$loader)

Static Public Attributes

- static [\\$files](#)
- static [\\$prefixLengthsPsr4](#)
- static [\\$prefixDirsPsr4](#)
- static [\\$classMap](#)

Detailed Description

Definition at line [7](#) of file [autoload_static.php](#).

Member Function Documentation

static [getInitializer](#) ([ClassLoader](#) \$loader)[**static**]

Definition at line [65](#) of file [autoload_static.php](#).

Field Documentation

\$classMap[**static**]

Initial value:

```
= array (
    'Composer\\InstalledVersions' => __DIR__ . '/../' .
    '/composer/InstalledVersions.php',
)
```

Definition at line [61](#) of file [autoload_static.php](#).

\$files[**static**]

Initial value:

```
= array (
    '7b11c4dc42b3b3023073cb14e519683c' => __DIR__ . '/../' .
    '/ralouphie/getallheaders/src/getallheaders.php',
    '6e3fae29631ef280660b3cdad06f25a8' => __DIR__ . '/../' .
    '/symfony/deprecation-contracts/function.php',
    '37a3dc5111fe8f707ab4c132ef1dbc62' => __DIR__ . '/../' .
    '/guzzlehttp/guzzle/src/functions_include.php',
)
```

Definition at line [9](#) of file [autoload_static.php](#).

\$prefixDirsPsr4[**static**]

Initial value:

```
= array (
    'Stripe\\' =>
        array (
            0 => __DIR__ . '/../' . '/stripe/stripe-php/lib',
        ),
    'Psr\\Http\\Message\\' =>
        array (
```

```

        0 => __DIR__ . '/../' . '/psr/http-factory/src',
        1 => __DIR__ . '/../' . '/psr/http-message/src',
    ),
    'Psr\\Http\\Client\\' =>
    array (
        0 => __DIR__ . '/../' . '/psr/http-client/src',
    ),
    'GuzzleHttp\\Psr7\\' =>
    array (
        0 => __DIR__ . '/../' . '/guzzlehttp/psr7/src',
    ),
    'GuzzleHttp\\Promise\\' =>
    array (
        0 => __DIR__ . '/../' . '/guzzlehttp/promises/src',
    ),
    'GuzzleHttp\\' =>
    array (
        0 => __DIR__ . '/../' . '/guzzlehttp/guzzle/src',
    ),
)

```

Definition at line [33](#) of file [autoload_static.php](#).

\$prefixLengthsPsr4[static]

Initial value:

```

= array (
    'S' =>
    array (
        'Stripe\\' => 7,
    ),
    'P' =>
    array (
        'Psr\\Http\\Message\\' => 17,
        'Psr\\Http\\Client\\' => 16,
    ),
    'G' =>
    array (
        'GuzzleHttp\\Psr7\\' => 16,
        'GuzzleHttp\\Promise\\' => 19,
        'GuzzleHttp\\' => 11,
    ),
)

```

Definition at line [15](#) of file [autoload_static.php](#).

The documentation for this class was generated from the following file:

- vendor/composer/[autoload_static.php](#)

InstalledVersions Class Reference

Static Public Member Functions

- static [getInstalledPackages](#) ()
- static [getInstalledPackagesByType](#) (\$type)
- static [isInstalled](#) (\$packageName, \$includeDevRequirements=true)
- static [satisfies](#) (VersionParser \$parser, \$packageName, \$constraint)
- static [getVersionRanges](#) (\$packageName)
- static [getVersion](#) (\$packageName)
- static [getPrettyVersion](#) (\$packageName)
- static [getReference](#) (\$packageName)
- static [getInstallPath](#) (\$packageName)
- static [getRootPackage](#) ()
- static [getRawData](#) ()

- static [getAllRawData](#) ()
- static [reload](#) (\$data)

Detailed Description

This class is copied in every [Composer](#) installed project and available to all

See also <https://getcomposer.org/doc/07-runtime.md#installed-versions>

To require its presence, you can require `composer-runtime-api ^2.0`

@final

Definition at line [27](#) of file [InstalledVersions.php](#).

Member Function Documentation

static [getAllRawData](#) () [`static`]

Returns the raw data of all [installed.php](#) which are currently loaded for custom implementations

Returns

```
array[] @psalm-return list<array{root: array{ name: string, pretty_version: string, version: string, reference: string|null, type: string, install_path: string, aliases: string[], dev: bool}, versions: array<string, array{pretty_version?: string, version?: string, reference?: string|null, type?: string, install_path?: string, aliases?: string[], dev_requirement: bool, replaced?: string[], provided?: string[]}>>>
```

Definition at line [285](#) of file [InstalledVersions.php](#).

static [getInstalledPackages](#) () [`static`]

Returns a list of all package names which are present, either by being installed, replaced or provided

Returns

```
string[] @psalm-return list<string>
```

Definition at line [52](#) of file [InstalledVersions.php](#).

static [getInstalledPackagesByType](#) (\$type) [`static`]

Returns a list of all package names with a specific type e.g. 'library'

Parameters

string	<i>\$type</i>	
--------	---------------	--

Returns

```
string[] @psalm-return list<string>
```

Definition at line [73](#) of file [InstalledVersions.php](#).

static [getInstallPath](#) (\$packageName) [`static`]

Parameters

string	<i>\$packageName</i>	
--------	----------------------	--

Returns

string|null If the package is being replaced or provided but is not really installed, null will be returned as install path. Packages of type metapackages also have a null install path.

Definition at line [231](#) of file [InstalledVersions.php](#).

static getPrettyVersion (\$packageName)[static]

Parameters

string	<i>\$packageName</i>	
--------	----------------------	--

Returns

string|null If the package is being replaced or provided but is not really installed, null will be returned as version, use satisfies or getVersionRanges if you need to know if a given version is present

Definition at line [189](#) of file [InstalledVersions.php](#).

static getRawData ()[static]

Returns the raw [installed.php](#) data for custom implementations

Deprecated:

Use getAllRawData() instead which returns all datasets for all autoloaders present in the process. getRawData only returns the first dataset loaded, which may not be what you expect.

Returns

array[] @psalm-return array{root: array{name: string, pretty_version: string, version: string, reference: string|null, type: string, install_path: string, aliases: string[], dev: bool}, versions: array<string, array{pretty_version?: string, version?: string, reference?: string|null, type?: string, install_path?: string, aliases?: string[], dev_requirement: bool, replaced?: string[], provided?: string[]}>}

Definition at line [262](#) of file [InstalledVersions.php](#).

static getReference (\$packageName)[static]

Parameters

string	<i>\$packageName</i>	
--------	----------------------	--

Returns

string|null If the package is being replaced or provided but is not really installed, null will be returned as reference

Definition at line [210](#) of file [InstalledVersions.php](#).

static getRootPackage ()[static]

Returns

array @psalm-return array{name: string, pretty_version: string, version: string, reference: string|null, type: string, install_path: string, aliases: string[], dev: bool}

Definition at line [248](#) of file [InstalledVersions.php](#).

static getVersion (\$packageName)[static]

Parameters

string	<i>\$packageName</i>	
--------	----------------------	--

Returns

string|null If the package is being replaced or provided but is not really installed, null will be returned as version, use satisfies or getVersionRanges if you need to know if a given version is present

Definition at line [168](#) of file [InstalledVersions.php](#).

static getVersionRanges (\$packageName)[static]

Returns a version constraint representing all the range(s) which are installed for a given package

It is easier to use this via isInstalled() with the \$constraint argument if you need to check whether a given version of a package is installed, and not just whether it exists

Parameters

string	<i>\$packageName</i>	
--------	----------------------	--

Returns

string Version constraint usable with composer/semver

Definition at line [137](#) of file [InstalledVersions.php](#).

static isInstalled (\$packageName, \$includeDevRequirements = true)[static]

Checks whether the given package is installed

This also returns true if the package name is provided or replaced by another package

Parameters

string	<i>\$packageName</i>	
bool	<i>\$includeDevRequirements</i>	

Returns

bool

Definition at line [97](#) of file [InstalledVersions.php](#).

static reload (\$data)[static]

Lets you reload the static array from another file

This is only useful for complex integrations in which a project needs to use this class but then also needs to execute another project's autoloader in process, and wants to ensure both projects have access to their version of [installed.php](#).

