

TP JSON 3

Plutôt que d'appeler une page pour chaque action sur une ressource (client, produit, facture), on préfère utiliser une page PHP unique pour chaque ressource.

L'action à effectuer sur la ressource sera cette fois décrite par la méthode HTTP à utiliser sur cette URL.

Ces méthodes sont standardisées au sein du protocole HTTP par la référence RFC 2616 <https://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html>.

1. Citez les différentes méthodes standards utilisables dans le protocole HTTP en précisant leurs utilités.

Pour chaque ressource de l'application (client, facture et produit), on va utiliser une page PHP qui permettra d'effectuer des actions en fonction des méthodes HTTP de la requête envoyée côté client.

Ces actions sont : l'ajout, la lecture, la modification, la suppression et forment ce que l'on appelle un CRUD (Create, Read, Update, Delete).

L'objectif est donc que pour chaque ressource (client, facture et produit), que l'on puisse à travers leurs propres pages (client.php, facture.php et produit.php), ajouter, lire, modifier et supprimer une ressource par le biais de la méthode HTTP choisie lors de l'appel des pages PHP.

2. Associez la bonne méthode HTTP pour chaque action du CRUD en complétant le tableau suivant :

Action sur la ressource	Méthode(s) HTTP à utiliser
Ajouter	
Lire	
Modifier	
Supprimer	

3. Modifiez les pages client.php, facture.php et produit.php afin de pouvoir ajouter, lire, modifier et supprimer une ressource en les appelant à travers les méthodes HTTP définies auparavant.

Pour les actions d'ajout, de suppression et de modification, on devra retourner un JSON avec un message indiquant la bonne exécution de la requête.

On doit pouvoir ajouter une nouvelle ressource via un JSON envoyé dans le corps de la requête, modifier une ressource existante en passant un JSON dans lequel on précise l'ID de la ressource et ses nouvelles valeurs, supprimer une ressource via son ID précisé dans un JSON, afficher toutes les ressources ou seulement une ressource spécifique en passant en l'ID de la ressource en paramètre GET de l'URL

`$_SERVER['REQUEST_METHOD']` → retourne la méthode HTTP utilisée

Vous testerez le bon fonctionnement de vos différentes méthodes à travers l'extension de navigateur RESTED, téléchargeable depuis le catalogue d'extensions Mozilla ou le Chrome Web Store.

<https://chrome.google.com/webstore/detail/rested/eelcnbccaccipfolokglfhmapdchbfg>

<https://addons.mozilla.org/fr/firefox/addon/rested/>

Le protocole HTTP utilise également des codes d'état (de 3 chiffres) dans les réponses HTTP permettant d'informer de l'état de la requête. Le plus connu de ces codes d'état est sans doute le code 404.

Ces codes sont normalisés à travers la RFC 2616 <https://tools.ietf.org/html/rfc2616#section-6.1.1>

Le premier chiffre permet de définir le type d'état de la réponse HTTP.

4. Précisez la signification de chaque premier chiffre des codes erreurs HTTP en complétant le tableau suivant :

Codes d'état	Signification
1xx	
2xx	
3xx	
4xx	
5xx	

Nous allons utiliser les codes HTTP dans les différentes réponses de chaque action afin de définir l'état de la réponse (succès, erreur, etc...). Cela permettra de prévenir des erreurs comme un identifiant de produit inconnu, des erreurs de connexion à la base de données ou en cas de succès.

La modification du code d'état HTTP en PHP s'effectue via la fonction `http_response_code` (<https://secure.php.net/manual/fr/function.http-response-code.php>).

5. Modifiez les pages PHP client.php, facture.php et produit.php afin de retourner des codes d'état HTTP adaptés aux divers cas de réponse.