PARTIE 4

Réaliser un site web avec l'architecture MVC

Dans ce module, vous allez :

- Maitriser les scripts d'accès aux données
- Réaliser l'architecture MVC
- Créer une API RESTful

08 heures

Activité 1

Vers une architecture MVC

Compétences visées :

- Mettre en place l'architecture MVC
- Réaliser une API RESTful
- Mettre en place les connaissances PHP de base

Recommandations clés :

- Bonne révision du résumé théorique
- Lire attentivement les consignes des exercices

08 heures

CONSIGNES

1. Pour le formateur :

- Rappeler les bases théoriques sur l’architecture MVC

2. Pour l’apprenant :

- Utiliser le résumé théorique
- Maitriser les notions de MVC
- Utiliser un paramètre comme jeton d’authentification

3. Conditions de réalisation :

- Base de donnée MySQL
- L'extension PHP Data Objects (PDO)
- Structure de base de MVC

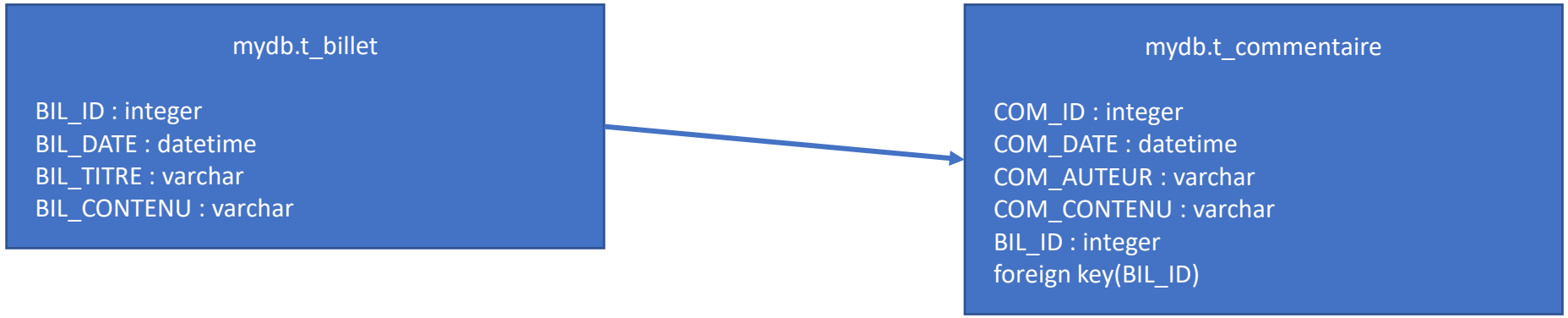
4. Critères de réussite :

- Le stagiaire est-il capable de :
  - Établir une connexion avec MySQL
  - Écrire, lire et supprimer les données de la base de données
  - Utiliser les notions MVC
  - Créer une API RESTful

Activité 1
Vers une architecture MVC

Exercice

1. Créer une base de données qui se compose de deux tables, l'une stockant les billets (articles) du blog et l'autre les commentaires associés aux articles.



PARTIE 4

Exercice

2. Insérer des données pour avoir le résultat suivant:

	BIL_ID	BIL_DATE	BIL_TITRE	BIL_CONTENU
▶	1	2022-02-21 00:25:42	Premier billet	Bonjour monde ! Ceci est le premier billet sur mon blog.
	2	2022-02-21 00:25:42	Au travail	Il faut enrichir ce blog dès maintenant.

Fig. : Données de la table t_billet

	COM_ID	COM_DATE	COM_AUTEUR	COM_CONTENU	BIL_ID
▶	1	2022-02-21 00:25:42	A. Nonyme	Bravo pour ce début	1
	2	2022-02-21 00:25:42	Moi	Merci ! Je vais continuer sur ma lancée	1

Fig. : Données de la table t_commentaire

Exercice

3. Créer un répertoire Modele qui contient le code d'accès aux données :
    - Modele.php : Classe abstraite Modèle. Centralise les services d'accès à une base de données et utilise l'API PDO.
    - Billet.php : Permet d'afficher les informations demandées de la table t_billet. Contient deux fonctions:
        - getBillets() : Renvoie la liste des billets du blog.
        - getBillet($idBillet) : Renvoie les informations sur un billet.
    - Commentaire.php : Une classe sur le même modèle que la classe Billet. Contient deux fonctions:
        - getCommentaires($idBillet) : Renvoie la liste des commentaires associés à un billet.
        - ajouterCommentaire($auteur, $contenu, $idBillet) : Ajoute un commentaire dans la base de données.

Exercice

4. Créer un répertoire Vue regroupant le code d'affichage :

- gabarit.php : (template en anglais) Ce modèle contiendra tous les éléments communs et permettra d'ajouter les éléments spécifiques à chaque vue.
- vueAccueil.php : Permet d'afficher la l'ensemble des titres billets (lien qui renvoie à la vueBillet.php) avec leurs date.
- vueBillet.php : Permet d'afficher l'ensemble des commentaires du Billet sélectionné. Permet d'ajouter un nouveau commentaire pour ce billet.
- vueErreur.php : Permet d'afficher un message d'erreur. Exemples :
  - http://localhost:122/index.php?action=billet&id=UIF => Affiche : Identifiant de billet non valide (toujours avec l'entête et le pied de la page)
  - http://localhost:122/index.php?action=billet => Affiche : Paramètre 'id' absent (toujours avec l'entête et le pied de la page)
- Vue.php : La classe Vue dont le rôle sera de gérer la génération des vues.

Exercice

5. Créer un répertoire Controleur regroupant le code d'affichage :

- ControleurAccueil.php : Est une classe qui permet de gérer l'accueil. Contient deux fonctions:
  - __construct() : Crée un nouveau objet Billet
  - accueil() : Affiche la liste de tous les billets du blog
- ControleurBillet.php : Permet de gérer l'affichage d'un billet. Contient deux fonctions:
  - __construct() : Crée deux nouveau objets Billet et Commentaire.
  - billet($idBillet) : Affiche les détails sur un billet
  - commenter($auteur, $contenu, $idBillet) : Ajoute un commentaire à un billet
- Routeur.php : le routeur dont la méthode principale est d'analyser la requête entrante pour déterminer l'action à entreprendre. Contient 4 fonctions:
  - __construct() : Crée deux nouveau objets ControleurAccueil et ControleurBillet.
  - routerRequete() : Route une requête entrante : exécution l'action associée
  - erreur($msgErreur) : Affiche une erreur
  - getParametre($tableau, $nom) : Recherche un paramètre dans un tableau

Exercice

6. Créer un répertoire Contenu regroupant le style:

- Style.css : Une feuille de style CSS est utilisée afin d'améliorer le rendu HTML