A typical case would be PHPUnit, where it would need to make sure it reads all the data it needs from this class, then call reload() with require \$CWD/vendor/composer/installed.php (or similar) as input to make sure the project in which it runs can then also use this class safely, without interference between PHPUnit's dependencies and the project's dependencies.

Parameters

array[]	<i>\$data</i>	A vendor/composer/installed.php data set
---------	---------------	--

Returns

void

@psalm-param array{root: array{name: string, pretty_version: string, version: string, reference: string|null, type: string, install_path: string, aliases: string[], dev: bool}, versions: array<string, array{pretty_version?: string, version?: string, reference?: string|null, type?: string, install_path?: string, aliases?: string[], dev_requirement: bool, replaced?: string[], provided?: string[]}>} \$data

Definition at line [308](#) of file [InstalledVersions.php](#).

static satisfies (VersionParser \$parser, \$packageName, \$constraint)[static]

Checks whether the given package satisfies a version constraint

e.g. If you want to know whether version 2.3+ of package foo/bar is installed, you would call:

[Composer](#)\InstalledVersions::satisfies(new VersionParser, 'foo/bar', '^2.3')

Parameters

VersionParser	<i>\$parser</i>	Install composer/semver to have access to this class and functionality
string	<i>\$packageName</i>	
string null	<i>\$constraint</i>	A version constraint to check for, if you pass one you have to make sure composer/semver is required by your package

Returns

bool

Definition at line [120](#) of file [InstalledVersions.php](#).

The documentation for this class was generated from the following file:

- vendor/composer/[InstalledVersions.php](#)

File Documentation

livesearch.php File Reference

livesearch.php

[Go to the documentation of this file.](#)

```
00001 <?php
00002 include('config/constants.php');
00003
00020 if (isset($_GET['q'])) {
00026     $search = mysqli_real_escape_string($conn, $_GET['q']); // Sanitize the search
input
00027     echo "Search Query: " . $search; // Debug output for search query
00028
00036     $sql = "SELECT * FROM food WHERE title LIKE '%$search%' OR description LIKE
 '%$search%'";
00037
00038     // Execute the query
00039     $res = mysqli_query($conn, $sql);
00040     $count = mysqli_num_rows($res);
00041
00042     if ($count > 0) {
00049         while ($row = mysqli_fetch_assoc($res)) {
00050             $id = $row['id'];
00051             $title = $row['title'];
00052             $price = $row['price'];
00053             $description = $row['description'];
00054             $image_name = $row['image_name'];
00055
00056             // HTML structure for displaying the search results
00057             echo '<div class="food-menu-box">';
00058             echo '<div class="food-menu-img">';
00059
00065             if ($image_name == "") {
00066                 echo "<div class='error'>Image not Available.</div>";
00067             } else {
00068                 echo '';
```

```

00069         }
00070
00071         echo '</div>';
00072         echo '<div class="food-menu-desc">';
00073         echo '<h4>' . $title . '</h4>';
00074         echo '<p class="food-price">Tk. ' . $price . '</p>';
00075         echo '<p class="food-detail">' . $description . '</p>';
00076         echo '<br>';
00077         echo '<a href="#" class="btn btn-primary">Order Now</a>';
00078         echo '</div>';
00079         echo '</div>';
00080     }
00081 } else {
00082     echo "÷ class='error'>Food not found.</div>";
00083 }
00084 }
00085 }
00086 }
00087 }
00088 }
00089 }
00090 ?>

```

order.php File Reference

Variables

- [if](#)(isset(\$_GET['food_id'])) [else](#)
Handles the order placement and payment processing for food orders.
- [if](#) (isset(\$_POST['stripeToken'])) {
Processes the payment via Stripe after receiving the Stripe token.

Variable Documentation

[if](#) (\$image_name=="") else

Initial value:

```

{
    header('location:' . SITEURL)
}

```

Handles the order placement and payment processing for food orders.

Check if the image for the selected food is available.

This script fetches the food details based on the food_id from the URL and displays a form for the user to fill in their delivery details and payment information. If the payment is successful via Stripe, the order is inserted into the database.

If the image is not available, an error message is displayed.

SQL query to fetch food details based on the provided food_id.

The query retrieves the title, price, and image name for the food item from the database.

Definition at line [32](#) of file [order.php](#).

[if](#) (isset(\$_POST['stripeToken'])) {

Processes the payment via Stripe after receiving the Stripe token.

Check if the image for the selected food is available.

If the payment is successful, the order details are inserted into the database.

Note

The payment uses Stripe's Checkout API for processing.

If the image is not available, an error message is displayed.

SQL query to insert the order details into the database.

The order information such as customer details, food ordered, quantity, and price are stored in the orders table.

Definition at line [109](#) of file [order.php](#).

order.php

[Go to the documentation of this file.](#)

```
00001 <?php include('partials-front/menu.php'); ?>
00002
00003 <?php
00011 if (isset($_GET['food_id'])) {
00012     $food_id = $_GET['food_id'];
00013
00019     $sql = "SELECT * FROM food WHERE id = $food_id";
00020     $res = mysqli_query($conn, $sql);
00021     $count = mysqli_num_rows($res);
00022
00023     if ($count == 1) {
00024         $row = mysqli_fetch_assoc($res);
00025         $title = $row['title'];
00026         $price = $row['price'];
00027         $image_name = $row['image_name'];
00028     } else {
00029         // Redirect if no food is found
00030         header('location:' . SITEURL);
00031     }
00032 } else {
00033     // Redirect if food_id is not set
00034     header('location:' . SITEURL);
00035 }
00036 ?>
00037
00038 <section class="food-search">
00039     <div class="container">
00040         <h2 class="text-center text-white">Fill this form to confirm your
order.</h2>
00041
00042         <form action="" method="POST" class="order">
00043             <fieldset>
00044                 <legend>Selected Food</legend>
00045
00046                 <div class="food-menu-img">
00047                     <?php
00053                     if ($image_name == "") {
00054                         echo "<div class='error'>Image not Available.</div>";
00055                     } else {
00056                         ?>
00057                         "
00058                             class="img-responsive img-curve">
00059                     <?php
00060                     }
00061                     ?>
00062                 </div>
00063
00064                 <div class="food-menu-desc">
00065                     <h3><?php echo $title; ?></h3>
00066                     <input type="hidden" name="food" value="<?php echo $title; ?>">
00067                     <p class="food-price">Tk.<?php echo $price; ?></p>
```



```

00068         <input type="hidden" name="price" value="<?php echo $price;
?>">
00069         <div class="order-label">Quantity</div>
00070         <input type="number" name="qty" class="input-responsive"
value="1" required>
00071     </div>
00072 </fieldset>
00073
00074     <fieldset>
00075         <legend>Delivery Details</legend>
00076         <div class="order-label">Full Name</div>
00077         <input type="text" name="full-name" placeholder="E.g. Dhrubo,
Homaيرا, Asif" class="input-responsive" required>
00078
00079         <div class="order-label">Phone Number</div>
00080         <input type="tel" name="contact" placeholder="E.g. 9843xxxxxx"
class="input-responsive" required>
00081
00082         <div class="order-label">Email</div>
00083         <input type="email" name="email" placeholder="E.g.
ahmed.dhrubo@northsouth.edu" class="input-responsive" required>
00084
00085         <div class="order-label">Address</div>
00086         <textarea name="address" rows="10" placeholder="E.g. Street, City,
Country" class="input-responsive" required></textarea>
00087
00088         <!-- Stripe payment form -->
00089         <script src="https://checkout.stripe.com/checkout.js"
class="stripe-button"
00090         data-key="pk_test_51NzgKAFTSKcvcvKsqaw00CM2ldnnjFH0MfMnT0smbYy65eGEz9rlvzDWRgWn2qYGgfo
UDS5SalimEvEYjfYxu2rJP00D4Tmb41B"
00091             data-amount="<?php echo $price * 100; ?>"
00092             data-name="Food Order"
00093             data-description="Payment for Food Order"
00094             data-image="https://example.com/logo.png"
00095             data-locale="auto"
00096             data-currency="usd">
00097         </script>
00098     </fieldset>
00099 </form>
00100
00101 <?php
00102 if (isset($_POST['stripeToken'])) {
00103     // Include Stripe PHP library
00104     require 'vendor/autoload.php';
00105
00106     // Set your Stripe API key
00107     \Stripe\Stripe::setApiKey('sk_test_51NzgKAFTSKcvcvKsxAHPRWNG3Gtv0SVGXk3vyOftXRDMiS95uI
6YwFNEgQJyuwa5wSzU8CvVRremEKMm3yZzYape001XT4xgDq');
00108
00109     // Get the payment token submitted by the form
00110     $token = $_POST['stripeToken'];
00111
00112     try {
00113         // Create a charge using the token
00114         $charge = \Stripe\Charge::create([
00115             'amount' => $price * 100, // Amount in cents (e.g., $10.00)
00116             'currency' => 'usd',
00117             'source' => $token,
00118             'description' => 'Payment for Food Order',
00119         ]);
00120
00121         // Payment successful, update database or perform other actions
00122         if ($charge->status === 'succeeded') {
00123             // Insert order details into the database
00124             $fullName = mysqli_real_escape_string($conn,
$_POST['full-name']);
00125             $contact = mysqli_real_escape_string($conn,
$_POST['contact']);
00126             $email = mysqli_real_escape_string($conn, $_POST['email']);
00127             $address = mysqli_real_escape_string($conn,
$_POST['address']);
00128             $foodName = mysqli_real_escape_string($conn, $_POST['food']);
00129             $price = mysqli_real_escape_string($conn, $_POST['price']);
00130             $quantity = mysqli_real_escape_string($conn, $_POST['qty']);

