

# HTML / CSS / PHP

## Todolist - CHAPITRE 3

# Les formulaires

## Objectif

Ajouter une nouvelle tâche dans la liste.

## Points abordés

- Gestion des formulaires
- Envoi de données via POST et GET

## Todo List

### Gestionnaire de tâches

#### Ajouter une tâche

ou

☐

 Passer le balai

☒

~~Saluer le boss~~

☐

 Couper l'ordi

# PHP

## Documentations

### Le formulaire d'ajout de tâches

Base du formulaire HTML pour l'ajout de tâches

```
<form class="form-add" method="post">
  <label for="item">Ajouter une tâche</label>
  <div class="fields">
    <input name="field-task" type="text" id="item" placeholder="Intitulé de la tâche" />
    ou
    <select name="select-task">
      <option value="">Choisir une tâche</option>
      <option value="Passer le balai">Passer le balai</option>
      <option value="Saluer le boss">Saluer le boss</option>
      <option value="Couper l'ordi">Couper l'ordi</option>
    </select>
  </div>
  <input type="hidden" name="mode" value="add" />

  <button type="submit">Ajouter</button>
</form>
```

# PHP

## Documentations

Soumission d'un formulaire

### Le bouton Submit

Le bouton "Submit" dans un formulaire HTML est un élément essentiel qui permet à l'utilisateur de soumettre les données du formulaire au serveur pour traitement.

Il est créé avec la balise **<input>** de type « submit » ou via la balise **<button>** pour créer un bouton "Submit" dans un formulaire HTML

```
<input type="submit" value="Ajouter"/>  
ou  
<button type="submit">Ajouter</button>
```

# PHP

## Documentations

Soumission d'un formulaire

### Attribut « action » d'un Formulaire HTML

L'attribut **action** est un attribut clé utilisé dans les balises **<form>** en HTML pour spécifier l'URL vers lequel les données du formulaire seront envoyées lorsqu'il est soumis.

Si l'attribut **action** est omis, le formulaire sera soumis à la même URL de la page actuelle, ce qui est souvent utilisé pour les formulaires qui se soumettent à eux-mêmes.

```
<form class="form-add" method="post" action="traitement.php">
```

# PHP

## Documentations

### Récupération des données de formulaire

L'une des fonctionnalités essentielles de PHP est sa capacité à traiter les données soumises via des formulaires HTML. Pour cela nous utiliserons 2 verbes HTTP : GET et POST.

#### Méthode GET :

La méthode GET permet de transmettre les données via l'URL, ce qui rend les informations visibles et compréhensibles pour l'utilisateur. La méthode GET est appropriée pour des opérations de lecture, de récupération de données et pour des scénarios où la visibilité des données dans l'URL, la simplicité et la possibilité de partager des liens sont des avantages

#### Méthode POST :

Les données soumises via POST sont encapsulées dans le corps de la requête HTTP. POST n'a pas de limitation stricte sur la quantité de données pouvant être envoyées, contrairement à GET, qui est limité par la longueur maximale de l'URL. Cela en fait un choix approprié pour l'envoi de données volumineuses.

POST permet l'envoi de données plus complexes, telles que des tableaux, des objets JSON, ou des données structurées, ce qui est utile pour des cas d'utilisation avancés.

# PHP

## Documentations

Traitement du formulaire

### Identifier la methode utilisée (verbe HTTP) POST ou GET ?

**\$\_SERVER["REQUEST\_METHOD"]** : superglobale en PHP qui permet de récupérer la méthode HTTP utilisée pour accéder à une page. Cette superglobale est utilisée pour déterminer si une page a été appelée avec la méthode GET ou POST.

Focus sur **\$\_SERVER** -> <https://www.php.net/manual/en/reserved.variables.server.php>

### Le formulaire est envoyé en POST

**\$\_POST** est une superglobale en PHP qui permet d'accéder aux données soumises via la méthode HTTP POST. C'est un tableau associatif des valeurs passées :

```
if($_SERVER['REQUEST_METHOD'] == "POST") {  
    $field_task = $_POST['field-task'];  
    $select_task = $_POST['select-task'];  
    // Suite du code
```

La clé(field\_task ou select-task) est la valeur de l'attribut « name » du champ du formulaire

# PHP

## Documentations

Traitement du formulaire

### Le formulaire est envoyé en GET

**\$\_GET** est une superglobale qui permet de récupérer des données envoyées vers un script PHP via l'URL en utilisant la méthode HTTP GET. Les données sont incluses dans l'URL sous forme de paramètres.

```
if ($_SERVER['REQUEST_METHOD'] == "GET") {  
    $field_task = $_GET['field-task'];  
    $select_task = $_GET['select-task'];  
    // Suite du code
```

### Utiliser les valeurs

Les variables `$field_task` et `$select_task` contiennent à présent les valeurs saisies dans le formulaire.



