

# Modern PHP Composer blog — Part 1

## Objectif

Créer un blog MVC avec Composer en plusieurs étapes.

## Interface

- Vous devez créer une page d'accueil `GET /`
- Vous devez créer une interface correcte (vous pouvez utiliser le framework de votre choix)
- Vous devez afficher des notifications en cas d'erreurs (voir liste des cas d'erreurs)

## Base de données

Vous devez créer une table `posts` qui contient les champs suivants :

Nom	Type	Description
id	int	Identifiant unique de l'utilisateur, généré automatiquement lors de l'insertion
created_at	timestamp	Date et heure à laquelle l'article a été créé, généré automatiquement lors de l'insertion
updated_at	timestamp	Date et heure à laquelle l'article a été mis à jour, généré automatiquement lors de l'insertion
title	varchar(255)	Titre de l'article
body	text	Contenu de l'article

## Routes des articles

Methode	URL	Action	Description
GET	/articles	postIndex()	Affichage de tous les articles
GET	/articles/show?id={id}	postShow(\$id)	Affichage d'un article (où {id} est l'id de l'article)
GET	/articles/create	postCreate()	Affichage du formulaire de création d'un article
POST	/articles	postStore()	Création d'un article
GET	/articles/edit?id={id}	postEdit(\$id)	Affichage du formulaire de modification d'un article
POST	/articles/edit	postUpdate()	Modification d'un article (où {id} est l'id de l'article)
POST	/articles/delete?id={id}	postDestroy(\$id)	Suppression d'un article (où {id} est l'id de l'article)

## Cas d'erreur

Fonctionnalité	Description
Affichage d'un article	<ul style="list-style-type: none"> <li>- Vérifier que le query parameter {id} existe et n'est pas vide, sinon renvoyer un erreur 400</li> <li>- Vérifier que l'article correspond au query parameter {id} existe, sinon renvoyer une erreur 404</li> </ul>
Création d'un article	<ul style="list-style-type: none"> <li>- Vérifier que les champs title et body existent et sont remplis, sinon renvoyer l'utilisateur sur le formulaire de création en gardant les informations qu'il a rentré et en affichant un message d'erreur</li> </ul>
Modification d'un article	<ul style="list-style-type: none"> <li>- Vérifier que l' {id} passé en paramètre corresponde bien à un article, sinon renvoyer une erreur 404</li> <li>- Vérifier que les champs title et body existent et sont remplis, sinon renvoyer l'utilisateur sur le formulaire de modification en gardant les informations qu'il a rentré et en affichant un message d'erreur</li> </ul>
Suppression d'un article	<ul style="list-style-type: none"> <li>- Vérifier que l' {id} passé en paramètre corresponde bien à un article, sinon renvoyer une erreur 404</li> </ul>

# Contraintes

- L'affichage d'un article doit afficher toutes les dates au format relatif ("Il y'a 3 jours") en utilisant [Carbon](#). Les dates sur les autres pages doivent être affichées au format français 
- Les données des articles doivent se trouver dans une base de données MySQL dont les paramètres de connexion sont stockés dans un fichier `.env` grâce à [PHP Dotenv](#)
- Gestion des erreurs avec [Whoops](#)