```

```

00138
00145             $insertOrderSQL = "INSERT INTO orders (customer_name,
customer_contact, customer_email, customer_address, food, price, qty)
00146                 VALUES ('$fullName', '$contact',
'$email', '$address', '$foodName', '$price', '$quantity')";
00147
00148             // Execute the SQL query
00149             if (mysqli_query($conn, $insertOrderSQL)) {
00150                 // Order inserted successfully
00151                 echo "Payment successful. Your order has been placed!";
00152             } else {
00153                 // Error inserting order
00154                 echo "Payment successful, but there was an issue storing the
order in the database.";
00155             }
00156         }
00157     } catch (\Stripe\Exception\CardException $e) {
00158         // Handle declined card
00159         echo "Payment failed: " . $e->getError()->message;
00160     }
00161 }
00162 ?>
00163 </div>
00164 </section>
00165
00166 <?php include('partials-front/footer.php'); ?>

```

search.php File Reference

Variables

- [if](#)(\$_SERVER["REQUEST_METHOD"]=="POST") [else](#) [\\$sql](#) = "SELECT * FROM food WHERE active='Yes'"
Handles the search functionality for food items based on user input.
- [\\$res](#) = mysqli_query(\$conn, \$sql)
- [\\$count](#) = mysqli_num_rows(\$res)
- [if](#)(\$count > 0) [else](#)

Variable Documentation

\$count = mysqli_num_rows(\$res)

Definition at line [116](#) of file [search.php](#).

\$res = mysqli_query(\$conn, \$sql)

Definition at line [115](#) of file [search.php](#).

[if](#) (\$_SERVER["REQUEST_METHOD"]=="POST") [else](#) [\\$sql](#) = "SELECT * FROM food WHERE active='Yes'"

Handles the search functionality for food items based on user input.

This script checks if a search query is submitted via the POST method, sanitizes the input, and then searches the database for matching food items by title or description. If food items are found, they are displayed; otherwise, a "Food not found" message is shown.


```

00048         ?>
00049
00050         <div class="food-menu-box">
00051             <div class="food-menu-img">
00052                 <?php
00058                     if ($image_name == "") {
00059                         echo "<div class='error'>Image not
Available.</div>";
00060                     } else {
00061                         ?>
00062                         
00063                     <?php
00064                     }
00065                     ?>
00066                 </div>
00067
00068                 <div class="food-menu-desc">
00069                     <h4><?php echo $title; ?></h4>
00070                     <p class="food-price"><?php echo $price; ?></p>
00071                     <p class="food-detail">
00072                         <?php echo $description; ?>
00073                     </p>
00074                     <br>
00075
00076                     <a href="#" class="btn btn-primary">Order Now</a>
00077                 </div>
00078             </div>
00079
00080             <?php
00081             }
00082         } else {
00083             echo "<div class='error'>Food not found.</div>";
00084         }
00085         ?>
00086
00087         <div class="clearfix"></div>
00088     </div>
00089 </section>
00090
00091 <?php
00092 include('partials-front/footer.php');
00093 } else {
00094     ?>
00095
00096     <section class="food-search text-center">
00097         <div class="container">
00098             <h2>Search for Foods</h2>
00099             <input type="text" size="30" id="search"
onkeyup="showResult(this.value)">
00100             <div id="livesearch"></div>
00101         </div>
00102     </section>
00103
00104     <section class="food-menu">
00105         <div class="container">
00106             <h2 class="text-center">Food Menu</h2>
00107
00108             <?php
00114             $sql = "SELECT * FROM food WHERE active='Yes'";
00115             $res = mysqli_query($conn, $sql);
00116             $count = mysqli_num_rows($res);
00117
00118             if ($count > 0) {
00119                 while ($row = mysqli_fetch_assoc($res)) {
00120                     // Variables for each active food item
00121                     $id = $row['id'];
00122                     $title = $row['title'];
00123                     $description = $row['description'];
00124                     $price = $row['price'];
00125                     $image_name = $row['image_name'];
00126                     ?>
00127
00128                     <div class="food-menu-box">
00129                         <div class="food-menu-img">
00130                             <?php
00136                             if ($image_name == "") {

```

```

00137             echo "<div class='error'>Image not
available.</div>";
00138         } else {
00139             ?>
00140             
00141             <?php
00142             }
00143             ?>
00144         </div>
00145
00146         <div class="food-menu-desc">
00147             <h4><?php echo $title; ?></h4>
00148             <p class="food-price">Tk.<?php echo $price; ?></p>
00149             <p class="food-detail">
00150                 <?php echo $description; ?>
00151             </p>
00152             <br>
00153
00154             <a href="<?php echo SITEURL; ?>order.php?food_id=<?php
echo $id; ?>" class="btn btn-primary">Order Now</a>
00155             </div>
00156         </div>
00157
00158         <?php
00159         }
00160     } else {
00161         echo "<div class='error'>Food not found.</div>";
00162     }
00163     ?>
00164
00165     <div class="clearfix"></div>
00166 </div>
00167 </section>
00168
00169 <script>
00170     function showResult(str) {
00171         if (str.length == 0) {
00172             document.getElementById("livesearch").innerHTML = "";
00173             return;
00174         }
00175         var xmlhttp = new XMLHttpRequest();
00176         xmlhttp.onreadystatechange = function () {
00177             if (this.readyState == 4 && this.status == 200) {
00178                 document.getElementById("livesearch").innerHTML =
this.responseText;
00179             }
00180         }
00181         xmlhttp.open("GET", "livesearch.php?q=" + str, true);
00182         xmlhttp.send();
00183     }
00184 </script>
00185
00186 <div class="clearfix"></div>
00187
00188 <?php
00189 include('partials-front/footer.php');
00190 }
00191 ?>