# PHP

## Documentations

### Gestion des formulaires

<https://www.php.net/manual/fr/language.variables.external.php>

### Les variables prédéfinies

<https://php.net/manual/en/reserved.variables.php>

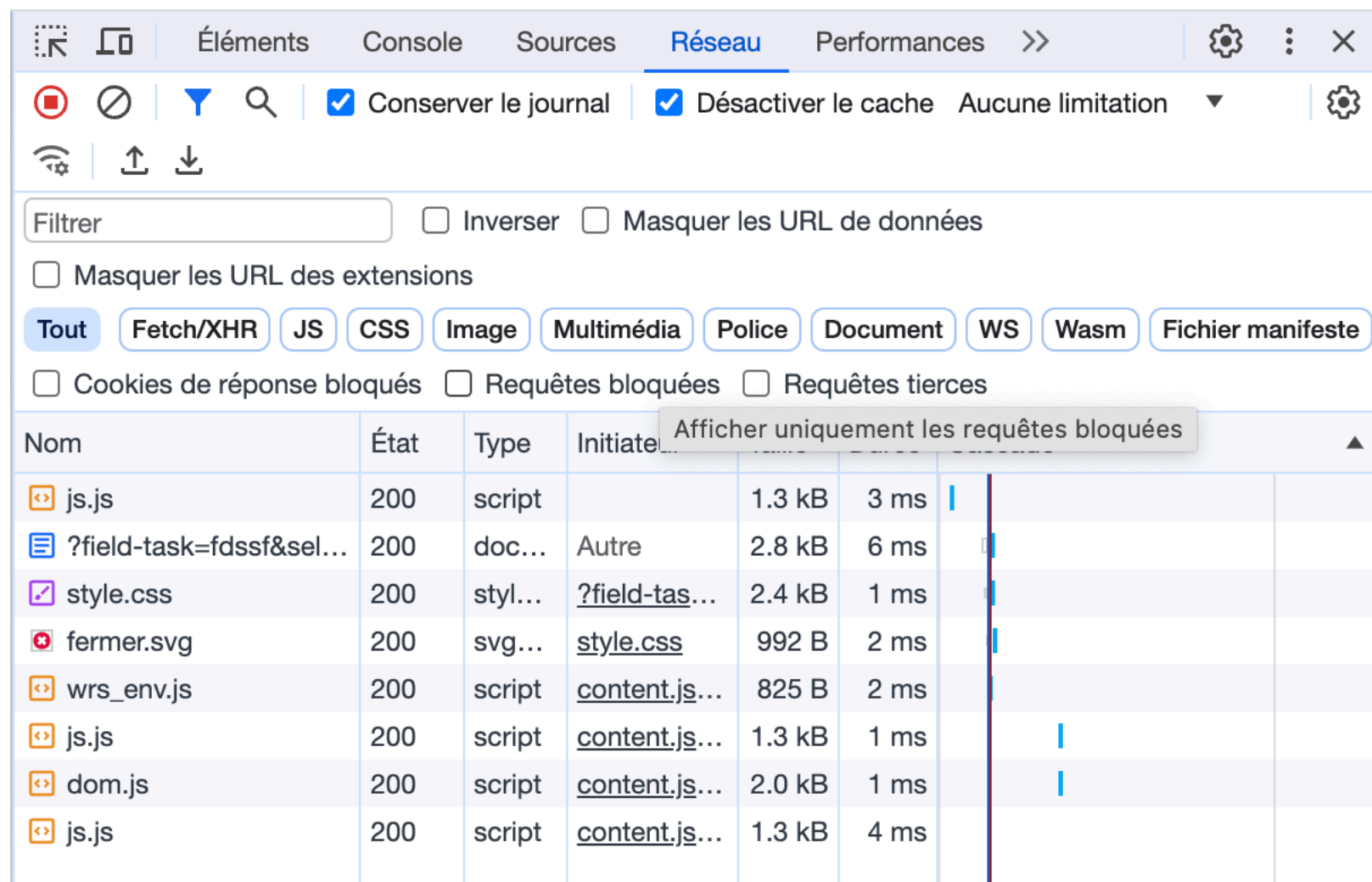
# PHP

## Documentations

Inspecter les échanges

Afin de voir les informations passés lors de la soumission du formulaire nous allons utiliser le débogueur interne du navigateur.

