IndoSwap Angular - Exact Copy-Paste Guide

📁 File 1: Create Token Interface

File: `src/app/models/token.interface.ts` **Action:** Create this file and copy this code:

```typescript
export interface Token {
  symbol: string;
  name: string;
  icon: string;
  balance: number;
  address?: string;
}

export interface SwapData {
  fromToken: Token;
  toToken: Token;
  fromAmount: number;
  toAmount: number;
  exchangeRate: number;
  priceImpact: number;
  tradingFee: number;
  minimumReceived: number;
}

export interface NotificationData {
  message: string;
  type: 'success' | 'error' | 'info';
  duration?: number;
}
```

🔧 File 2: Wallet Service

File: `src/app/services/wallet.service.ts` **Action:** Replace the entire content with:

typescript

```typescript
import { Injectable } from '@angular/core';
import { BehaviorSubject, Observable } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class WalletService {
  private isConnectedSubject = new BehaviorSubject<boolean>(false);
  private addressSubject = new BehaviorSubject<string>('');

  isConnected$: Observable<boolean> = this.isConnectedSubject.asObservable();
  address$: Observable<string> = this.addressSubject.asObservable();

  async connectWallet(): Promise<void> {
    return new Promise((resolve) => {
      setTimeout(() => {
        this.isConnectedSubject.next(true);
        this.addressSubject.next('0x1234...5678');
        resolve();
      }, 1000);
    });
  }

  disconnectWallet(): void {
    this.isConnectedSubject.next(false);
    this.addressSubject.next('');
  }

  get isConnected(): boolean {
    return this.isConnectedSubject.value;
  }

  get address(): string {
    return this.addressSubject.value;
  }
}
```

💱 File 3: Swap Service

File: `src/app/services/swap.service.ts` **Action:** Replace the entire content with:

typescript

```typescript
import { Injectable } from '@angular/core';
import { BehaviorSubject, Observable } from 'rxjs';
import { Token, SwapData } from '../models/token.interface';

@Injectable({
  providedIn: 'root'
})
export class SwapService {
  private exchangeRates: { [key: string]: number } = {
    'BNB-BUSD': 285.43,
    'BUSD-BNB': 0.0035,
    'BNB-USDT': 284.87,
    'USDT-BNB': 0.0035,
    'BUSD-USDT': 0.998,
    'USDT-BUSD': 1.002
  };

  private defaultTokens: Token[] = [
    { symbol: 'BNB', name: 'Binance Coin', icon: 'bnb', balance: 2.45 },
    { symbol: 'BUSD', name: 'Binance USD', icon: 'busd', balance: 1250.00 },
    { symbol: 'USDT', name: 'Tether USD', icon: 'usdt', balance: 500.00 }
  ];

  private swapDataSubject = new BehaviorSubject<SwapData>({
    fromToken: this.defaultTokens[0],
    toToken: this.defaultTokens[1],
    fromAmount: 0,
    toAmount: 0,
    exchangeRate: this.exchangeRates['BNB-BUSD'],
    priceImpact: 0.01,
    tradingFee: 0.25,
    minimumReceived: 0
  });

  swapData$: Observable<SwapData> = this.swapDataSubject.asObservable();

  getTokens(): Token[] {
    return this.defaultTokens;
  }

  calculateSwap(fromAmount: number): SwapData {
    const currentData = this.swapDataSubject.value;
    const rateKey = `${currentData.fromToken.symbol}-${currentData.toToken.symbol}`;
    const exchangeRate = this.exchangeRates[rateKey] || 1;
    const toAmount = fromAmount * exchangeRate;
    const minimumReceived = toAmount * 0.995;
    const newSwapData: SwapData = {
      ...currentData,
      fromAmount,
      toAmount,
      exchangeRate,
      minimumReceived
    };

    this.swapDataSubject.next(newSwapData);
    return newSwapData;
  }

  swapTokens(): void {
    const currentData = this.swapDataSubject.value;
    const newSwapData: SwapData = {
      ...currentData,
      fromToken: currentData.toToken,
      toToken: currentData.fromToken,
      fromAmount: 0,
      toAmount: 0,
      exchangeRate: this.exchangeRates[`${currentData.toToken.symbol}-${currentData.fromToken.s
      minimumReceived: 0
    };

    this.swapDataSubject.next(newSwapData);
  }

  async executeSwap(swapData: SwapData): Promise<boolean> {
    return new Promise((resolve) => {
      setTimeout(() => {
        resolve(true);
      }, 2000);
    });
  }

  get currentSwapData(): SwapData {
    return this.swapDataSubject.value;
  }
}
```

