

Boutillier Paul  
HARDUIN Giovanni

# Node.js

# RAPPORT

- 1- Conception de l'application
- 2- Avantages et inconvénients de Node et Express
- 3- Comment avons-nous développé et pourquoi?
- 4- Les difficultés rencontrées
- 5- Conclusion

# 1 - Conception de l'application

Nos différents modèles :

```
// Application models
const User = mongoose.model("User", {username: String, password: String})
const Dish = mongoose.model("Dish", {name: String, description: String, allergens: String, price: Number})
const Cart = mongoose.model("Cart", {user: Object, dishes: Array, price: Number, deliveryAddress: String})
```

Nos différentes routes :

Auth	User	Dish	Cart
POST /login Route permettant de se connecter et de récupérer un token	GET /me Route permettant d'obtenir des infos sur notre profil	GET /dishes Route permettant de récupérer la liste des plats	POST /cart/finalize Route permettant de finaliser la commande
POST /register Route permettant de s'enregistrer		POST /dishes Route permettant d'ajouter un plat	GET /cart Route permettant d'obtenir le panier de l'utilisateur
		GET /dishes/:id Route permettant de récupérer un plat grâce à son id	POST /cart/:id Route permettant d'ajouter un plat au panier grâce à son id
			PUT /cart/:id Route permettant de supprimer un plat du panier grâce à son id
			DELETE /cart Route permettant de supprimer tout le panier

# L'utilisation des web token :

Création d'une fonction permettant de filtrer les requêtes en la passant en paramètres sur les routes ayant besoin d'une authentification

```
/**
 * Token utilities
 */
function createToken(user) {
  // Create token
  const token = jwt.sign({
    id: user.id,
    username: user.username
  }, SECRET, { expiresIn: '3 hours' })

  return token
}

// Decode token
async function decodeToken(req, res){

  // Get token
  const token = req.headers.authorization && extractBearerToken(req.headers.authorization)
  // Decode token
  const decoded = jwt.decode(token, {complete: false})

  return decoded
}
```

On remarque ci-dessus des fonctions permettant la création et le décryptage d'un token

```

/* Get bearer from header */
const extractBearerToken = headerValue => {
  if (typeof headerValue !== 'string') {
    return false
  }

  const matches = headerValue.match(/(bearer)\s+(\S+)/i)
  return matches && matches[2]
}

/* Check token */
const checkTokenMiddleware = (req, res, next) => {
  // Get token
  const token = req.headers.authorization && extractBearerToken(req.headers.authorization)

  // Check if exist
  if (!token) {
    return res.status(401).json({ message: 'Error. Need a token' })
  }

  // Check token validity
  jwt.verify(token, SECRET, (err, decodedToken) => {
    if (err) {
      res.status(401).json({ message: 'Error. Bad token' })
    } else {
      return next()
    }
  })
}

```

Ci-dessus la fonction checkTokenMiddleware permettant l'authentification sur les routes souhaitées

```

/**
 * Route list
 */

// Route to get info on our account (token decode)
app.get("/me", checkTokenMiddleware, async (req, res) => {

  const decoded = await decodeToken(req, res)

  return res.status(200).json({ content: decoded })
})

```

Ci-dessus un exemple d'utilisation de la fonction permettant de filtrer les requêtes

## 2 - Avantages et inconvénients de Node et Express

Avantages :

- Le 1er avantage à citer est le fait d'avoir eu un professeur excellent dans cette matière qui nous a donné goût à ce langage et qui nous a permis d'apprendre et d'évoluer de manière simple et efficace.
- Permet l'utilisation de scripts JavaScript côté serveur.
- Connaissant déjà JavaScript il permet de réaliser des applications rapidement.
- Node.js est très rapide dans son exécution.
- NPM manager est également un gros plus qui permet de gérer les dépendances pour une application.

Inconvénients :

- C'est une addiction, une fois qu'on y a goûté on a du mal à s'en passer
- Ce n'est pas très bien adapté pour travailler avec des BDD relationnelles.

## 3 - Comment avons nous développé et pourquoi ?

Nous avons utilisé les outils suivants afin de développer l'application :

- VSCode afin de développer l'application car c'est notre éditeur de code préféré.
- GitHub afin de collaborer efficacement et pouvoir avoir un aperçu sur l'état d'avancement global du projet
- Postman dans le but de tester notre API, logiciel simple et très efficace.
- MongoDB Compass pour pouvoir contrôler le contenu de notre base de donnée et effectuer des jeux de tests

## **4 - Les difficultés rencontrées**

Nous n'avons pas rencontré de difficultés réellement bloquantes étant donné que Paul avait déjà travaillé avec cette technologie lors de son alternance précédente il a pu aider Giovanni lorsqu'il bloquait sur certaines fonctionnalités, nous avons donc pu avancer de manière assez rapide et efficace en nous répartissant le travail de manière équitable.

## **5 - Conclusion**

En conclusion Giovanni a pu monter en compétences sur des technologies qu'il ne connaissait pas avec l'appui de Paul afin de le débloquent dans certaines situations, Paul a pu lui, approfondir ses connaissances. Le contexte / sujet de TP étant intéressant et correspondant à une application de notre quotidien, cela a pu éveiller notre curiosité et accroître notre implication pour ce projet.

Merci à notre professeur Sid pour ce module plus qu'intéressant et dans la bonne humeur créant une ambiance de travail motivante.