```

vendor/autoload.php File Reference

Variables

- [if](#) (PHP_VERSION_ID < 50600)
-

Variable Documentation

if(PHP_VERSION_ID< 50600) ()

Definition at line [5](#) of file [autoload.php](#).

autoload.php

[Go to the documentation of this file.](#)

```
00001 <?php
00002
00003 // autoload.php @generated by Composer
00004
00005 if (PHP_VERSION_ID < 50600) {
00006     if (!headers_sent()) {
00007         header('HTTP/1.1 500 Internal Server Error');
00008     }
00009     $err = 'Composer 2.3.0 dropped support for autoloading on PHP <5.6 and you are
running '.PHP_VERSION.', please upgrade PHP or use Composer 2.2 LTS via "composer
self-update --2.2". Aborting.'.PHP_EOL;
00010     if (!ini_get('display_errors')) {
00011         if (PHP_SAPI === 'cli' || PHP_SAPI === 'phpdbg') {
00012             fwrite(STDERR, $err);
00013         } elseif (!headers_sent()) {
00014             echo $err;
00015         }
00016     }
00017     trigger_error(
00018         $err,
00019         E_USER_ERROR
00020     );
00021 }
00022
00023 require_once __DIR__ . '/composer/autoload_real.php';
00024
00025 return ComposerAutoloaderInit4b55c7ef33235a0353c1b6b079f86964::getLoader();
```

vendor/composer/autoload_classmap.php File Reference

Variables

- [\\$vendorDir](#) = dirname(__DIR__)
- [\\$baseDir](#) = dirname(\$vendorDir)

Variable Documentation

\$baseDir = dirname(\$vendorDir)

Definition at line [6](#) of file [autoload_classmap.php](#).

\$vendorDir = dirname(__DIR__)

Definition at line [5](#) of file [autoload_classmap.php](#).

autoload_classmap.php

[Go to the documentation of this file.](#)

```
00001 <?php
00002
00003 // autoload_classmap.php @generated by Composer
00004
00005 $vendorDir = dirname(__DIR__);
00006 $baseDir = dirname($vendorDir);
00007
00008 return array(
00009     'Composer\\InstalledVersions' => $vendorDir .
00010     '/composer/InstalledVersions.php',
00011 );
```

vendor/composer/autoload_files.php File Reference

Variables

- [\\$vendorDir](#) = `dirname(__DIR__)`
- [\\$baseDir](#) = `dirname($vendorDir)`

Variable Documentation

`$baseDir = dirname($vendorDir)`

Definition at line [6](#) of file [autoload_files.php](#).

`$vendorDir = dirname(__DIR__)`

Definition at line [5](#) of file [autoload_files.php](#).

autoload_files.php

[Go to the documentation of this file.](#)

```
00001 <?php
00002
00003 // autoload_files.php @generated by Composer
00004
00005 $vendorDir = dirname(__DIR__);
00006 $baseDir = dirname($vendorDir);
00007
00008 return array(
00009     '7b11c4dc42b3b3023073cb14e519683c' => $vendorDir .
00010     '/ralouphie/getallheaders/src/getallheaders.php',
00011     '6e3fae29631ef280660b3cdad06f25a8' => $vendorDir .
00012     '/symfony/deprecation-contracts/function.php',
00013     '37a3dc5111fe8f707ab4c132ef1dbc62' => $vendorDir .
00014     '/guzzlehttp/guzzle/src/functions_include.php',
00015 );
```

```
00012 );
```

vendor/composer/autoload_namespaces.php File Reference

Variables

- [\\$vendorDir](#) = dirname(__DIR__)
- [\\$baseDir](#) = dirname(\$vendorDir)

Variable Documentation

`$baseDir = dirname($vendorDir)`

Definition at line [6](#) of file [autoload_namespaces.php](#).

`$vendorDir = dirname(__DIR__)`

Definition at line [5](#) of file [autoload_namespaces.php](#).

autoload_namespaces.php

[Go to the documentation of this file.](#)

```
00001 <?php
00002
00003 // autoload_namespaces.php @generated by Composer
00004
00005 \$vendorDir = dirname(__DIR__);
00006 \$baseDir = dirname(\$vendorDir);
00007
00008 return array(
00009 );
```

vendor/composer/autoload_psr4.php File Reference

Variables

- [\\$vendorDir](#) = dirname(__DIR__)
- [\\$baseDir](#) = dirname(\$vendorDir)

Variable Documentation

`$baseDir = dirname($vendorDir)`

Definition at line [6](#) of file [autoload_psr4.php](#).

\$vendorDir = dirname(__DIR__)

Definition at line [5](#) of file [autoload_psr4.php](#).

autoload_psr4.php

[Go to the documentation of this file.](#)

```
00001 <?php
00002
00003 // autoload_psr4.php @generated by Composer
00004
00005 $vendorDir = dirname(__DIR__);
00006 $baseDir = dirname($vendorDir);
00007
00008 return array(
00009     'Stripe\\' => array($vendorDir . '/stripe/stripe-php/lib'),
00010     'Psr\\Http\\Message\\' => array($vendorDir . '/psr/http-factory/src',
00011     $vendorDir . '/psr/http-message/src'),
00012     'Psr\\Http\\Client\\' => array($vendorDir . '/psr/http-client/src'),
00013     'GuzzleHttp\\Psr7\\' => array($vendorDir . '/guzzlehttp/psr7/src'),
00014     'GuzzleHttp\\Promise\\' => array($vendorDir . '/guzzlehttp/promises/src'),
00015     'GuzzleHttp\\' => array($vendorDir . '/guzzlehttp/guzzle/src'),
00016 );
```

vendor/composer/autoload_real.php File Reference

Data Structures

class [ComposerAutoloaderInit4b55c7ef33235a0353c1b6b079f86964](#)

autoload_real.php

[Go to the documentation of this file.](#)

```
00001 <?php
00002
00003 // autoload_real.php @generated by Composer
00004
00005 class ComposerAutoloaderInit4b55c7ef33235a0353c1b6b079f86964
00006 {
00007     private static $loader;
00008
00009     public static function loadClassLoader($class)
00010     {
00011         if ('Composer\Autoload\ClassLoader' === $class) {
00012             require __DIR__ . '/ClassLoader.php';
00013         }
00014     }
00015
00016     public static function getLoader()
00017     {
00018         if (null !== self::$loader) {
00019             return self::$loader;
00020         }
00021
00022         require __DIR__ . '/platform_check.php';
00023     }
00024 }
```

```

00027 spl_autoload_register(array('ComposerAutoloaderInit4b55c7ef33235a0353c1b6b079f86964',
'loadClassLoader'), true, true);
00028     self::$loader = $loader = new
\Composer\Autoload\ClassLoader(\dirname(__DIR__));
00029 spl_autoload_unregister(array('ComposerAutoloaderInit4b55c7ef33235a0353c1b6b079f86964'
, 'loadClassLoader'));
00030
00031     require __DIR__ . '/autoload_static.php';
00032
call_user_func(\Composer\Autoload\ComposerStaticInit4b55c7ef33235a0353c1b6b079f86964::
getInitializer($loader));
00033
00034     $loader->register(true);
00035
00036     $filesToLoad =
\Composer\Autoload\ComposerStaticInit4b55c7ef33235a0353c1b6b079f86964::$files;
00037     $requireFile = \Closure::bind(static function ($fileIdentifier, $file) {
00038         if (empty($GLOBALS['__composer_autoload_files'][$fileIdentifier])) {
00039             $GLOBALS['__composer_autoload_files'][$fileIdentifier] = true;
00040
00041             require $file;
00042         }
00043     }, null, null);
00044     foreach ($filesToLoad as $fileIdentifier => $file) {
00045         $requireFile($fileIdentifier, $file);
00046     }
00047
00048     return $loader;
00049 }
00050 }

