

SESSION ET COOKIES

Le protocole HTTP est sans état ; afin d'associer une requête à toute autre requête, il faut un moyen de stocker les données de l'utilisateur entre les demandes HTTP. Les cookies et les paramètres d'URL sont deux moyens appropriés pour transporter des données entre le client et le serveur. Mais ils sont tous deux lisibles et se trouvent du côté du client. Les sessions résolvent ce problème.

Vous attribuez au client un identifiant et il effectue toutes les demandes ultérieures en utilisant cet identifiant. Les informations associées au client sont stockées sur le serveur en lien avec cet ID.

Si la table n'est pas créée , créer la table session dans votre base de données

Exemple avec stockage de l'id de session dans mysql.

```
CREATE TABLE IF NOT EXISTS `sessions` (  
  `session_id` varchar(128) COLLATE utf8mb4_bin NOT NULL,  
  `expires` int(11) unsigned NOT NULL,  
  `data` mediumtext COLLATE utf8mb4_bin,  
  PRIMARY KEY (`session_id`)  
) ENGINE=InnoDB
```

```
npm init  
npm i express --save  
npm i express-session --save  
npm i express-mysql-session
```

dans un fichier js node:

```
const express = require('express')  
const cookieParser = require('cookie-parser')  
const session = require('express-session')  
const mysqlStore = require('express-mysql-session')  
let app = express()  
  
app.use(cookieParser())  
  
// options de connection à a base de données  
let options = {  
  host:"localhost",  
  port: 3306 ,
```

```

    user: 'root',
    password: '',
    database: 'session_test'
}

let sessionStore = new mysqlStore(options)

app.use(session({
  secret: "c'est un secret",
  store: sessionStore,
  resave: true,
  saveUninitialized: true,
  cookie: { maxAge: 60000 }
}))

// routage
app.get('/', (req,res) => {
  if (req.session.pages_vues) {
    console.log(req.cookies)
    req.session.pages_vues++
    res.send("Vous avez visité cette page" +
req.sessions.pages_vues + "fois" )
  }
  else
  {
    req.sessions.pages_vues = 1
    res.cookie('nom','Jam',{expire: 360000 })
    res.send("Bienvenue sur cette page pour la
1ere fois")
  }
})

app.get('/efface', (req,res) => {
  res.clearCookie('nom');
  res.send("cookie nom effacé");
})

app.listen(3000)

// en inspectant notre page le cookie connect.sid est
créé. C'est l'équivalent du PHP_SESSID de PHP

```