Edgar Chiquillo

# Bindings & Events

Requisitos:

1. Mostrar todos los objetos del array "serverCharacters" en el html.
2. Por defecto no se puede editar ningún objeto del array.
3. Al dar al botón "Edit" te permite editar todos los campos del objeto y automáticamente cambia el nombre a "Save" del botón.
4. Además al dar al botón "Edit" se mostrará en formato texto el objeto correspondiente.
5. Al dar al botón "Save" se guardan los cambios en la variable pertinente.
6. Además al dar al botón "Save" se mostrará en formato texto el objeto correspondiente con las modificaciones hechas.
7. Tenéis que usar los conceptos siguientes: Events & Two Way Binding.
8. + Modificaciones

## Código HTML

```html
<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-widht, initial-scale=1.0">
    <title>Bindings & Events</title>
    <link rel="stylesheet" href="app.component.css">
</head>

<div class="card-container">
    <div *ngFor="let character of serverCharacters; let i = index" class="card">
        <div *ngIf="!character.editable; else editMode">
            <h4>Name:</h4>
            <p>{{ character.name }}</p>

            <div *ngIf="showDetails[i]">
                <!-- Mostrar detalles aquí -->
                <p>Strength: {{ character.strength }}</p>
                <p>Agility: {{ character.agility }}</p>
                <p>Intelligence: {{ character.intelligence }}</p>
                <p>Life: {{ character.life }}</p>
                <button (click)="editCharacter(character)">Edit</button>
            </div>
            <button (click)="showCharacterDetails(i)">
                {{ showDetails[i] ? 'Hide' : 'Show' }} <!-- Cambio de texto del botón según el estado -->
            </button>
        </div>
        <ng-template #editMode>
            <h2>Edit </h2>
            <input [(ngModel)]="character.name" />
            <input [(ngModel)]="character.strength" />
            <input [(ngModel)]="character.agility" />
            <input [(ngModel)]="character.intelligence" />
            <input [(ngModel)]="character.life" />
            <button (click)="saveCharacter(character)">Save</button>
            <button (click)="showCharacterDetails(i)">
                {{ showDetails[i] ? 'Hide' : 'Show' }} <!-- Cambio de texto del botón según el estado -->
            </button>
        </ng-template>
    </div>
</div>

</html>
```

## Código TS

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

interface Character {
  name: string;
  strength: number;
  agility: number;
  intelligence: number;
  life: number;
  editable?: boolean;
}

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {

  serverCharacters: Character[] = [];
  character: any;
  showDetails: { [key: number]: boolean } = {};

  constructor() {
    // Ejemplo de respuesta de un servidor en formato JSON
    const serverJson = `[
      {"name": "Yugui", "strength": 18, "agility": 11, "intelligence": 15, "life": 30 },
      {"name": "Jaden", "strength": 14, "agility": 8, "intelligence": 20, "life": 20 },
      {"name": "Yusei", "strength": 18, "agility": 18, "intelligence": 18, "life": 40 },
      {"name": "Yuma", "strength": 10, "agility": 20, "intelligence": 8, "life": 18 },
      {"name": "Yusaku", "strength": 18, "agility": 6, "intelligence": 16, "life": 34 }
    ]`;

    // Parseamos la información y la convertimos directamente en un array de "Character"
    this.serverCharacters = JSON.parse(serverJson);
  }

  editCharacter(character: Character) {
    character.editable = true;
  }

  saveCharacter(character: Character) {
    character.editable = false;
  }

  showCharacterDetails(index: number) {
    this.showDetails[index] = !this.showDetails[index];
  }
}
```

## Código Module TS

src > app > TS app.module.ts > ...

```ts
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppComponent } from './app.component';
import { FormsModule } from '@angular/forms';

@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    FormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }

```

## CSS

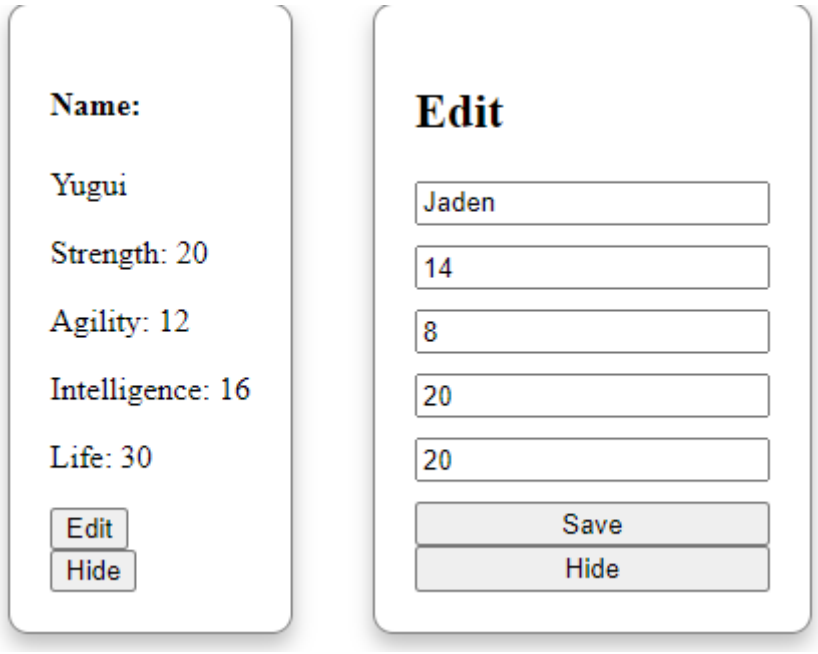
```css
.card-container {
  display: flex;
  flex-wrap: wrap;
}

.card {
  display: grid;
  margin: 20px;
  padding: 20px;
  border: 0.5px solid gray;
  border-radius: 10px;
  box-shadow: 0px 6px 10px rgba(0, 0, 0, 0.25);
}

.card>input {
  margin-bottom: 10px;
}
```

## Resultado


Name:
Yugui
Strength: 20
Agility: 12
Intelligence: 16
Life: 30
Edit
Hide
Name:
Jaden
Strength: 14
Agility: 8
Intelligence: 20
Life: 20
Edit
Hide
Name:
Yusei
Show
Name:
Yuma
Show
Name:
Yusaku
Show


Name:
Yugui
Strength: 20
Agility: 12
Intelligence: 16
Life: 30
Edit
Hide
Edit
Jaden
14
8
20
20
Save
Hide

**Name:**

Yugui

Show

**Name:**

Jaden

Strength: 17

Agility: 20

Intelligence: 25

Life: 25

Edit

Hide