```

vendor/composer/autoload_static.php File Reference

Data Structures

class

[ComposerStaticInit4b55c7ef33235a0353c1b6b079f86964](#)Namespaces

- namespace [Composer\Autoload](#)

autoload_static.php

[Go to the documentation of this file.](#)

```

00001 <?php
00002
00003 // autoload_static.php @generated by Composer
00004
00005 namespace Composer\Autoload;
00006
00007 class ComposerStaticInit4b55c7ef33235a0353c1b6b079f86964
00008 {
00009     public static $files = array (
00010         '7b11c4dc42b3b3023073cb14e519683c' => __DIR__ . '/../.. ' .
'/ralouphie/getallheaders/src/getallheaders.php',
00011         '6e3fae29631ef280660b3cdad06f25a8' => __DIR__ . '/../.. ' .
'/symfony/deprecation-contracts/function.php',
00012         '37a3dc5111fe8f707ab4c132ef1dbc62' => __DIR__ . '/../.. ' .
'/guzzlehttp/guzzle/src/functions_include.php',
00013     );
00014
00015     public static $prefixLengthsPsr4 = array (
00016         'S' =>

```

```

00017     array (
00018         'Stripe\\' => 7,
00019     ),
00020     'P' =>
00021     array (
00022         'Psr\\Http\\Message\\' => 17,
00023         'Psr\\Http\\Client\\' => 16,
00024     ),
00025     'G' =>
00026     array (
00027         'GuzzleHttp\\Psr7\\' => 16,
00028         'GuzzleHttp\\Promise\\' => 19,
00029         'GuzzleHttp\\' => 11,
00030     ),
00031 );
00032
00033 public static $prefixDirsPsr4 = array (
00034     'Stripe\\' =>
00035     array (
00036         0 => __DIR__ . '/../' . '/stripe/stripe-php/lib',
00037     ),
00038     'Psr\\Http\\Message\\' =>
00039     array (
00040         0 => __DIR__ . '/../' . '/psr/http-factory/src',
00041         1 => __DIR__ . '/../' . '/psr/http-message/src',
00042     ),
00043     'Psr\\Http\\Client\\' =>
00044     array (
00045         0 => __DIR__ . '/../' . '/psr/http-client/src',
00046     ),
00047     'GuzzleHttp\\Psr7\\' =>
00048     array (
00049         0 => __DIR__ . '/../' . '/guzzlehttp/psr7/src',
00050     ),
00051     'GuzzleHttp\\Promise\\' =>
00052     array (
00053         0 => __DIR__ . '/../' . '/guzzlehttp/promises/src',
00054     ),
00055     'GuzzleHttp\\' =>
00056     array (
00057         0 => __DIR__ . '/../' . '/guzzlehttp/guzzle/src',
00058     ),
00059 );
00060
00061 public static $classMap = array (
00062     'Composer\\InstalledVersions' => __DIR__ . '/../' .
00063     '/composer/InstalledVersions.php',
00064 );
00065 public static function getInitializer(ClassLoader $loader)
00066 {
00067     return \Closure::bind(function () use ($loader) {
00068         $loader->prefixLengthsPsr4 =
00069         ComposerStaticInit4b55c7ef33235a0353clb6b079f86964::\$prefixLengthsPsr4;
00069         $loader->prefixDirsPsr4 =
00070         ComposerStaticInit4b55c7ef33235a0353clb6b079f86964::\$prefixDirsPsr4;
00070         $loader->classMap =
00071         ComposerStaticInit4b55c7ef33235a0353clb6b079f86964::\$classMap;
00071     }, null, ClassLoader::class);
00072 }
00073 }
00074 }

```

vendor/composer/ClassLoader.php File Reference

Data Structures

class [ClassLoader](#)Namespaces

- namespace [Composer\Autoload](#)

ClassLoader.php

[Go to the documentation of this file.](#)

```
00001 <?php
00002
00003 /*
00004  * This file is part of Composer.
00005  *
00006  * (c) Nils Adermann <naderman@naderman.de>
00007  *      Jordi Boggiano <j.boggiano@seld.be>
00008  *
00009  * For the full copyright and license information, please view the LICENSE
00010  * file that was distributed with this source code.
00011  */
00012
00013 namespace Composer\Autoload;
00014
00043 class ClassLoader
00044 {
00045     private static $includeFile;
00046
00047     private $vendorDir;
00048
00049     // PSR-4
00050     private $prefixLengthsPsr4 = array();
00051     private $prefixDirsPsr4 = array();
00052     private $fallbackDirsPsr4 = array();
00053
00054     // PSR-0
00055     private $prefixesPsr0 = array();
00056     private $fallbackDirsPsr0 = array();
00057
00058     private $useIncludePath = false;
00059
00060     private $classMap = array();
00061
00062     private $classMapAuthoritative = false;
00063
00064     private $missingClasses = array();
00065
00066     private $apcuPrefix;
00067
00068     private static $registeredLoaders = array();
00069
00070     public function __construct($vendorDir = null)
00071     {
00072         $this->vendorDir = $vendorDir;
00073         self::initializeIncludeClosure();
00074     }
00075
00076     public function getPrefixes()
00077     {
00078         if (!empty($this->prefixesPsr0)) {
00079             return call_user_func_array('array_merge',
00080                 array_values($this->prefixesPsr0));
00081         }
00082
00083         return array();
00084     }
00085
00086     public function getPrefixesPsr4()
00087     {
00088         return $this->prefixDirsPsr4;
00089     }
00090
00091     public function getFallbackDirs()
00092     {
00093         return $this->fallbackDirsPsr0;
00094     }
00095 }
```

```

00143 public function getFallbackDirsPsr4()
00144 {
00145     return $this->fallbackDirsPsr4;
00146 }
00147
00151 public function getClassMap()
00152 {
00153     return $this->classMap;
00154 }
00155
00161 public function addClassMap(array $classMap)
00162 {
00163     if ($this->classMap) {
00164         $this->classMap = array_merge($this->classMap, $classMap);
00165     } else {
00166         $this->classMap = $classMap;
00167     }
00168 }
00169
00180 public function add($prefix, $paths, $prepend = false)
00181 {
00182     $paths = (array) $paths;
00183     if (!$prefix) {
00184         if ($prepend) {
00185             $this->fallbackDirsPsr0 = array_merge(
00186                 $paths,
00187                 $this->fallbackDirsPsr0
00188             );
00189         } else {
00190             $this->fallbackDirsPsr0 = array_merge(
00191                 $this->fallbackDirsPsr0,
00192                 $paths
00193             );
00194         }
00195     }
00196     return;
00197 }
00198
00199 $first = $prefix[0];
00200 if (!isset($this->prefixesPsr0[$first][$prefix])) {
00201     $this->prefixesPsr0[$first][$prefix] = $paths;
00202 }
00203 return;
00204 }
00205 if ($prepend) {
00206     $this->prefixesPsr0[$first][$prefix] = array_merge(
00207         $paths,
00208         $this->prefixesPsr0[$first][$prefix]
00209     );
00210 } else {
00211     $this->prefixesPsr0[$first][$prefix] = array_merge(
00212         $this->prefixesPsr0[$first][$prefix],
00213         $paths
00214     );
00215 }
00216 }
00217
00230 public function addPsr4($prefix, $paths, $prepend = false)
00231 {
00232     $paths = (array) $paths;
00233     if (!$prefix) {
00234         // Register directories for the root namespace.
00235         if ($prepend) {
00236             $this->fallbackDirsPsr4 = array_merge(
00237                 $paths,
00238                 $this->fallbackDirsPsr4
00239             );
00240         } else {
00241             $this->fallbackDirsPsr4 = array_merge(
00242                 $this->fallbackDirsPsr4,
00243                 $paths
00244             );
00245         }
00246     } elseif (!isset($this->prefixDirsPsr4[$prefix])) {
00247         // Register directories for a new namespace.
00248         $length = strlen($prefix);
00249         if ('\\' !== $prefix[$length - 1]) {

