GESTIÓN DE DEPENDENCIAS CON COMPOSER

¿Qué es un gestor de dependencias?

Un **gestor de dependencias** es una herramienta que se utiliza en el desarrollo de software para administrar las bibliotecas y paquetes de código que un proyecto necesita. Sobre todo es útil para bibliotecas desarrolladas por terceros y que se utilizan para añadir funcionalidades específicas a una aplicación.

Los **gestores de dependencias** ayudan a simplificar la gestión de estas bibliotecas, asegurando que se descarguen e instalen de manera correcta, gestionando las versiones, y resolviendo las dependencias entre ellas.

¿Qué es Composer?

Composer es un gestor de dependencias específico para PHP, y es ampliamente utilizado en el desarrollo de aplicaciones web en PHP.

Composer permite definir y gestionar las dependencias de un proyecto PHP de una manera sencilla. A través de un archivo llamado "**composer.json**", puedes especificar qué bibliotecas o paquetes de código necesita tu proyecto, así como las versiones específicas de esas dependencias. **Composer** se encargará de descargar las bibliotecas necesarias desde el repositorio de paquetes de PHP, que es Packagist, y las instalará en tu proyecto de manera automatizada.

Algunas ventajas de usar Composer incluyen:

1. **Gestión de dependencias**: Composer resuelve y descarga automáticamente las dependencias de tu proyecto, lo que simplifica la configuración de tu entorno de desarrollo.
2. **Versionamiento preciso**: Puedes especificar las versiones exactas de las bibliotecas que necesitas, lo que ayuda a garantizar la compatibilidad de las versiones y evita problemas potenciales.
3. **Actualizaciones sencillas**: Puedes actualizar fácilmente las bibliotecas a nuevas versiones con un simple comando, y Composer se encargará de resolver cualquier conflicto de dependencias.
4. **Autocarga de clases**: Composer genera un archivo de autocarga de clases que simplifica la inclusión de las clases de las bibliotecas en tu código.

Composer es una herramienta de gestión de dependencias para PHP que facilita la Incorporación y gestión de bibliotecas de terceros en tus proyectos PHP. Composer permite administrar las dependencias de tu proyecto y asegurarte de que todas las bibliotecas estén actualizadas y funcionando correctamente.

Instalación de Composer en Windows

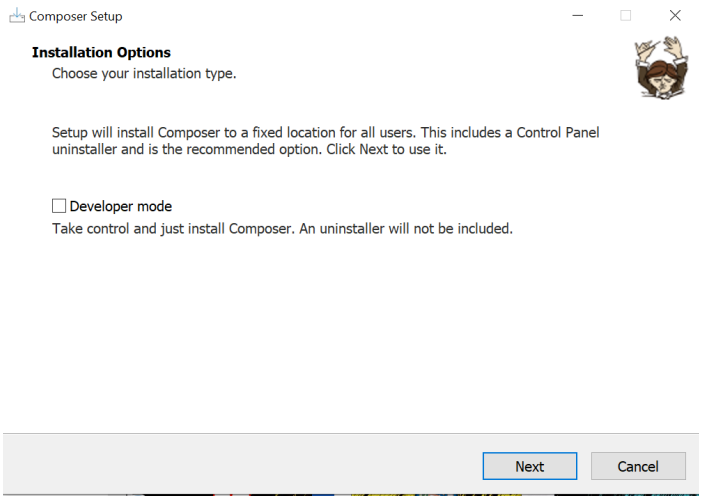
Paso 1: Descargar Composer

Para descargar Composer, visita **(https://getcomposer.org/download/)** y descarga el instalador de Composer para Windows. Deberías obtener un archivo llamado `Composer-Setup.exe`.

Este fichero te permitirá una instalación sencilla y automática.

Paso 2: Instalar Composer

Ejecuta el archivo `Composer-Setup.exe` que descargaste en el Paso 1. El instalador te guiará a través del proceso de instalación, que incluirá la configuración de PHP y Composer en tu sistema. Puedes elegir la instalación automática.

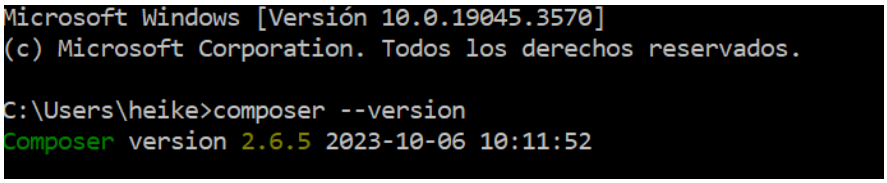


Paso 3: Comprobar la instalación

Para verificar que Composer se ha instalado correctamente, abre una terminal (cmd) y ejecuta el siguiente comando:

composer --version

Deberías ver la versión de Composer que has instalado.



Paso 4: Crear archivo json

Crea un directorio **ejemploComposer** dentro de tu directorio de trabajo de Servidor.

Navega hasta el directorio en la terminal y ejecuta

composer init

Este comando te guiará a través de la creación de un archivo composer.json donde definirás las dependencias de tu proyecto. Responde las preguntas que se te hagan, como el nombre de tu proyecto y la descripción.

1. **Package name**: El nombre de tu proyecto.
2. **Description**: Una breve descripción de tu proyecto.
3. **Author**: Tu nombre o el nombre de tu equipo.
4. **Minimum Stability**: Puedes dejarlo en "stable" a menos que tengas una razón específica para cambiarlo.
5. **Package Type**: Puedes dejarlo en "project" si estás configurando un proyecto desde cero.
6. **License**: Elige una licencia para tu proyecto. Puedes consultar choosealicense.com para obtener información sobre diferentes tipos de licencias.
7. **Define your dependencies**: Aquí puedes especificar las dependencias que necesitas para tu proyecto. Puedes dejarlo en blanco y luego añadir las dependencias directamente en el fichero json. Nosotros añadiremos de manera manual en el fichero composer.json las dependencias.

