

# Utilisation de la session

## Création du fichier database/security.php

Nous créons un fichier dans le dossier `database` que nous appelons `security.php` :

```
<?php

function isLoggedIn(): array | false
{
    global $pdo;
    $sessionId = $_COOKIE['session'] ?? '';
    if ($sessionId) {
        $statementSession = $pdo->prepare('SELECT * from session WHERE id=:id');
        $statementSession->bindValue(':id', $sessionId);
        $statementSession->execute();
        $session = $statementSession->fetch();
        if ($session) {
            $statementUser = $pdo->prepare('SELECT * FROM user WHERE id=:id');
            $statementUser->bindValue(':id', $session['userid']);
            $statementUser->execute();
            $user = $statementUser->fetch();
        }
    }
    return $user ?? false;
}
```

C'est la même logique que nous avons vue dans le chapitre précédent.

## Modification de index.php

Nous modifions `index.php` en vérifiant que l'utilisateur est logué ou non tout en haut du fichier :

```
<?php
require __DIR__ . '/database/database.php';
require __DIR__ . '/database/security.php';
$currentUser = isLoggedIn();
// ...
```

## Modification de header.php

Maintenant que nous avons accès à l'utilisateur et pouvons savoir s'il est connecté, nous allons simplement modifier la barre de navigation :

```
<?php
$currentUser = $currentUser ?? false;
?>

<header>
    <a href="/" class="logo">Dyma Blog</a>
    <ul class="header-menu">
        <?php if ($currentUser) : ?>
            <li class=<?= $_SERVER['REQUEST_URI'] === '/form-article.php' ? 'active' : ''>
                <a href="/form-article.php">Écrire un article</a>
            </li>
            <li class=<?= $_SERVER['REQUEST_URI'] === '/profile.php' ? 'active' : '' ?> h
header-profile">
                <a href="/profile.php">Ma page</a>
            </li>
            <li>
                <a href="/auth-logout.php">Déconnexion</a>
            </li>
        <?php else : ?>
            <li class=<?= $_SERVER['REQUEST_URI'] === '/auth-register.php' ? 'active' : ''>
                <a href="/auth-register.php">Inscription</a>
            </li>
            <li class=<?= $_SERVER['REQUEST_URI'] === '/auth-login.php' ? 'active' : '' ?>
```

```

>
    <a href="/auth-login.php">Connexion</a>
  </li>
<?php endif; ?>
</ul>
</header>

```

## Modification de ArticleDB.php

Nous enlevons l'import de \$pdo :

```

<?php
class ArticleDB
{
    private PDOStatement $statementCreateOne;
    private PDOStatement $statementUpdateOne;
    private PDOStatement $statementDeleteOne;
    private PDOStatement $statementReadOne;
    private PDOStatement $statementReadAll;

    function __construct(private PDO $pdo)
    {
        $this->statementCreateOne = $pdo->prepare('
            INSERT INTO article (
                title,
                category,
                content,
                image
            ) VALUES (
                :title,
                :category,
                :content,
                :image
            )
        ');
        $this->statementUpdateOne = $pdo->prepare('
            UPDATE article
            SET
                title=:title,
                category=:category,
                content=:content,
                image=:image
            WHERE id=:id
        ');
        $this->statementReadOne = $pdo->prepare('SELECT * FROM article WHERE id=:id');
        $this->statementReadAll = $pdo->prepare('SELECT * FROM article');
        $this->statementDeleteOne = $pdo->prepare('DELETE FROM article WHERE id=:id');
    }

    public function fetchAll()
    {
        $this->statementReadAll->execute();
        return $this->statementReadAll->fetchAll();
    }

    public function fetchOne(string $id)
    {
        $this->statementReadOne->bindValue(':id', $id);
        $this->statementReadOne->execute();
        return $this->statementReadOne->fetch();
    }

    public function deleteOne(string $id)
    {
        $this->statementDeleteOne->bindValue(':id', $id);
        $this->statementDeleteOne->execute();
        return $id;
    }

    public function createOne($article)
    {
        $this->statementCreateOne->bindValue(':title', $article['title']);
        $this->statementCreateOne->bindValue(':content', $article['content']);
        $this->statementCreateOne->bindValue(':category', $article['category']);
        $this->statementCreateOne->bindValue(':image', $article['image']);
        $this->statementCreateOne->execute();
        return $this->fetchOne($this->pdo->lastInsertId());
    }

    public function updateOne($article)

```

```
{  
    $this->statementUpdateOne->bindValue(':title', $article['title']);  
    $this->statementUpdateOne->bindValue(':content', $article['content']);  
    $this->statementUpdateOne->bindValue(':category', $article['category']);  
    $this->statementUpdateOne->bindValue(':image', $article['image']);  
    $this->statementUpdateOne->bindValue(':id', $article['id']);  
    $this->statementUpdateOne->execute();  
    return $article;  
}  
}
```

```
return new ArticleDB($pdo);
```