

Session:

1.Introduction :

Comme les applications HTTP sont sans état, les sessions permettent de stocker des informations sur l'utilisateur à travers plusieurs requêtes. Ces informations sur l'utilisateur sont généralement placées dans un **store / backend** persistant auquel il est possible d'accéder lors de requêtes ultérieures.

Dans laravel, le fichier de configuration de session est stocké dans `'app/config/session.php'`.

2.Interagir avec la session

2.1. *Récupération des données*

Il y a deux façons principales de travailler avec les données de session dans Laravel : avec le helper **session** global et via une instance de **Request**. Tout d'abord, examinons l'accès à la session via une instance de requête, qui peut être indiquée par un type sur une fermeture de route ou une méthode de contrôleur. N'oubliez pas que les dépendances des méthodes de contrôleur sont automatiquement injectées par le conteneur de services Laravel :

```
public function show(Request $request,Student $student)
{

    $value = $request->session()->get('_token','default');
    dd($value);
    return view('students.profile', ['student' => $student]);
}
```

`//output`

```
"c3mc8An5ySsbbbIrVxWsHdy0Lh8InycGrx2bTkwe" // app\Http\Controllers\StudentController.php:72
```

Lorsque vous récupérez un élément dans la session, vous pouvez également passer une valeur par défaut comme deuxième argument de la méthode **get**. Cette valeur par défaut sera renvoyée si la clé spécifiée n'existe pas

dans la session. Si vous passez une fermeture comme valeur par défaut à la méthode `get` et que la clé demandée n'existe pas, la fermeture sera exécutée et son résultat sera renvoyé :

```
$value = $request->session()->get('key', 'default');

$value = $request->session()->get('key', function () {
    return 'default';
});
```

- Le helper session

Vous pouvez également utiliser la fonction PHP de session globale pour récupérer et stocker des données dans la session. Lorsque la fonction de session est appelée avec un argument unique de type chaîne, elle renvoie la valeur de la clé de session. Lorsque le helper est appelé avec un tableau de paires **clé/valeur**, ces valeurs seront stockées dans la session :

```
Route::get('/home', function () {
    // Retrieve a piece of data from the session...
    $value = session('key');

    // Specifying a default value...
    $value = session('key', 'default');

    // Store a piece of data in the session...
    session(['key' => 'value']);
});
```

- Récupération de toutes les données de la session

Si vous souhaitez récupérer toutes les données de la session, vous pouvez utiliser la méthode `all` :

```
$data = $request->session()->all() ;  
//output
```

```
array:4 [▼ // app\Http\Controllers\StudentController.php:80  
  "_token" => "7z1rYDhqqTA9uMe4015XE2C2NsD2NhxDuV7tHrJv"  
  "key" => "value"  
  "_previous" => array:1 [▶]  
  "_flash" => array:2 [▶]  
]
```

- Déterminer si un élément existe dans la session

Pour déterminer si un élément est présent dans la session, vous pouvez utiliser la méthode **has**. La méthode **has** renvoie **true** si l'élément est présent et n'est pas nul :

```
if ($request->session()->has('users')) {  
    //  
}
```

Pour déterminer si un élément est présent dans la session, même si sa valeur est nulle, vous pouvez utiliser la méthode **exists** :

```
if ($request->session()->exists('users')) {  
    //  
}
```

2.2. Stockage des données

Pour stocker des données dans la session, vous utiliserez généralement la méthode **put** de l'instance **Request** ou via la fonction **session** :

```
// Via a request instance...  
$request->session()->put('key', 'value');  
// Via the global "session" helper...  
session(['key' => 'value']);
```

- Insérer des valeurs de session de type tableau

La méthode push peut être utilisée pour insérer une nouvelle valeur dans une valeur de session qui est un tableau. Par exemple, si la clé `user.teams` contient un tableau de noms d'équipes, vous pouvez insérer une nouvelle valeur dans le tableau de la manière suivante :

```
$request->session()->put('teams',['programmers']);  
$request->session()->push('teams', 'developers');  
dd(session('teams'));
```

