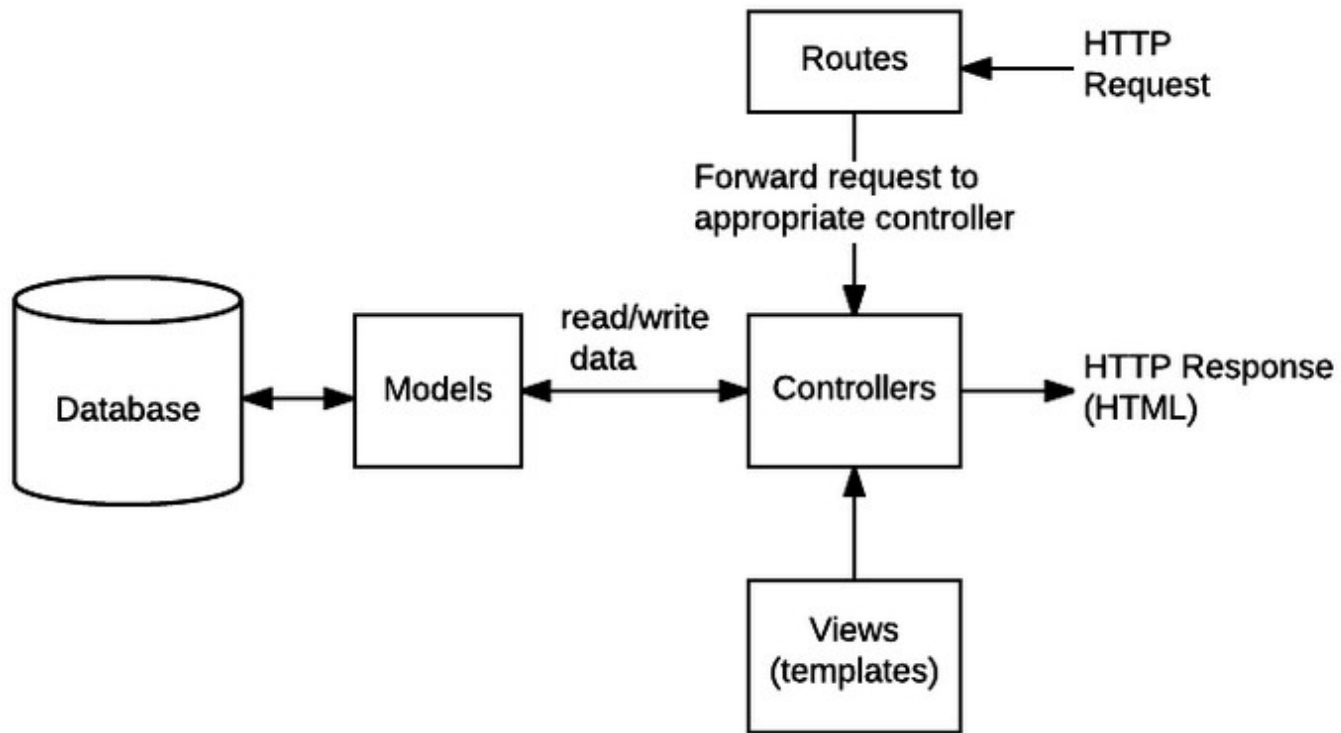


NodeJS : CRUD – Projet



Projet

- On crée un projet avec **Express-Generator** (ou on repart du projet existant de démo utilisant MySQL)
- On applique l'architecture **MVC**
- On installe les dépendances définies dans le fichier **package.json**
- On complète le fichier **app.js**
- On adapte la présentation dans **style.css**

Package.json

```
{
  "name": "listecontacts",
  "version": "0.0.0",
  "private": true,
  ▶ (débogage)
  "scripts": {
    "devstart": "nodemon ./bin/www",
    "start": "node ./bin/www"
  },
  "dependencies": {
    "cookie-parser": "~1.4.4",
    "debug": "~2.6.9",
    "ejs": "~2.6.1",
    "express": "~4.16.1",
    "http-errors": "~1.6.3",
    "moment": "^2.29.2",
    "morgan": "~1.9.1",
    "mysql": "^2.18.1"
  },
  "devDependencies": {
    "nodemon": "^2.0.15"
  }
}
```

App.js

- On ajoute la gestion des routes pour le CRUD des messages
- On adapte la page d'erreur

App.js – 1ère partie

```
var createError = require('http-errors');
var express = require('express');
var path = require('path');
var cookieParser = require('cookie-parser');
var logger = require('morgan');

var indexRouter = require('./routes/index');
//var usersRouter = require('./routes/users'); // pas besoin de ce router car pas de gestion
//des utilisateurs
var messagesRouter = require('./routes/messages'); // ce router gère les opérations sur les
//messages

var app = express();

// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'ejs');

app.use(logger('dev'));
app.use(express.json());
app.use(express.urlencoded({ extended: false }));
app.use(cookieParser());
app.use(express.static(path.join(__dirname, 'public')));

app.use('/', indexRouter);
//app.use('/users', usersRouter); // pas besoin de ce router car pas de gestion des
//utilisateurs
app.use('/messages', messagesRouter); // ce router regroupe les URL pointant vers des
//méthodes de gestion des messages
```

App.js – 2ème partie

```
// catch 404 and forward to error handler
app.use(function(req, res, next) {
  next(createError(404));
});

// error handler
app.use(function(err, req, res, next) {
  // set locals, only providing error in development
  res.locals.message = err.message;
  res.locals.error = req.app.get('env') === 'development' ? err : {};

  // render the error page
  res.status(err.status || 500);
  //res.render('error'); // modification de la page affichant les messages d'erreur
  res.render('error', {titre:"Erreur", message:"Une erreur s'est produite", error:err,
    active:""});
});

module.exports = app;
```

Style.css

- On améliore la présentation :
 - Du formulaire de contact
 - Des liens de navigation
 - De la liste des messages
 - Du tableau des messages
 - Des autres formulaires
 - Etc

Style.css – 1ère partie

```
body {
  padding: 50px;
  font: 14px "Lucida Grande", Helvetica, Arial, sans-serif;
}

a {
  color:  #00B7FF;
}

/* Formulaire de contact */
form {
  display: flex;
  flex-direction: column;
  align-items: center;
}

label, input {
  width: 100%;
  height: 30px;
}

input[type="submit"] {
  width: 50%;
  height: 50px;
}

.reponse {
  border: 2px solid  black;
  border-radius: 10px;
  padding: 1em;
}
```


Style.css – 2ème partie

```
/* Navigation */

#liens {
  display: flex;
  flex-flow: row nowrap;
  justify-content: space-between;
}

#liens a {
  display: block;
}

/* Liste des messages */
#messages article {
  border: 2px solid black;
  border-radius: 5px;
  background-color: lightgrey;
  padding: 5px;
  margin-bottom: 5px;
}

#messages article p {
  font-style: italic;
}
```

Style.css – 3ème partie

```
/* Tableau des messages */
#messages table, #messages tr, #messages th, #messages td {
    border: 2px solid black;
    border-collapse: collapse;
    padding: 5px;
}

#messages tr:nth-child(even) {
    background-color: lightgrey;
}

/* Modification d'un message */
#confirmbuttons {
    display: flex;
    flex-flow: row nowrap;
    justify-content: space-between;
    width: 100%;
}

#confirmbuttons input[type="submit"], #confirmform button, #editform input[type="submit"] {
    display: block;
    height: 50px;
    width: 200px;
}
```