**Click droit dans la page -> Inspecter -> Onglet Réseau / Network**



The screenshot shows the Chrome DevTools Network tab. The top bar includes tabs for 'Éléments', 'Console', 'Sources', 'Réseau' (selected), and 'Performances'. Below the tabs, there are checkboxes for 'Conserver le journal' (checked), 'Désactiver le cache' (checked), and 'Aucune limitation'. A search bar and a filter dropdown are also present. The main area displays a list of network requests with columns for 'Nom', 'État', 'Type', 'Initiateur', 'Taille', and 'Durée'. A tooltip 'Afficher uniquement les requêtes bloquées' is visible over the 'Initiateur' column. The list includes requests for 'js.js', '?field-task=fdssf&sel...', 'style.css', 'fermer.svg', 'wrs\_env.js', and several other 'js.js' files.

Nom	État	Type	Initiateur	Taille	Durée
js.js	200	script		1.3 kB	3 ms
?field-task=fdssf&sel...	200	doc...	Autre	2.8 kB	6 ms
style.css	200	styl...	?field-tas...	2.4 kB	1 ms
fermer.svg	200	svg...	style.css	992 B	2 ms
wrs_env.js	200	script	content.js...	825 B	2 ms
js.js	200	script	content.js...	1.3 kB	1 ms
dom.js	200	script	content.js...	2.0 kB	1 ms
js.js	200	script	content.js...	1.3 kB	4 ms

# PHP

## Documentations

Inspecter les échanges

En sélectionnez le fichier « document » `?field-task=fdssf&select-task...`

Dans la partie en bas à droite, sélectionnez l'onglet « Charge utile »

The screenshot shows the Chrome DevTools Network tab with the 'Réseau' (Network) panel selected. The 'Charge utile' (Payload) view is active, displaying the request body for the selected request. The request is a GET request to the URL `?field-task=fdssf&select-task...`. The request body is a JSON object with the following properties:

- `field-task`: fdssf
- `select-task`:
- `mode`: add

The 'Données de formulaire' (Form data) view is also visible, showing the same data. The 'Paramètres de chaîne de requête' (Query string parameters) view is also visible, showing the same data.

Nom	En-têtes	Charge utile	Aperçu	Réponse	Initiateur	Délai
js.js						
<code>?field-task=fdssf&amp;select-task...</code>		<b>Paramètres de chaîne de requête</b> afficher la source afficher les données encodées en URL <code>field-task: fdssf</code> <code>select-task:</code> <code>mode: add</code>				
style.css						
fermer.svg						
wrs_env.js						
js.js		<b>Données de formulaire</b> afficher la source afficher les données encodées en URL <code>field-task: test</code> <code>select-task:</code> <code>mode: add</code>				
dom.js						
js.js						

On pourra également observer la réponse -> Onglet « Réponse »

# PHP

## Documentations

Inspecter les échanges

Essayez de changer la méthode du formulaire en GET

The screenshot shows the Chrome DevTools Network tab. The top bar includes tabs for Elements, Console, Sources, Réseau (selected), Performances, Mémoire, Appli, Sécurité, Lighthouse, and Enregistreur. Below the tabs, there are checkboxes for 'Conserver le journal' (checked), 'Désactiver le cache' (checked), and 'Aucune limitation'. A search bar and filters are also present. The main area displays a list of requests, including '?field-task=dsqdsq&select-ta...', 'style.css', 'fermer.svg', 'wrs\_env.js', 'js.js', 'dom.js', and 'js.js'. The selected request is expanded, showing the 'Charge utile' (Payload) tab. The payload details include 'Paramètres de chaîne de requête' (Request body parameters) with the following values: 'field-task: dsqdsq', 'select-task:', and 'mode: add'.

Pour retrouver la méthode HTTP utilisée (GET ou POST), vous pouvez l'observer dans l'onglet « En-têtes »

Observez également la barre d'adresse du navigateur

# PHP

## Documentations

### Gérer les erreurs

Afin de gérer les erreurs nous utiliserons le débogueur interne de PHP.

Ajoutez **tout en haut** de votre fichier PHP :

```
<?php

ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
ini_set('error_log', 'error.log');
```

Gestion des erreurs en PHP -> <https://www.php.net/manual/fr/function.ini-set.php>

# Exercice 1

Ajouter une tâche dans la liste via la formulaire.

Gérer la différence (algorithme) entre le champ texte et la liste déroulante.

Temps : 2H

## Todo List

### Gestionnaire de tâches

#### Ajouter une tâche

 ou

☐

 Passer le balai

☒

 Saluer le boss

☐

 Couper l'ordi

☐

 Nouvelle tâche