`//output`

```
array:2 [▼ // app\Http\Controllers\StudentController.php:82  
  0 => "programmers"  
  1 => "developers"  
]
```

- Récupérer un élément et l'oublier

La méthode pull permet de récupérer et de supprimer un élément de la session en une seule instruction :

```
$value = Session::pull('key', 'default');  
$value = $request->session()->pull('key', 'default');
```

- Incrémentation et décrémentation des valeurs de session

Si vos données de session contiennent un nombre entier que vous souhaitez incrémenter ou décrémenter, vous pouvez utiliser les méthodes d'incrémentation et de décrémentation :

```
$request->session()->increment('count');  
  
$request->session()->increment('count', $incrementBy = 2);  
  
$request->session()->decrement('count');  
  
$request->session()->decrement('count', $decrementBy = 2);
```

```
echo Session::get('count'); // display value of count 2 4 6 8 ...
```

- Données flash

Il peut arriver que vous souhaitiez stocker des éléments dans la session pour la prochaine demande. Vous pouvez le faire en utilisant la méthode **flash**. Les données stockées dans la session à l'aide de cette méthode seront disponibles immédiatement et pendant la demande HTTP suivante. Après la demande HTTP suivante, les données flashées seront supprimées. Les données flash sont principalement utiles pour les messages d'état de courte durée :

```
$request->session()->flash('status', 'Task was successful!');
```

- Régénérer l'ID de session

Si vous souhaitez régénérer tous les **ID** de la session, vous pouvez utiliser la méthode `regenerate()`.

```
Session::regenerate();  
$request->session()->regenerate();
```

- 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.
- La méthode **pull()** la retournera et supprimera cette valeur de la session.

```
// Forget a single key...  
$request->session()->forget('name');  
  
// Forget multiple keys...  
$request->session()->forget(['name', 'status']);
```

La méthode **forget** permet de supprimer un élément de données de la session. Si vous souhaitez supprimer toutes les données de la session, vous pouvez utiliser la méthode **flush** :

```
$request->session()->flush();  
Session::flush();
```

1. Activité

Étape 1 – Créez un contrôleur appelé *SessionController*.

Étape 2 – Copiez le code suivant dans un fichier à `app/Http/Controllers/SessionController.php`.

```
<?php  
  
namespace App\Http\Controllers;  
  
use Illuminate\Http\Request;  
use App\Http\Requests;  
use App\Http\Controllers\Controller;  
  
class SessionController extends Controller {  
    public function accessSessionData(Request $request) {  
        if($request->session()->has('my_name'))  
            echo $request->session()->get('my_name');  
        else  
            echo 'No data in the session';  
    }  
    public function storeSessionData(Request $request) {  
        $request->session()->put('my_name','Virat Gandhi');  
        echo "Data has been added to session";  
    }  
    public function deleteSessionData(Request $request) {  
        $request->session()->forget('my_name');  
        echo "Data has been removed from session.";  
    }  
}
```

Étape 3 – Ajoutez les lignes suivantes dans le fichier app/Http/routes.php .

```
Route::get('session/get',[SessionController::class, 'accessSessionData'] );  
Route::get('session/set',[SessionController::class, 'storeSessionData']);  
Route::get('session/remove',[SessionController::class, 'deleteSessionData']);
```

- **Étape 4** – Visitez l'URL suivante pour définir les données en session :
<http://localhost:8000/session/set> .

Étape 5 - La sortie apparaîtra comme indiqué dans l'image suivante.

Data has been added to session

Étape 6 - Visitez l'URL suivante pour **obtenir les données de la session** .

<http://localhost:8000/session/get>

Étape 7 - La sortie apparaîtra comme indiqué dans l'image suivante.

Virat Gandhi

Étape 8 - Visitez l'URL suivante pour **supprimer les données de session** .

<http://localhost:8000/session/remove>

Étape 9 - Vous verrez un message comme indiqué dans l'image suivante.

Data has been removed from session.