

NodeJS : MVC

- Architecture MVC
- Réorganiser l'application
- Rôles de chaque élément dans le MVC
- Ajout des Contrôleurs
- Modification des Routeurs
- Gestion de la logique de l'application

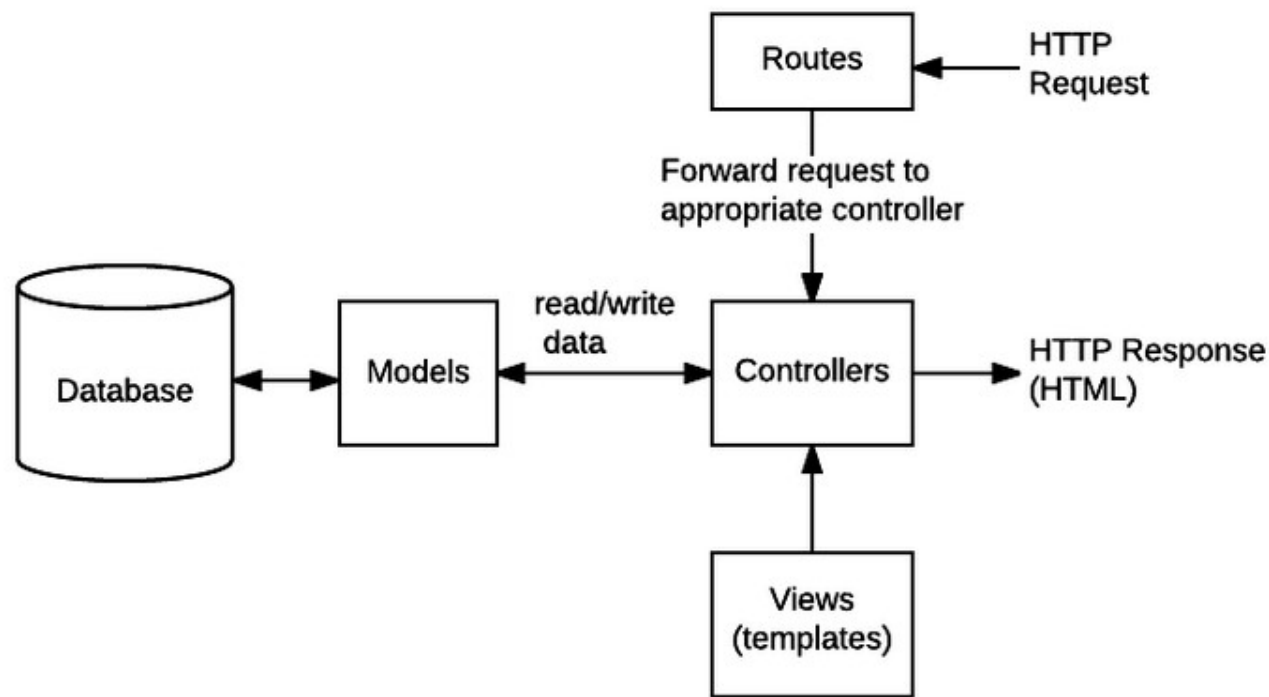
MVC (Model View Controller)

- **Model** : cela représente les données et les entités de l'application. Les données sont fournies par une base de données ou un service Web
- **View** : une vue est la couche présentation qui est responsable de l'affichage des informations pour l'utilisateur final
- **Controller** : le contrôleur fait le lien entre les modèles et les vues afin d'afficher la bonne page en fonction de la requête de l'utilisateur

Réorganiser l'application

- **Plus on ajoute de fonctionnalités**, plus l'application devient difficile à gérer
- En plus, on a mis la logique dans les **routeurs** alors qu'ils n'ont qu'un **rôle d'aiguillage**
=> à modifier
- On doit plutôt ajouter des **contrôleurs** qui vont contenir la logique et dialoguer avec les **modèles** et les **vues**
- Cela va simplifier les **routeurs** et leur redonner leur rôle initial

Architecture MVC



Structure **MVC** de l'application web avec Express
voir un [exemple complet sur MDN](#)