🔔 File 4: Notification Service

File: `src/app/services/notification.service.ts` **Action:** Replace the entire content with:

```typescript
import { Injectable } from '@angular/core';
import { BehaviorSubject, Observable } from 'rxjs';
import { NotificationData } from '../models/token.interface';

@Injectable({
  providedIn: 'root'
})
export class NotificationService {
  private notificationSubject = new BehaviorSubject<NotificationData | null>(null);
  notification$: Observable<NotificationData | null> = this.notificationSubject.asObservable();

  showNotification(message: string, type: 'success' | 'error' | 'info' = 'info', duration: numb
    const notification: NotificationData = { message, type, duration };
    this.notificationSubject.next(notification);

    setTimeout(() => {
      this.notificationSubject.next(null);
    }, duration);
  }
}
```

🏠 File 5: App Component

File: `src/app/app.component.ts` **Action:** Replace the entire content with:

```typescript
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent {
  title = 'indoswap-angular';
}
```

File: `src/app/app.component.html` **Action:** Replace the entire content with:

```html
<div class="background-animation">
  <div class="bg-circle"></div>
  <div class="bg-circle"></div>
  <div class="bg-circle"></div>
</div>

<app-header></app-header>

<div class="container">
  <app-hero></app-hero>
  <app-stats></app-stats>
  <app-swap></app-swap>
  <app-features></app-features>
</div>

<app-footer></app-footer>
<app-notification></app-notification>
```

File: `src/app/app.component.scss` **Action:** Create this file and add:

scss

```scss
.background-animation {
  position: fixed;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  z-index: -1;
  opacity: 0.1;
}

.bg-circle {
  position: absolute;
  border-radius: 50%;
  background: rgba(255, 255, 255, 0.1);
  animation: float 6s ease-in-out infinite;

  &:nth-child(1) {
    width: 80px;
    height: 80px;
    top: 20%;
    left: 10%;
    animation-delay: -2s;
  }

  &:nth-child(2) {
    width: 120px;
    height: 120px;
    top: 60%;
    right: 10%;
    animation-delay: -4s;
  }

  &:nth-child(3) {
    width: 60px;
    height: 60px;
    top: 80%;
    left: 80%;
    animation-delay: -1s;
  }
}

@keyframes float {
  0%, 100% { transform: translateY(0px) rotate(0deg); }
  50% { transform: translateY(-20px) rotate(180deg); }
}
```

```
.container {
  max-width: 1200px;
  margin: 0 auto;
  padding: 2rem;

  @media (max-width: 768px) {
    padding: 1rem;
  }
}
```

📦 File 6: App Module

File: `src/app/app.module.ts` **Action:** Replace the entire content with:

typescript

```typescript
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { FormsModule } from '@angular/forms';
import { CommonModule } from '@angular/common';

import { AppComponent } from './app.component';
import { HeaderComponent } from './components/header/header.component';
import { HeroComponent } from './components/hero/hero.component';
import { StatsComponent } from './components/stats/stats.component';
import { SwapComponent } from './components/swap/swap.component';
import { FeaturesComponent } from './components/features/features.component';
import { FooterComponent } from './components/footer/footer.component';
import { NotificationComponent } from './components/notification/notification.component';

@NgModule({
  declarations: [
    AppComponent,
    HeaderComponent,
    HeroComponent,
    StatsComponent,
    SwapComponent,
    FeaturesComponent,
    FooterComponent,
    NotificationComponent
  ],
  imports: [
    BrowserModule,
    FormsModule,
    CommonModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

🌐 File 7: Global Styles

File: `src/styles.scss` **Action:** Replace the entire content with:

scss

```scss
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

html, body {
  height: 100%;
  margin: 0;
  padding: 0;
}

body {
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, sans-serif;
  background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
  min-height: 100vh;
  color: white;
  overflow-x: hidden;
}
```

✋ STOP HERE AND TEST

Before continuing with components, let's test what we have:

bash

```bash
npm install
ng serve
```

If everything works, you should see a basic page with the background gradient. Continue to the next section for components.

🧩 Next: Copy All Components (Part 2)

Ready for the components? Let me know when you've completed Part 1 and I'll give you the exact component code to copy!