Paso 5: Añadir dependencias en composer.json

Edita el fichero composer.json y añade las dependencias en el apartado require.

En nuestro caso añadiremos la de la librería PHPMailer

```
{
   "name": "ejemploComposer",
   "description": "Ejemplo de uso de Composer con PHPMailer",
   "require": {
      "phpmailer/phpmailer": "^6.0"
   }
}
```

Paso 6: Haz que composer instale

Ahora, ejecuta el siguiente comando para que Composer descargue e instale las dependencias en tu proyecto:

composer install

Composer buscará las bibliotecas en el repositorio Packagist, descargará las versiones especificadas y las instalará en un directorio llamado "vendor" en tu proyecto.

Parte 2: Uso de Composer con PHPMailer

¿Qué es PHPMailer?

PHPMailer es una biblioteca de PHP que permite enviar correos electrónicos desde tu aplicación de forma sencilla. Se utiliza ampliamente en aplicaciones web para enviar correos electrónicos autenticados y personalizados.

Configuración de Gmail en 2023

Para configurar Gmail a día de hoy en Windows, sigue estos pasos:

1. Inicia sesión en tu cuenta de Gmail en tu navegador.

2. Accede a [Mi cuenta](https://myaccount.google.com/).

3. En la barra lateral izquierda, haz clic en "Seguridad".

4. En la sección "Acceso de aplicaciones menos seguras", activa la opción "Permitir el acceso de aplicaciones menos seguras". Esto permitirá que PHPMailer envíe correos electrónicos a través de tu cuenta de Gmail.

Configurar la autenticación de dos factores (si es necesario)

Si tienes habilitada la autenticación de dos factores en tu cuenta de Gmail, debes generar una contraseña de aplicación para usar con PHPMailer:

1. Visita [Mi cuenta](https://myaccount.google.com/).

2. En la barra lateral izquierda, haz clic en "Seguridad".

3. En la sección "Iniciar sesión en Google", haz clic en "Contraseña de aplicación".

4. Selecciona "Correo" como la aplicación y "Otra (nombre personalizado)" como el dispositivo.

5. Haz clic en "Generar".

6. Copia la contraseña generada y úsala como la contraseña en el código PHPMailer.

Con estos pasos, deberías poder enviar correos electrónicos desde tu aplicación PHP en Windows utilizando Gmail como servidor SMTP.

Uso de PHPMAILER

Ahora vemos el código que escribiremos para utilizar la clase PHPMAILER

require 'vendor/autoload.php';

Esta línea es importante porque carga el archivo de autoloading generado por Composer, que se encargará de cargar automáticamente las clases necesarias de las bibliotecas PHPMailer, así como de cualquier otra biblioteca que hayas definido en tu archivo `composer.json`.

use PHPMailer\PHPMailer\PHPMailer;

use PHPMailer\PHPMailer\SMTP;

use PHPMailer\PHPMailer\Exception;

Estas líneas importan las clases que necesitas de PHPMailer para enviar correos electrónicos. PHPMailer es una biblioteca muy versátil para enviar correos electrónicos en PHP.

// Crea una instancia de PHPMailer

$mail = new PHPMailer();

Aquí, se crea una instancia de la clase `PHPMailer` para trabajar con el envío de correos electrónicos.

try {

// Configura el servidor SMTP

$mail->isSMTP();

$mail->Host = 'tu_servidor_smtp';

$mail->SMTPAuth = true;

$mail->Username = 'tu_correo_electronico';

$mail->Password = 'tu_contraseña';

$mail->SMTPSecure = 'tls';

$mail->Port = 587;

En este bloque de código, se configuran los detalles del servidor SMTP que usarás para enviar el correo. Debes proporcionar la información específica de tu servidor SMTP, tu dirección de correo electrónico y tu contraseña en los campos correspondientes.

// Configura los destinatarios

$mail->setFrom('tucorreo@gmail.com', 'Tu Nombre');

$mail->addAddress('destinatario@example.com', 'Nombre del Destinatario');

Aquí se configuran los destinatarios del correo electrónico, especificando tanto la dirección de correo electrónico de origen como la del destinatario.

// Contenido del correo

$mail->isHTML(true);

$mail->Subject = 'Asunto del Correo';

$mail->Body = 'Este es el cuerpo del correo electrónico.';

En esta parte, se establece el contenido del correo. Puedes definir el asunto y el cuerpo del correo aquí. Además, se indica que el correo es en formato HTML (`$mail->isHTML(true)`), por lo que el contenido se interpretará como HTML.

```
    // Enviar el correo
    $mail->send();
    echo 'El correo se ha enviado con éxito.';
} catch (Exception $e) {
    echo "El correo no se pudo enviar. Error: {$mail->ErrorInfo}";
}
```

Finalmente, se intenta enviar el correo utilizando el método `send()` de la instancia de `PHPMailer`. Si el envío es exitoso, se muestra un mensaje de éxito. Si ocurre un error, se captura una excepción de tipo `Exception` y se muestra un mensaje de error que incluye detalles del error.

Este es un ejemplo de cómo utilizar PHPMailer para enviar correos electrónicos en PHP. Recuerda que debes configurar los detalles del servidor SMTP, la dirección de correo electrónico y la contraseña con la información adecuada para que funcione en tu caso particular.