```

```

00250         throw new \InvalidArgumentException("A non-empty PSR-4 prefix must
end with a namespace separator.");
00251     }
00252     $this->prefixLengthsPsr4[$prefix[0]][$prefix] = $length;
00253     $this->prefixDirsPsr4[$prefix] = $paths;
00254 } elseif ($prepend) {
00255     // Prepend directories for an already registered namespace.
00256     $this->prefixDirsPsr4[$prefix] = array_merge(
00257         $paths,
00258         $this->prefixDirsPsr4[$prefix]
00259     );
00260 } else {
00261     // Append directories for an already registered namespace.
00262     $this->prefixDirsPsr4[$prefix] = array_merge(
00263         $this->prefixDirsPsr4[$prefix],
00264         $paths
00265     );
00266 }
00267 }
00268
00278 public function set($prefix, $paths)
00279 {
00280     if (!$prefix) {
00281         $this->fallbackDirsPsr0 = (array) $paths;
00282     } else {
00283         $this->prefixesPsr0[$prefix[0]][$prefix] = (array) $paths;
00284     }
00285 }
00286
00298 public function setPsr4($prefix, $paths)
00299 {
00300     if (!$prefix) {
00301         $this->fallbackDirsPsr4 = (array) $paths;
00302     } else {
00303         $length = strlen($prefix);
00304         if ('\\' !== $prefix[$length - 1]) {
00305             throw new \InvalidArgumentException("A non-empty PSR-4 prefix must
end with a namespace separator.");
00306         }
00307         $this->prefixLengthsPsr4[$prefix[0]][$prefix] = $length;
00308         $this->prefixDirsPsr4[$prefix] = (array) $paths;
00309     }
00310 }
00311
00319 public function setUseIncludePath($useIncludePath)
00320 {
00321     $this->useIncludePath = $useIncludePath;
00322 }
00323
00330 public function getUseIncludePath()
00331 {
00332     return $this->useIncludePath;
00333 }
00334
00343 public function setClassMapAuthoritative($classMapAuthoritative)
00344 {
00345     $this->classMapAuthoritative = $classMapAuthoritative;
00346 }
00347
00353 public function isClassMapAuthoritative()
00354 {
00355     return $this->classMapAuthoritative;
00356 }
00357
00365 public function setApcuPrefix($apcuPrefix)
00366 {
00367     $this->apcuPrefix = function_exists('apcu fetch') &&
filter_var(ini_get('apc.enabled'), FILTER_VALIDATE_BOOLEAN) ? $apcuPrefix : null;
00368 }
00369
00375 public function getApcuPrefix()
00376 {
00377     return $this->apcuPrefix;
00378 }
00379
00387 public function register($prepend = false)
00388 {

```

```

00389         spl_autoload_register(array($this, 'loadClass'), true, $prepend);
00390
00391         if (null === $this->vendorDir) {
00392             return;
00393         }
00394
00395         if ($prepend) {
00396             self::$registeredLoaders = array($this->vendorDir => $this) +
self::$registeredLoaders;
00397         } else {
00398             unset(self::$registeredLoaders[$this->vendorDir]);
00399             self::$registeredLoaders[$this->vendorDir] = $this;
00400         }
00401     }
00402
00408     public function unregister()
00409     {
00410         spl_autoload_unregister(array($this, 'loadClass'));
00411
00412         if (null !== $this->vendorDir) {
00413             unset(self::$registeredLoaders[$this->vendorDir]);
00414         }
00415     }
00416
00423     public function loadClass($class)
00424     {
00425         if ($file = $this->findFile($class)) {
00426             $includeFile = self::$includeFile;
00427             $includeFile($file);
00428
00429             return true;
00430         }
00431
00432         return null;
00433     }
00434
00442     public function findFile($class)
00443     {
00444         // class map lookup
00445         if (isset($this->classMap[$class])) {
00446             return $this->classMap[$class];
00447         }
00448         if ($this->classMapAuthoritative || isset($this->missingClasses[$class]))
{
00449             return false;
00450         }
00451         if (null !== $this->apcuPrefix) {
00452             $file = apcu_fetch($this->apcuPrefix.$class, $hit);
00453             if ($hit) {
00454                 return $file;
00455             }
00456         }
00457
00458         $file = $this->findFileWithExtension($class, '.php');
00459
00460         // Search for Hack files if we are running on HHVM
00461         if (false === $file && defined('HHVM_VERSION')) {
00462             $file = $this->findFileWithExtension($class, '.hh');
00463         }
00464
00465         if (null !== $this->apcuPrefix) {
00466             apcu_add($this->apcuPrefix.$class, $file);
00467         }
00468
00469         if (false === $file) {
00470             // Remember that this class does not exist.
00471             $this->missingClasses[$class] = true;
00472         }
00473
00474         return $file;
00475     }
00476
00482     public static function getRegisteredLoaders()
00483     {
00484         return self::$registeredLoaders;
00485     }
00486

```

```

00492     private function findFileWithExtension($class, $ext)
00493     {
00494         // PSR-4 lookup
00495         $logicalPathPsr4 = strtr($class, '\\', DIRECTORY_SEPARATOR) . $ext;
00496
00497         $first = $class[0];
00498         if (isset($this->prefixLengthsPsr4[$first])) {
00499             $subPath = $class;
00500             while (false !== $lastPos = strrpos($subPath, '\\')) {
00501                 $subPath = substr($subPath, 0, $lastPos);
00502                 $search = $subPath . '\\';
00503                 if (isset($this->prefixDirsPsr4[$search])) {
00504                     $pathEnd = DIRECTORY_SEPARATOR . substr($logicalPathPsr4,
00505 $lastPos + 1);
00506                     foreach ($this->prefixDirsPsr4[$search] as $dir) {
00507                         if (file_exists($file = $dir . $pathEnd)) {
00508                             return $file;
00509                         }
00510                     }
00511                 }
00512             }
00513
00514             // PSR-4 fallback dirs
00515             foreach ($this->fallbackDirsPsr4 as $dir) {
00516                 if (file_exists($file = $dir . DIRECTORY_SEPARATOR . $logicalPathPsr4))
00517                 {
00518                     return $file;
00519                 }
00520             }
00521
00522             // PSR-0 lookup
00523             if (false !== $pos = strrpos($class, '\\')) {
00524                 // namespaced class name
00525                 $logicalPathPsr0 = substr($logicalPathPsr4, 0, $pos + 1)
00526                     . strtr(substr($logicalPathPsr4, $pos + 1), '_',
00527 DIRECTORY_SEPARATOR);
00528             } else {
00529                 // PEAR-like class name
00530                 $logicalPathPsr0 = strtr($class, '_', DIRECTORY_SEPARATOR) . $ext;
00531             }
00532             if (isset($this->prefixesPsr0[$first])) {
00533                 foreach ($this->prefixesPsr0[$first] as $prefix => $dirs) {
00534                     if (0 === strpos($class, $prefix)) {
00535                         foreach ($dirs as $dir) {
00536                             if (file_exists($file = $dir . DIRECTORY_SEPARATOR .
00537 $logicalPathPsr0)) {
00538                                 return $file;
00539                             }
00540                         }
00541                     }
00542                 }
00543
00544                 // PSR-0 fallback dirs
00545                 foreach ($this->fallbackDirsPsr0 as $dir) {
00546                     if (file_exists($file = $dir . DIRECTORY_SEPARATOR . $logicalPathPsr0))
00547                     {
00548                         return $file;
00549                     }
00550                 }
00551
00552                 // PSR-0 include paths.
00553                 if ($this->useIncludePath && $file =
00554 stream_resolve_include_path($logicalPathPsr0)) {
00555                     return $file;
00556                 }
00557
00558                 return false;
00559             }
00560
00561             private static function initializeIncludeClosure()
00562             {
00563                 if (self::$includeFile !== null) {
00564                     return;
00565                 }

