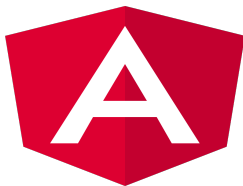


# Angular : quelques outils

**Achref El Mouelhi**

Docteur de l'université d'Aix-Marseille  
Chercheur en programmation par contrainte (IA)  
Ingénieur en génie logiciel

`elmouelhi.achref@gmail.com`



## 1 Génération de PDF

# Angular

## Génération de PDF : deux modules **NodeJS** nécessaires

- `html2canvas`
  - pour convertir le composant en canvas
  - <https://www.npmjs.com/package/html2canvas>
- `jsPDF`
  - pour convertir le canvas en PDF
  - <https://www.npmjs.com/package/jspdf>

# Angular

html2canvas : **installation**

```
npm install --save html2canvas
```

© Achref EL MOUL

# Angular

html2canvas : **installation**

```
npm install --save html2canvas
```

Vérifier qu'un répertoire `html2canvas` a bien été créé dans `node_modules`

# Angular

## jsPDF : installation

```
npm install --save html2canvas
```

© Achref EL MOU

# Angular

## jsPDF : installation

```
npm install --save html2canvas
```

Vérifier qu'un répertoire `jspdf` a bien été créé dans `node_modules`

# Angular

## Pour commencer

- créer un composant `pdf` dans le module `cours`
- créer un chemin `/pdf` permettant d'afficher ce composant

# Angular

Ajoutons le contenu suivant dans `pdf.component.html`

```
<div #pdf>
  <table>
    <tr>
      <th>Nom</th>
      <th>Prénom</th>
      <th>Ville</th>
    </tr>
    <tr>
      <td>Wick</td>
      <td>John</td>
      <td>Lyon</td>
    </tr>
    <tr>
      <td>Mitroglou</td>
      <td>Kostas</td>
      <td>Marseille</td>
    </tr>
    <tr>
      <td>Hakimi</td>
      <td>Achraf</td>
      <td>Paris</td>
    </tr>
  </table>
</div>

<button (click)="downloadPdf()">Télécharger</button>
```

# Angular

Dans `pdf.component.ts`, commençons par importer les deux modules installés

```
import { Component, OnInit } from '@angular/core';
import { jsPDF } from 'jspdf';
import html2canvas from 'html2canvas';

@Component({
  selector: 'app-pdf',
  templateUrl: './pdf.component.html',
  styleUrls: ['./pdf.component.css']
})
export class PdfComponent implements OnInit {

  constructor() { }

  ngOnInit(): void {
  }

  downloadPdf() {
  }
}
```

Utilisons ensuite `@ViewChild` pour récupérer l'élément à convertir en PDF

```
import { Component, OnInit, ViewChild, ElementRef } from '@angular/core';
import { jsPDF } from 'jspdf';
import html2canvas from 'html2canvas';

@Component({
  selector: 'app-pdf',
  templateUrl: './pdf.component.html',
  styleUrls: ['./pdf.component.css']
})
export class PdfComponent implements OnInit {

  @ViewChild('pdf', { static: false }) toPdf!: ElementRef;

  constructor() { }

  ngOnInit(): void {
  }

  downloadPdf() {
  }
}
```

# Angular

## Ensuite, on transforme

- 1 balise  $\Rightarrow$  canvas
- 2 canvas  $\Rightarrow$  image
- 3 image  $\Rightarrow$  pdf

# Angular

## Transformons la balise sélectionnée en canvas

```
import { Component, OnInit, ViewChild, ElementRef } from '@angular/core';
import { jsPDF } from 'jspdf';
import html2canvas from 'html2canvas';

@Component({
  selector: 'app-pdf',
  templateUrl: './pdf.component.html',
  styleUrls: ['./pdf.component.css']
})
export class PdfComponent implements OnInit {

  @ViewChild('pdf', { static: false }) toPdf!: ElementRef;

  constructor() { }

  ngOnInit(): void {
  }

  downloadPdf() {
    html2canvas(this.toPdf.nativeElement)
      .then(canvas => {

      });
  }
}
```

Construisons un objet `jsPDF` en préservant les dimensions du canvas et en choisissant l'orientation selon la largeur et la hauteur

```
import { Component, OnInit, ViewChild, ElementRef } from '@angular/core';
import { jsPDF } from 'jspdf';
import html2canvas from 'html2canvas';

@Component({
  selector: 'app-pdf',
  templateUrl: './pdf.component.html',
  styleUrls: ['./pdf.component.css']
})
export class PdfComponent implements OnInit {

  @ViewChild('pdf', { static: false }) toPdf!: ElementRef;

  constructor() { }
  ngOnInit(): void { }

  downloadPdf() {
    html2canvas(this.toPdf.nativeElement)
      .then(canvas => {
        const pdfWidth = canvas.width;
        const pdfHeight = canvas.height;
        var pdf;

        if (pdfWidth > pdfHeight) {
          pdf = new jsPDF('l', 'px', [pdfWidth, pdfHeight]); // l pour landscape
        } else {
          pdf = new jsPDF('p', 'px', [pdfWidth, pdfHeight]); // p pour portrait
        }
      });
  }
}
```

# Angular

Convertissons le canvas en image de type png (on peut choisir un autre format)

```
downloadPdf() {  
  html2canvas(this.toPdf.nativeElement)  
    .then(canvas => {  
      const pdfWidth = canvas.width;  
      const pdfHeight = canvas.height;  
      var pdf;  
  
      if (pdfWidth > pdfHeight) {  
        pdf = new jsPDF('l', 'px', [pdfWidth, pdfHeight]);  
      } else {  
        pdf = new jsPDF('p', 'px', [pdfWidth, pdfHeight]);  
      }  
  
      const imgData = canvas.toDataURL("image/png", 1.0);  
  
    });  
}
```

# Angular

Ajoutons l'image à notre objet jsPDF

```
downloadPdf() {  
  html2canvas(this.toPdf.nativeElement)  
    .then(canvas => {  
      const pdfWidth = canvas.width;  
      const pdfHeight = canvas.height;  
      var pdf;  
  
      if (pdfWidth > pdfHeight) {  
        pdf = new jsPDF('l', 'px', [pdfWidth, pdfHeight]);  
      } else {  
        pdf = new jsPDF('p', 'px', [pdfWidth, pdfHeight]);  
      }  
  
      const imgData = canvas.toDataURL("image/png", 1.0);  
      const width = pdf.internal.pageSize.getWidth();  
      const height = pdf.internal.pageSize.getHeight();  
      pdf.addImage(imgData, 0, 0, width, height);  
    });  
}
```

# Angular

Transformons l'image en fichier PDF (`file` sera le nom du fichier téléchargé)

```
downloadPdf() {  
  html2canvas(this.toPdf.nativeElement)  
    .then(canvas => {  
      const pdfWidth = canvas.width;  
      const pdfHeight = canvas.height;  
      var pdf;  
  
      if (pdfWidth > pdfHeight) {  
        pdf = new jsPDF('l', 'px', [pdfWidth, pdfHeight]);  
      } else {  
        pdf = new jsPDF('p', 'px', [pdfWidth, pdfHeight]);  
      }  
  
      const imgData = canvas.toDataURL("image/png", 1.0);  
      const width = pdf.internal.pageSize.getWidth();  
      const height = pdf.internal.pageSize.getHeight();  
      pdf.addImage(imgData, 0, 0, width, height);  
      pdf.save('file.pdf');  
    });  
}
```