Laravel - Ajax

[sommaire]

1. **Objectifs**

- **Présentation**
- **Utilisation de la session**

Présentation

- **Ajax** (**A**synchronous **J**avaScript and **X**ML) est un ensemble de techniques de développement Web utilisant de nombreuses technologies Web utilisées côté client pour créer des applications Web asynchrones.
- **AJAX** est principalement utilisé pour effectuer des requêtes HTTP sans faille pour lire, écrire, mettre à jour et supprimer les données du serveur. **AJAX** est un outil qui fait le consensus entre le client et le serveur. L'utilisation d'**AJAX** est du temps primordial, évidemment du point de vue du développement web.
- **AJAX** est utilisé via **jQuery**, et un simple lien jQuery peut le propulser. Le jQuery offre d'excellentes fonctionnalités, et AJAX en fait partie.

Utilisation de la session

- La session peut être accédée de plusieurs manières, via la méthode session de la requête HTTP , la façade Session ou la fonction session d'assistance.
- Lorsque l'assistant session est appelé sans arguments, il renverra tout l'objet de session.
- Par exemple: `session()->regenerate();`

- **Stockage d'un élément dans la session**
  - Il existe deux méthodes différentes dans Laravel Framework que vous pouvez utiliser pour définir la valeur dans la session. Le premier utilise la fonction d'assistance de session **`session()`**.
  - `Session::put('key', 'value');`
  - `session(['key' => 'value']);`
  - Deuxièmement, vous pouvez utiliser l' instance `Request` pour définir la valeur dans la session.
  - `$request-> session () -> put ([ 'key' => 'value' ]);`

apcpedagogie.com

- **Pousser une valeur sur une valeur de session de baie**
  - `Session::push('user.teams', 'developers');`
- **Récupération d'un élément de la session**
  - `$value = Session::get('key');`
  - `$value = session('key');`
- **Récupérer un article ou renvoyer une valeur par défaut**
  - `$value = Session::get('key', 'default');`
  - `$value = Session::get('key', function() { return 'default'; });`
- **Récupérer un élément et l'oublier**
  - `$value = Session::pull('key', 'default');`
- **Récupération de toutes les données de la session**
  - Si vous souhaitez récupérer toutes les données disponibles dans la session, vous pouvez utiliser la méthode **all**.
  - `$data = Session::all();`
- **Déterminer si un élément existe dans la session**
  - Pour déterminer si une valeur est présente dans la session, vous pouvez utiliser la méthode **has**.
  - La méthode **has** renvoie **true** si la valeur est présente et **null** si la valeur n'existe pas.
  - `if (Session::has('users')) { // }`
  - ou

    `if ($request-> session() -> has ( 'key' ) { // }`
- **Suppression d'un élément de la session**
  - La méthode `forget` supprimera l'élément spécifié de la session
  - Si vous souhaitez d'abord obtenir la valeur de l'élément, puis supprimer cet élément de la session, vous pouvez utiliser la méthode `pull`
  - La différence entre la méthode `forget()` et la méthode `pull()` est que:
    - La méthode `forget()` ne retournera pas la valeur de la session.

apcpedagogie.com

        - La méthode `pull()` la retournera et supprimera cette valeur de la session.
    - `Session::forget('key');`

- **Suppression de tous les éléments de la session**

    - Si vous souhaitez supprimer tous les éléments de la session, vous pouvez utiliser la méthode `flush()`:
    - Parfois, vous souhaiterez peut-être stocker des éléments dans la session uniquement pour la demande suivante. Vous pouvez le faire en utilisant la méthode flash.
    - Les données stockées dans la session à l'aide de cette méthode ne seront disponibles que lors de la requête HTTP suivante, puis seront supprimées.
    - Les données Flash sont principalement utiles pour les messages d'état de courte durée.
    - `Session::flush();`

- **Régénérer l'ID de session**

    - Si vous souhaitez régénérer tous les **ID** de la session, vous pouvez utiliser la méthode `regenerate()`.
    - `Session::regenerate();`
    - ou `$request->session()-> régénérer();`

Pilote de session mise en disposition

- Le gestionnaire de sessions de Laravel supporte différents pilotes de gestions de session
- Les différents pilotes mis à disposition en session laravel sont les suivants:
    - **File** : Ici, les sessions sont stockées dans le répertoire - storage / framework / sessions.
    - **Cookie** : Les cookies sécurisés et cryptés sont l'endroit où les sessions sont stockées.
    - **Base de données** : les sessions ici sont stockées dans une base de données relationnelle.
    - **Tableau** : Un tableau PHP est l'endroit où les sessions sont stockées et elles ne seront plus conservées.
    - **Memcached/Redis** : Ce sont des magasins et des sessions rapides et basés sur le cache.
- Le pilote par défaut de la session est défini sur fichier lorsque vous créez une nouvelle application Laravel. Le lecteur de fichiers est idéal pour le développement local, mais pour une application de production, vous devez utiliser Redis ou Memcached pour améliorer les performances de session.
- Vous pouvez modifier les paramètres du pilote de `session` à partir du fichier de configuration de la session située dans `config/session.php`.

apcpedagogie.com

```
<?php

use Illuminate\Support\Str;

return [

    /*
    |--------------------------------------------------------------------------
    | Default Session Driver
    |--------------------------------------------------------------------------
    |
    | This option controls the default session "driver" that will be used on
    | requests. By default, we will use the lightweight native driver but
    | you may specify any of the other wonderful drivers provided here.
    |
    | Supported: "file", "cookie", "database", "apc",
    |            "memcached", "redis", "dynamodb", "array"
    |
    */

    'driver' => env('SESSION_DRIVER', 'file'),

    /*
```

Sommaire du cours Laravel

Source : https://apcpedagogie.com/laravel-ajax · téléchargé le 03/11/2026 · © APC Pédagogie · apcpedagogie.com

apcpedagogie.com