

Installation express-session :

- Exécuter : **npm install express-session**

Utilisation :

```
var session = require('express-session')
```

```
app.use(session({
  secret : 'abc',
  resave : false,
  saveUninitialized : true,
  cookie : { secure : false }
}))
```

EXEMPLE : TP5a

App.js

```
app.route('/tp5')
  .get((req, res) => {
    if(req.session.erreur){
      res.locals.erreur = req.session.erreur;
      req.session.erreur = undefined
    }
    res.render('tp5',{titre : 'tp5'})
  })
  .post((req,res) => {
    if(req.body.mess === undefined || req.body.mess === ''){
      req.session.erreur = 'il y a une erreur'
      res.redirect('/tp5')
    }
  });
```

tp5.ejs

```
<% if (locals.erreur) { %>
  <p> <%= erreur %> </p>
<% } %>
<form action="/tp5" method="post">
  <label for="mess">message</label> <br>
  <textarea name="mess" id="mess" cols="30" rows="10"></textarea> <br>
  <button type="submit"> Envoyer </button>
</form>
```

Installation mysql :

- Exécuter : **npm install mysql**

EXEMPLE : TP5b

App.js

```
app.use(require('./middlewares/flash'))

app.route('/tp5b')
  .get((req, res) => {
    let Message = require('./models/message')
    // console.log(req.session)
    Message.selectionneretout(function (messages) {
      res.render('tp5b',{titre : 'tp5', messages: messages})
    })
  })
  .post((req,res) => {
    if(req.body.mess === undefined || req.body.mess === ''){
      req.flash('erreur','aucun message')
    }
    else{
      let Message = require('./models/message')
      Message.inserer(req.body.mess, function(){
        req.flash('success','merci')
      })
    }
    res.redirect('/tp5b')
  });
```

// middlewares/flash.js

```
module.exports = function(req,res,next) {
  if(req.session.flash){
    res.locals.flash = req.session.flash;
    req.session.flash = undefined
  }

  req.flash = function (type,content) {
    if(req.session.flash === undefined){
      req.session.flash = {}
    }
    req.session.flash[type] = content
  }
  next();
}
```

```
// models/message.js

let cnx = require('./db')

class Message {
  static inserer(content,cb) {
    cnx.query(
      'INSERT INTO messages SET content = ?, created = ?',
      [content, new Date()],
      (err, result) => {
        if(err) throw err
        cb(result)
      })
  }
  static selectionnertout(cb){
    cnx.query(
      'SELECT * FROM messages ',
      (err, rows) => {
        if(err) throw err
        cb(rows)
      })
  }
}
}
```

```
// db.js
let mysql = require('mysql');

var cnx = mysql.createConnection({
  host : 'localhost',
  user : 'root',
  password : '',
  database : 'blog'
});
cnx.connect();

module.exports = cnx;
```

tp5b.ejs

```
<a href="/">Accueil</a> <br>
<ul>
  <% for (message of messages) { %>
    <li> <%= message.content %> , posté le <%= message.created %>    </li>
    <% } %>
  </ul>
  <% if (locals.flash && locals.flash.erreur) { %>
    <p> <%= flash.erreur %> </p>
  <% } %>
  <form action="/tp5b" method="post">
    <label for="mess">votre message</label> <br>
    <textarea name="mess" id="mess" cols="30" rows="10"></textarea> <br>
    <button type="submit"> Envoyer </button>
  </form>
```

EXERCICE : FORMULAIRE D'INSCRIPTION.

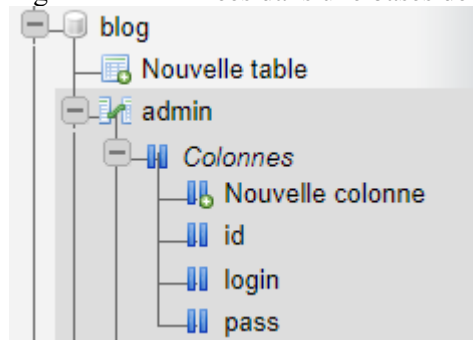
Créer en HTML un formulaire d'inscription comportant plusieurs champs dont au moins un nom, login, mdp et confirmation mdp. Le but de cet exercice est de faire en sorte que le serveur vérifie que le mot de passe est correct, et qu'il réponde un message du type "Bonjour Prenom Nom, ton compte est bien créé".

Faire en sorte que la soumission du formulaire provoque l'envoi (submit) des réponses à la page `http://localhost:3003/` à l'aide de la méthode POST.

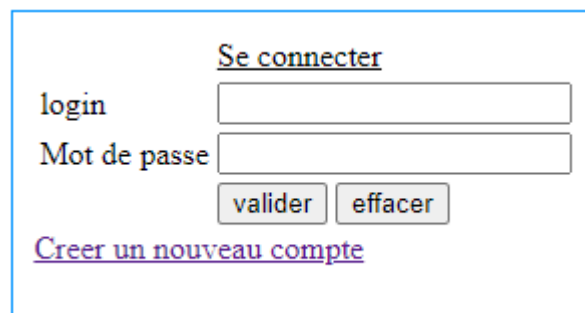
- Créer un serveur express et y ajouter une route POST qui récupère les champs du formulaire.
- Faire en sorte que cette route renvoie un message de confirmation d'inscription ou d'erreur selon que le mot de passe soit bien dupliqué ou non. En cas d'erreur l'utilisateur est invité à retaper son mot de passe dans le formulaire (sans que les autres champs soient effacés).



- Lorsqu'un formulaire est soumis, enregistrer les données dans une bases de données : table `admin(id,login,pass)`



- Créer une page de connexion



- Tant que les login et mot de passe sont correcte, afficher les données via la table `articles(id,title,content,date)`

