PREPARATION D'UNE PROGRAMMATION AJAX

Résumé de votre projet Ajax :
Ajout et modification d'un commentaire par un utilisateur connecté

	Formulez-le en français (dossier fonctionnel)	**Donnez des détails techniques (format, protocole, paramètres, champs, balises, id, etc.)**
ÉVÉNEMENT **Quelle est l'événement qui déclenche l'Ajax ?**	Lorsque l'utilisateur clique sur le bouton envoyer après avoir écrit son commentaire	Dans le fichier film.php : Bouton ENVOYER : id = « envoie » →Lance la fonction envoyerCommentaire()
REQUÊTE **Quelle est la question posée au serveur ?** **Quelle information envoyez-vous au serveur en posant la question ?**	Ajout ou modification du commentaire dans la base de données, à partir de l'id du film, de l'id de l'utilisateur et du commentaire tapé	**Vérifie si l'utilisateur a déjà commenté le film, recherche du commentaire dans la base de données :** url : Clapii-projet-web/Action/récupérer-commentaire.php Méthode GET paramètres : id_film et id_utilisateur **Ajout si l'utilisateur n'a pas encore commenté, appel de la fonction nouveauCommentaire(…):** url : Clapii-projet-web/Action/traitement-ajouter-commentaire.php Méthode POST paramètres : id_film, id_utilisateur et text **Modification si l'utilisateur a déjà commenté, appel de la fonction modifierCommentaire(…) :** url : Clapii-projet-web/Action/traitement-modification-commentaire.php Méthode POST paramètres : id_film, id_utilisateur et text
ACTION **Quelle action le serveur accompli-t-il avant de répondre ?**	Cherche le commentaire dans la base de données : →Ajoute un nouveau commentaire dans la base de données **OU** →Modifie le commentaire déjà existant	**Recherche du commentaire :** Fichier : Action/recuperer-commentaire.php CommentaireDAO::recupererCommentaireParIdUtilisateurEtIdFilm(…) Requête : `SELECT * FROM commentaire WHERE id_film = :id_film AND id_utilisateur = :id_utilisateur` **Si l'utilisateur n'a pas encore commenté :** Fichier : Action/traitement-ajouter-commentaire.php CommentaireDAO::insererCommentaire(…) Requête : `INSERT INTO commentaire( id_utilisateur, id_film, text) VALUES ( :id_utilisateur, :id_film, :text)` **Si l'utilisateur a déjà commenté :** Fichier : Action/traitement-modification-commentaire.php CommentaireDAO::modifierCommentaireParId(…) Requête : `UPDATE commentaire SET text = :text WHERE id = :id`

RÉPONSE **Quelle information le serveur vous répond-t-il ?**	Le serveur répond par le commentaire qu'il a trouvé ou par un message d'erreur La réponse engendre l'ajout (appel de nouveauCommentaire(…)) ou la modification (appel de modifierCommentaire(…))	Le commentaire retourné est sous format JSON ou égal à -1 si aucun commentaire correspondant à l'id du film et l'id de l'utilisateur n'est trouvé Dans le premier cas le serveur répond par la modification du commentaire déjà présent dans la base de données et retourne un message informant de l'état de la modification (effectuée ou non) Dans le second cas le serveur répond par l'ajout du commentaire dans la base de données et retourne un message informant de l'état de l'insertion
AFFICHAGE **Quelle rétroaction donnez-vous à l'utilisateur**	L'utilisateur retourne à la page initiale du film et peut remarquer dans le cadre « Mon commentaire » le changement qu'il a apporté à son commentaire, il peut ensuite le supprimer ou le modifier	Appel de la méthode toggleCommentaire() qui permet de changer l'affichage de la page passant de la liste de tous les commentaires à l'affichage modification de commentaire **Elément disparaissant :** Champ de modification du commentaire, id= « ecrireCommentaire » **Eléments apparaissant :** Commentaire de l'utilisateur, id= « commentaireEnregistré » Liste de tous les commentaires, class= « listeCommentaires » **Eléments modifiés :** Message sur le bouton COMMENTER/ANNULER, id= « Commenter »