Contrôleur – messages

- Ce sont les **contrôleurs** qui doivent dialoguer avec les modèles (et les vues)
- La **logique** est déplacée dans les **contrôleurs** et non les routeurs => dossier « controllers »

```
// on importe l'objet Message et la connection à la DB au travers du modèle
const Message = require("../models/message.model");

// import du module "moment.js" pour l'affichage des dates/heures
const moment = require('moment');

console.log("on passe dans controllers/message.controller.js");

// Créer et sauver un nouveau message
exports.create = (req, res) => {

};

// Lire tous les messages
exports.readAll = (req, res) => {

};
```

Routeur – messages

- Le **routeur** doit communiquer avec les contrôleurs et non pas les modèles et les vues
- En fonction de la route demandée, il s'adresse au contrôleur adéquat

```
var express = require('express');
var router = express.Router();

const messages = require("../controllers/message.controller.js");

console.log("on passe dans routes/messages.js");

// Définition des routes pour gérer les messages
// Toutes les routes définies dans ce router commencent par l'URL : localhost:8080/messages

// Récupération de tous les messages : ReadAll
// URL => localhost:8080/messages/
router.get('/', messages.readAll);

// Création d'un message : Create
// URL => localhost:8080/messages/create
router.post('/create', messages.create);

module.exports = router;
```

Logique – créer un message

```
// Créer et sauver un nouveau message
exports.create = (req, res) => {
  console.log("POST Créer un message");
  const titrePage = "Formulaire reçu";
  const lenom = req.body.nom;
  const lemessage = req.body.msg;

  // Valider le contenu de la requête (est-ce-que le formulaire a été rempli ?)
  if ((!req.body)||!(lenom=="")||!(lemessage=="")) {
    console.log("Le contenu ne peut pas être vide !");
    res.redirect('/contact_form'); // retourner au formulaire
  } else {
    console.log(req.body);
    // Créer mon message avec le modèle
    const unMsg = new Message({
      nom: req.body.nom,
      msg: req.body.msg
    });

    Message.create(unMsg, function(err, data){
      if (err) {
        res.status(500).send({
          message: "Erreur pendant la création du message"
        });
      } else {
        console.log('Data = ', data);
        res.render('traiter_form', {titre: titrePage, nom: unMsg.nom, msg: unMsg.msg});
      }
    });
  }
};
```

Logique – lire les messages

```
// Lire tous les messages
exports.readAll = (req, res) => {
  console.log("GET Tous les messages");
  Message.readAll(function(err, data){
    if (err) {
      res.status(500).send({
        message: "Erreur pendant la lecture de tous les messages"
      });
    } else {
      console.log('Data = ', data);
      const titrePage = "Liste des messages";
      moment.locale('fr');
      res.render('listeMessages', {titre: titrePage, donnees: data, moment: moment});
    }
  });
};
```


Contrôleur à la racine

```
console.log("on passe dans controllers/root.controller.js");

exports.home = (req, res, next) => {
  res.render('index', { title: 'Express' });
};

exports.form = (req, res, next) => {
  res.render('contact_form', {titre: "Contact"});
};

exports.traitement = (req, res, next) => {
  let lenom = req.body.nom;
  let lemessage = req.body.msg;

  res.render('traiter_form', {titre: "Formulaire reçu", nom: lenom, msg: lemessage});
};
```

Routeur à la racine

```
var express = require('express');
var router = express.Router();
//var bodyParser = require('body-parser'); // avant Express v4.16

const root = require("../controllers/root.controller.js");

console.log("on passe dans routes/index.js");

// Définition des routes pour gérer les pages à la racine
// Toutes les routes définies dans ce router commencent par l'URL : localhost:8080/

/* GET home page. */
router.get('/', root.home);

/* Afficher le formulaire de contact qui va envoyer les données */
router.get('/contact', root.form);

/* Utiliser le middleware intégré body-parser pour analyser la requête POST */
//var urlencodedParser = bodyParser.urlencoded({ extended: false });
var urlencodedParser = express.urlencoded({extended:false});

/* Recevoir les données et afficher la page */
router.post('/traitement', urlencodedParser, root.traitement);

module.exports = router;
```