```



```

00566
00575     self::$includeFile = \Closure::bind(static function($file) {
00576         include $file;
00577     }, null, null);
00578 }
00579 }

```

vendor/composer/installed.php File Reference

installed.php

[Go to the documentation of this file.](#)

```

00001 <?php return array(
00002     'root' => array(
00003         'name' => '__root__',
00004         'pretty_version' => 'dev-master',
00005         'version' => 'dev-master',
00006         'reference' => '66d8380a1e731ae7f6abb7df4897af5902cf98e8',
00007         'type' => 'library',
00008         'install_path' => __DIR__ . '/../../',
00009         'aliases' => array(),
00010         'dev' => true,
00011     ),
00012     'versions' => array(
00013         '__root__' => array(
00014             'pretty_version' => 'dev-master',
00015             'version' => 'dev-master',
00016             'reference' => '66d8380a1e731ae7f6abb7df4897af5902cf98e8',
00017             'type' => 'library',
00018             'install_path' => __DIR__ . '/../../',
00019             'aliases' => array(),
00020             'dev_requirement' => false,
00021         ),
00022         'guzzlehttp/guzzle' => array(
00023             'pretty_version' => '7.8.0',
00024             'version' => '7.8.0.0',
00025             'reference' => '1110f66a6530a40fe7aea0378fe608ee2b2248f9',
00026             'type' => 'library',
00027             'install_path' => __DIR__ . '/../guzzlehttp/guzzle',
00028             'aliases' => array(),
00029             'dev_requirement' => false,
00030         ),
00031         'guzzlehttp/promises' => array(
00032             'pretty_version' => '2.0.1',
00033             'version' => '2.0.1.0',
00034             'reference' => '111166291a0f8130081195ac4556a5587d7f1b5d',
00035             'type' => 'library',
00036             'install_path' => __DIR__ . '/../guzzlehttp/promises',
00037             'aliases' => array(),
00038             'dev_requirement' => false,
00039         ),
00040         'guzzlehttp/psr7' => array(
00041             'pretty_version' => '2.6.1',
00042             'version' => '2.6.1.0',
00043             'reference' => 'be45764272e8873c72dbe3d2edcfdcc3bc9f727',
00044             'type' => 'library',
00045             'install_path' => __DIR__ . '/../guzzlehttp/psr7',
00046             'aliases' => array(),
00047             'dev_requirement' => false,
00048         ),
00049         'psr/http-client' => array(
00050             'pretty_version' => '1.0.3',
00051             'version' => '1.0.3.0',
00052             'reference' => 'bb5906edc1c324c9a05aa0873d40117941e5fa90',
00053             'type' => 'library',
00054             'install_path' => __DIR__ . '/../psr/http-client',

```

```

00055         'aliases' => array(),
00056         'dev_requirement' => false,
00057     ),
00058     'psr/http-client-implementation' => array(
00059         'dev_requirement' => false,
00060         'provided' => array(
00061             0 => '1.0',
00062         ),
00063     ),
00064     'psr/http-factory' => array(
00065         'pretty_version' => '1.0.2',
00066         'version' => '1.0.2.0',
00067         'reference' => 'e616d01114759c4c489f93b099585439f795fe35',
00068         'type' => 'library',
00069         'install_path' => __DIR__ . '/../psr/http-factory',
00070         'aliases' => array(),
00071         'dev_requirement' => false,
00072     ),
00073     'psr/http-factory-implementation' => array(
00074         'dev_requirement' => false,
00075         'provided' => array(
00076             0 => '1.0',
00077         ),
00078     ),
00079     'psr/http-message' => array(
00080         'pretty_version' => '2.0',
00081         'version' => '2.0.0.0',
00082         'reference' => '402d35bcb92c70c026d1a6a9883f06b2ead23d71',
00083         'type' => 'library',
00084         'install_path' => __DIR__ . '/../psr/http-message',
00085         'aliases' => array(),
00086         'dev_requirement' => false,
00087     ),
00088     'psr/http-message-implementation' => array(
00089         'dev_requirement' => false,
00090         'provided' => array(
00091             0 => '1.0',
00092         ),
00093     ),
00094     'ralouphie/getallheaders' => array(
00095         'pretty_version' => '3.0.3',
00096         'version' => '3.0.3.0',
00097         'reference' => '120b605dfefb996808c31b6477290a714d356e822',
00098         'type' => 'library',
00099         'install_path' => __DIR__ . '/../ralouphie/getallheaders',
00100         'aliases' => array(),
00101         'dev_requirement' => false,
00102     ),
00103     'stripe/stripe-php' => array(
00104         'pretty_version' => 'v12.6.0',
00105         'version' => '12.6.0.0',
00106         'reference' => 'e0c15e4cbf252e708b937482bb1a50fa7e01bc9d',
00107         'type' => 'library',
00108         'install_path' => __DIR__ . '/../stripe/stripe-php',
00109         'aliases' => array(),
00110         'dev_requirement' => false,
00111     ),
00112     'symfony/deprecation-contracts' => array(
00113         'pretty_version' => 'v3.3.0',
00114         'version' => '3.3.0.0',
00115         'reference' => '7c3aff79d10325257a001fcf92d991f24fc967cf',
00116         'type' => 'library',
00117         'install_path' => __DIR__ . '/../symfony/deprecation-contracts',
00118         'aliases' => array(),
00119         'dev_requirement' => false,
00120     ),
00121 ),
00122 );

```

vendor/composer/InstalledVersions.php File Reference

Data Structures

class [InstalledVersions](#)Namespaces

- namespace [Composer](#)

InstalledVersions.php

[Go to the documentation of this file.](#)

```
00001 <?php
00002
00003 /*
00004  * This file is part of Composer.
00005  *
00006  * (c) Nils Adermann <naderman@naderman.de>
00007  *      Jordi Boggiano <j.boggiano@seld.be>
00008  *
00009  * For the full copyright and license information, please view the LICENSE
00010  * file that was distributed with this source code.
00011  */
00012
00013 namespace Composer;
00014
00015 use Composer\Autoload\ClassLoader;
00016 use Composer\Semver\VersionParser;
00017
00027 class InstalledVersions
00028 {
00033     private static $installed;
00034
00038     private static $scanGetVendors;
00039
00044     private static $installedByVendor = array();
00045
00052     public static function getInstalledPackages()
00053     {
00054         $packages = array();
00055         foreach (self::getInstalled() as $installed) {
00056             $packages[] = array_keys($installed['versions']);
00057         }
00058
00059         if (1 === \count($packages)) {
00060             return $packages[0];
00061         }
00062
00063         return array_keys(array_flip(\call_user_func_array('array_merge',
$packages)));
00064     }
00065
00073     public static function getInstalledPackagesByType($type)
00074     {
00075         $packagesByType = array();
00076
00077         foreach (self::getInstalled() as $installed) {
00078             foreach ($installed['versions'] as $name => $package) {
00079                 if (isset($package['type']) && $package['type'] === $type) {
00080                     $packagesByType[] = $name;
00081                 }
00082             }
00083         }
00084
00085         return $packagesByType;
00086     }
00087
00097     public static function isInstalled($packageName, $includeDevRequirements =
true)
```

```

00098     {
00099         foreach (self::getInstalled() as $installed) {
00100             if (isset($installed['versions'][$packageName])) {
00101                 return $includeDevRequirements ||
!isset($installed['versions'][$packageName]['dev_requirement']) ||
$installed['versions'][$packageName]['dev_requirement'] === false;
00102             }
00103         }
00104
00105         return false;
00106     }
00107
00120     public static function satisfies(VersionParser $parser, $packageName,
$constraint)
00121     {
00122         $constraint = $parser->parseConstraints((string) $constraint);
00123         $provided =
$parser->parseConstraints(self::getVersionRanges($packageName));
00124
00125         return $provided->matches($constraint);
00126     }
00127
00137     public static function getVersionRanges($packageName)
00138     {
00139         foreach (self::getInstalled() as $installed) {
00140             if (!isset($installed['versions'][$packageName])) {
00141                 continue;
00142             }
00143
00144             $ranges = array();
00145             if (isset($installed['versions'][$packageName]['pretty_version'])) {
00146                 $ranges[] =
$installed['versions'][$packageName]['pretty_version'];
00147             }
00148             if (array_key_exists('aliases',
$installed['versions'][$packageName])) {
00149                 $ranges = array_merge($ranges,
$installed['versions'][$packageName]['aliases']);
00150             }
00151             if (array_key_exists('replaced',
$installed['versions'][$packageName])) {
00152                 $ranges = array_merge($ranges,
$installed['versions'][$packageName]['replaced']);
00153             }
00154             if (array_key_exists('provided',
$installed['versions'][$packageName])) {
00155                 $ranges = array_merge($ranges,
$installed['versions'][$packageName]['provided']);
00156             }
00157
00158             return implode(' || ', $ranges);
00159         }
00160
00161         throw new \OutOfBoundsException('Package "' . $packageName . '" is not
installed');
00162     }
00163
00168     public static function getVersion($packageName)
00169     {
00170         foreach (self::getInstalled() as $installed) {
00171             if (!isset($installed['versions'][$packageName])) {
00172                 continue;
00173             }
00174
00175             if (!isset($installed['versions'][$packageName]['version'])) {
00176                 return null;
00177             }
00178
00179             return $installed['versions'][$packageName]['version'];
00180         }
00181
00182         throw new \OutOfBoundsException('Package "' . $packageName . '" is not
installed');
00183     }
00184
00189     public static function getPrettyVersion($packageName)
00190     {

