Create Angular Cash Ticket

Pour créer et imprimer un ticket de caisse en utilisant Angular, vous pouvez suivre les étapes suivantes :

* Créez une nouvelle application Angular en utilisant la commande ***ng new***
* Créez un composant pour le ticket de caisse en utilisant la commande ***ng generate component ticket***
* Modifiez le Template HTML du composant de ticket pour inclure les informations que vous souhaitez afficher sur le ticket, telles que les articles achetés, le total, les taxes, etc.
* Ajoutez des styles CSS au composant pour donner au ticket un aspect professionnel et cohérent.
* Dans le contrôleur du composant, utilisez le code pour remplir les données du ticket en utilisant des données fournies par l’application parente ou une API REST.
* Utilisez la fonction JavaScript window.print() pour imprimer le ticket lorsque vous en avez besoin.

Voici un exemple de code pour un composant de ticket :

<div class="ticket">

  <h1 class="ticket-header">Ticket de caisse</h1>

  <table class="ticket-table">

    <tr \*ngFor="let item of items">

      <td>{{ item.name }}</td>

      <td class="right">{{ item.price | currency }}</td>

    </tr>

  </table>

  <div class="ticket-total">

    Total: {{ total | currency }}

  </div>

</div>

<style>

.ticket {

  width: 80mm;

  font-family: Arial, sans-serif;

}

.ticket-header {

  text-align: center;

}

.ticket-table {

  width: 100%;

  border-collapse: collapse;

}

.ticket-table td {

  border: 1px solid #ccc;

  padding: 5px;

}

.right {

  text-align: right;

}

.ticket-total {

  text-align: right;

  margin-top: 20px;

  font-weight: bold;

}

</style>

Pour imprimer le ticket de caisse en utilisant Angular, vous pouvez ajouter un bouton dans le template HTML du composant de ticket, qui déclenchera la fonction JavaScript window.print() lorsqu’il est cliqué. Voici un exemple de code :

<div class=”ticket”>

  <!—Code HTML du composant de ticket ◊

</div>

<button (click)=”printTicket()”>Imprimer</button>

Et voici un exemple de code pour le contrôleur du composant de ticket :

Import { Component, OnInit } from ‘@angular/core’;

@Component({

  Selector: ‘app-ticket’,

  templateUrl: ‘./ticket.component.html’,

  styleUrls: [‘./ticket.component.css’]

})

Export class TicketComponent implements OnInit {

  // Code pour remplir les données du ticket

  Constructor() { }

  ngOnInit() {

  }

  printTicket() {

    window.print();

  }

}

Lorsque vous cliquez sur le bouton “Imprimer”, la fonction printTicket sera appelée et déclenchera l’impression du ticket de caisse. Vous pouvez ajuster la mise en page du ticket en modifiant les styles CSS pour qu’ils s’affichent correctement sur le papier.

Le composant ticket serait le composant qui affiche les informations du ticket de caisse. Il peut inclure des informations telles que les articles achetés, le total, les taxes, etc. et sera généralement formaté en utilisant du HTML et du CSS.

Voici un exemple de code pour le composant ticket :

Import { Component, Input } from ‘@angular/core’;

@Component({

  Selector: ‘app-ticket’,

  Template: `

    <div class=”ticket”>

      <h1 class=”ticket-header”>Ticket de caisse</h1>

      <table class=”ticket-table”>

        <tr \*ngFor=”let item of items”>

          <td>{{ item.name }}</td>

          <td class=”right”>{{ item.price | currency }}</td>

        </tr>

      </table>

      <div class=”ticket-total”>

        Total: {{ total | currency }}

      </div>

    </div>

  `,

  Styles: [`

    .ticket {

      Width: 80mm;

      Font-family: Arial, sans-serif;

    }

    .ticket-header {

      Text-align: center;

    }

    .ticket-table {

      Width: 100%;

      Border-collapse: collapse;

    }

    .ticket-table td {

      Border: 1px solid #ccc;

      Padding: 5px;

    }

    .right {

      Text-align: right;

    }

    .ticket-total {

      Text-align: right;

      Margin-top: 20px;

      Font-weight: bold;

    }

  `]

})

Export class TicketComponent {

  @Input() items;

  @Input() total;

}

Dans ce code, nous utilisons le système de modèle intégré d’Angular pour définir le template HTML du composant, ainsi que les styles CSS associés. Nous utilisons également la directive \*ngFor pour boucler sur les articles et les afficher dans un tableau HTML. Le composant prend en entrée les articles (items) et le total (total) à afficher sur le ticket, qui sont passés en tant qu’attributs lors de l’utilisation du composant dans une autre partie de l’application.

import { Component, Input } from '@angular/core';

@Component({

  selector: 'app-ticket',

  template: `

    <div class="ticket">

      <h1 class="ticket-header">Ticket de caisse</h1>

      <table class="ticket-table">

        <tr \*ngFor="let item of items">

          <td>{{ [item.name](http://item.name/) }}</td>

          <td class="right">{{ item.price | currency }}</td>

        </tr>

      </table>

      <div class="ticket-total">

        Total: {{ total | currency }}

      </div>

    </div>

  `,

  styles: [`

    .ticket {

      width: 80mm;

      font-family: Arial, sans-serif;

    }

    .ticket-header {

      text-align: center;

    }

    .ticket-table {

      width: 100%;

      border-collapse: collapse;

    }

    .ticket-table td {

      border: 1px solid #ccc;

      padding: 5px;

    }

    .right {

      text-align: right;

    }

    .ticket-total {

      text-align: right;

      margin-top: 20px;

      font-weight: bold;

    }

  `]

})

export class TicketComponent {

  @Input() items;

  @Input() total;

  ngOnInit() {

    this.items = [ { name: 'Article 1', price: 10.99 }, { name: 'Article 2', price: 5.49 }, { name: 'Article 3', price: 3.99 } ];

    this.total = this.items.reduce((acc, item) => acc + item.price, 0);

  }

}