

---

# TECHNOS LOG

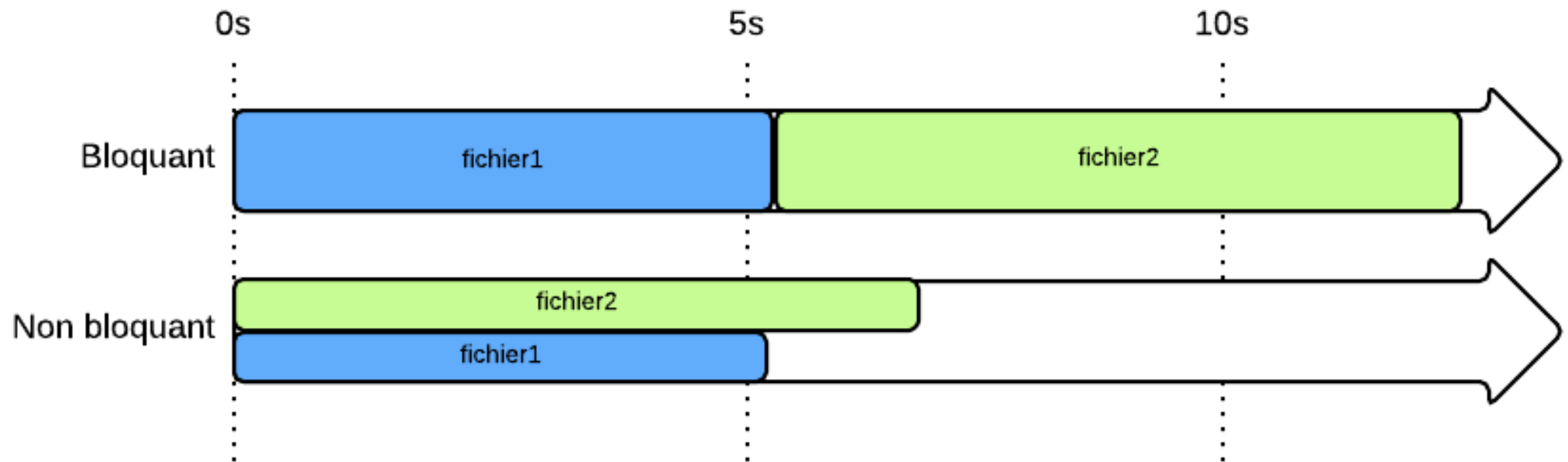
## 01.1 NODEJS - INTRODUCTION



# NODEJS ?

- Javascript coté serveur
- Basé sur le moteur V8 de google
- Bas niveau
- Ultra performant
- <https://nodejs.org/en/>

# MODÈLE NON BLOQUANT



# LES CALLBACKS

- Fonctions dans les fonctions
- Permet d'exécuter un code à la fin d'un autre code

```
/**
 * Load all user
 * @param req
 * @param res
 */
PersonController.getAll = function(req, res){
    Util.info('Load all user');
    Person.find({}).exec(function(err, results){
        if(err){
            res.status(400).json({message : "Error Loading Person"})
        }else{
            res.status(200).json(results)
        }
    })
};
```

# MODULE NODEJS

- Noyau node très faible donc besoin de module externe
- Utilisation *require* pour charger un module
- Utilisation *exports* pour partager les fonctions du module

```
Util = require('./app/helpers/appUtils');
```

```
'use strict';

//----- Controller Appointment
var appUtils = module.exports,
    moment = require('moment');

appUtils.info = function(message, jsonObj)
{
    console.log('[' + global.pid + '][ ' + moment().format('YYYY.MM.DD HH:mm:ss') + ' ] INFO: ' + message);

    if ( jsonObj != null )
        console.log(exports.inspect(jsonObj));
};
```

# CONSTRUCTION DE SERVEUR WEB

- Bas niveau donc c'est à vous de le faire ;)

```
var http = require('http');  
var server = http.createServer(function(req, res) {  
  res.writeHead(200);  
  res.end('Hello world');  
});  
server.listen(8080);
```

# NPM - NODE PACKAGE MANAGER

- <http://npmjs.org>
  - Site de référence des modules disponibles
- Présent dans le répertoire `node_modules`
- Ligne de commandes
  - `npm init`
  - `npm search mysql //` — Recherche des modules mysql
  - `npm install mysql //` — Installation du module
  - `npm install -g mysql //` — Installation du module de maniere globale
  - `npm update //` — Mise à jour
  - `npm install mysql --save //` — Installation + enregistrement apps



# MES PREMIERS PAS

LIVE SESSION – PLEASE STAY TUNE







SITE INTERNET AVEC NODEJS



## UTILISATION DES MIDDLEWARE EXPRESSJS

```
// Only use logger for development environment
if (process.env.NODE_ENV === 'development') {
  app.use(logger( format: 'dev'));
}

// assign the template engine to .html files
app.engine('html', consolidate[config.templateEngine]);

// set .html as the default extension
app.set('view engine', 'html');

// Set views path, template engine and default layout
app.set('views', config.publicPath + '/templates');
```

# FAIRE UN RENDU AVEC MUSTACHE

- `res.render(« templateUrl»,params)`

```
res.render('authByExternalCallSuccess',{
  redirectTo : currentConfig.states.success.link+'?salt='+salt,
  title      : currentConfig.states.success.title,
  message    : currentConfig.states.success.subTitle,
  refreshTime : currentConfig.states.success.refreshTime || 3
});
```

```
<h2>{{title}}</h2>

{{#message}}
<p>{{message}}</p>
{{/message}}

{{#params}}
<div class="code">
  <pre>{{params}}</pre>
</div>
{{/params}}
```



# MON SECOND PAS

LIVE SESSION – PLEASE STAY TUNE AGAIN