```

```

00191         foreach (self::getInstalled() as $installed) {
00192             if (!isset($installed['versions'][$packageName])) {
00193                 continue;
00194             }
00195
00196             if (!isset($installed['versions'][$packageName]['pretty_version'])) {
00197                 return null;
00198             }
00199
00200             return $installed['versions'][$packageName]['pretty_version'];
00201         }
00202
00203         throw new \OutOfBoundsException('Package "' . $packageName . '" is not
00204         installed');
00205     }
00206
00210     public static function getReference($packageName)
00211     {
00212         foreach (self::getInstalled() as $installed) {
00213             if (!isset($installed['versions'][$packageName])) {
00214                 continue;
00215             }
00216
00217             if (!isset($installed['versions'][$packageName]['reference'])) {
00218                 return null;
00219             }
00220
00221             return $installed['versions'][$packageName]['reference'];
00222         }
00223
00224         throw new \OutOfBoundsException('Package "' . $packageName . '" is not
00225         installed');
00226     }
00231     public static function getInstallPath($packageName)
00232     {
00233         foreach (self::getInstalled() as $installed) {
00234             if (!isset($installed['versions'][$packageName])) {
00235                 continue;
00236             }
00237
00238             return isset($installed['versions'][$packageName]['install_path']) ?
00239             $installed['versions'][$packageName]['install_path'] : null;
00240         }
00241
00242         throw new \OutOfBoundsException('Package "' . $packageName . '" is not
00243         installed');
00244     }
00248     public static function getRootPackage()
00249     {
00250         $installed = self::getInstalled();
00251
00252         return $installed[0]['root'];
00253     }
00254
00262     public static function getRawData()
00263     {
00264         @trigger_error('getRawData only returns the first dataset loaded, which may
00265         not be what you expect. Use getAllRawData() instead which returns all datasets for all
00266         autoloaders present in the process.', E_USER_DEPRECATED);
00267
00268         if (null === self::$installed) {
00269             // only require the installed.php file if this file is loaded from its
00270             // dumped location,
00271             // and not from its source location in the composer/composer package,
00272             // see https://github.com/composer/composer/issues/9937
00273             if (substr(__DIR__, -8, 1) !== 'C') {
00274                 self::$installed = include __DIR__ . '/installed.php';
00275             } else {
00276                 self::$installed = array();
00277             }
00278         }
00279
00280         return self::$installed;
00281     }

```

```

00285     public static function getAllRawData()
00286     {
00287         return self::getInstalled();
00288     }
00289
00308     public static function reload($data)
00309     {
00310         self::$installed = $data;
00311         self::$installedByVendor = array();
00312     }
00313
00318     private static function getInstalled()
00319     {
00320         if (null === self::$scanGetVendors) {
00321             self::$scanGetVendors = method_exists('Composer\Autoload\ClassLoader',
00322             'getRegisteredLoaders');
00323         }
00324         $installed = array();
00325
00326         if (self::$scanGetVendors) {
00327             foreach (ClassLoader::getRegisteredLoaders() as $vendorDir => $loader)
00328             {
00329                 if (isset(self::$installedByVendor[$vendorDir])) {
00330                     $installed[] = self::$installedByVendor[$vendorDir];
00331                 } elseif (is_file($vendorDir.'/composer/installed.php')) {
00332                     $required = require $vendorDir.'/composer/installed.php';
00333                     $installed[] = self::$installedByVendor[$vendorDir] =
00334                     $required;
00335                     if (null === self::$installed && strpos($vendorDir.'/composer',
00336                     '\\', '/') === strpos(__DIR__, '\\', '/')) {
00337                         self::$installed = $installed[count($installed) - 1];
00338                     }
00339                 }
00340             }
00341
00342             if (null === self::$installed) {
00343                 // only require the installed.php file if this file is loaded from its
00344                 // dumped location,
00345                 // and not from its source location in the composer/composer package,
00346                 // see https://github.com/composer/composer/issues/9937
00347                 if (substr(__DIR__, -8, 1) !== 'C') {
00348                     $required = require __DIR__.'/installed.php';
00349                     self::$installed = $required;
00350                 } else {
00351                     self::$installed = array();
00352                 }
00353             }
00354
00355             if (self::$installed !== array()) {
00356                 $installed[] = self::$installed;
00357             }
00358
00359             return $installed;
00360         }
00361     }

```

vendor/composer/platform_check.php File Reference

Variables

- [\\$issues](#) = array()
-

Variable Documentation

`$issues = array()`

Definition at line [5](#) of file [platform_check.php](#).

platform_check.php

[Go to the documentation of this file.](#)

```
00001 <?php
00002
00003 // platform_check.php @generated by Composer
00004
00005 \$issues = array();
00006
00007 if (!(PHP_VERSION_ID >= 80100)) {
00008     \$issues[] = 'Your Composer dependencies require a PHP version ">= 8.1.0". You
00009 are running ' . PHP_VERSION . '.';
00010 }
00011 if (\$issues) {
00012     if (!headers_sent()) {
00013         header('HTTP/1.1 500 Internal Server Error');
00014     }
00015     if (!ini_get('display_errors')) {
00016         if (PHP_SAPI === 'cli' || PHP_SAPI === 'phpdbg') {
00017             fwrite(STDERR, 'Composer detected issues in your platform:' .
PHP_EOL.PHP_EOL . implode(PHP_EOL, \$issues) . PHP_EOL.PHP_EOL);
00018         } elseif (!headers_sent()) {
00019             echo 'Composer detected issues in your platform:' . PHP_EOL.PHP_EOL .
str_replace('You are running '.PHP_VERSION.'.', '', implode(PHP_EOL, \$issues)) .
PHP_EOL.PHP_EOL;
00020         }
00021     }
00022     trigger_error(
00023         'Composer detected issues in your platform: ' . implode(' ', \$issues),
00024         E_USER_ERROR
00025     );
00026 }
```

Index

INDEX