

Esercitazione 4

Studente: Michele Luigi Greco - s256665

Comandi da lanciare nella directory principale

- Run `npm install`
- Run `npm start`

La pagina è adesso disponibile alla url: localhost:4200

Struttura dati

- Studenti presenti nella tabella visualizzata a schermo:

```
const STUDENT_DATA: Student[] = [
  { id: 's25001', group: 'team1', name: 'Hydrogen', firstName: 'Santini' },
  { id: 's25002', group: 'team2', name: 'Helium', firstName: 'Salta' },
  { id: 's25003', group: 'team3', name: 'Lithium', firstName: 'Marchio' },
  { id: 's25004', group: 'team1', name: 'Beryllium', firstName: 'Greco' },
  { id: 's25005', group: 'team2', name: 'Boron', firstName: 'Caso' },
  { id: 's25006', group: 'team3', name: 'Carbon', firstName: 'Bruno' },
  { id: 's25007', group: 'team', name: 'Nitrogen', firstName: 'Ferrero' },
  { id: 's25008', group: 'team', name: 'Oxygen', firstName: 'Moretti' },
  { id: 's25009', group: '<none>', name: 'Fluorine', firstName: 'De Sanctis' },
  { id: 's25010', group: '<none>', name: 'Neon', firstName: 'Lorenzi' }];
```

- Studenti presenti come option del campo autocomplete:

```
const options: Student[] = [
  { id: 's299001', group: 'team2' , name: 'New_Hydrogen', firstName: 'New_Santini' },
  { id: 's299002', group: 'team3' , name: 'New_Helium', firstName: 'New_Salta' },
  { id: 's299003', group: 'team5' , name: 'New_Lithium', firstName: 'New_Marchio' },
  { id: 's299004', group: 'team6' , name: 'New_Beryllium', firstName: 'New_Greco' },
  { id: 's299005', group: 'team7' , name: 'New_Boron', firstName: 'New_Caso' },
  { id: 's299006', group: 'team3' , name: 'New_Carbon', firstName: 'New_Bruno' },
  { id: 's299007', group: 'team9' , name: 'New_Nitrogen', firstName: 'New_Ferrero' },
  { id: 's299008', group: 'team7' , name: 'New_Oxygen', firstName: 'New_Moretti' },
  { id: 's299009', group: 'team6' , name: 'New_Fluorine', firstName: 'New_De Sanctis' },
  { id: 's299010', group: 'team0' , name: 'New_Neon', firstName: 'New_Lorenzi' }];
```

- Array che contiene gli studenti selezionati nella tabella:

```
selection = new SelectionModel<Student>(true, []);
```

Ogni volta che viene selezionato/deselezionato uno studente dalla tabella, viene aggiunto/rimosso l'object studente dall'array `selection.selected`

- Array che contiene il titolo delle colonne da visualizzare:

```
columnsToDisplay: string[] = ['select', 'id', 'name', 'firstName', 'group'];
```

- Dichiarazione della tabella e inizializzazione del datasource fittizio

```
@ViewChild('MatTable')
table: MatTable<Student>;
dataSource = new MatTableDataSource<Student>(STUDENT_DATA);
```

CheckBox Master

La checkox master è composta da tre stadi:

- Stato iniziale: `[]`

Nessuno studente è selezionato, l'array `selection.selected` è vuoto.

- Stato intermedio: `[-]`

Uno o più studenti selezionati, ma non tutti. Gli object student corrispondenti agli studenti selezionati vengono aggiunti all'array `selection.selected`

- Stato finale : [V]

Al click sulla checkbox master vengono selezionati/deselezionati tutti gli studenti della tabella. Gli Object Student di questi studenti vengono aggiunti/rimossi all'array `selection.selected`. Lo stato della checkbox master è ALL CHECKED.

Eliminare uno studente dalla tabella:

1. Selezioniamo i primi 2 studenti della tabella.
2. 2 Object di tipo student venno inseriti in `selection.selected`, che corrispondono agli studenti selezionati.
3. E' possibile cliccare sul button "Delete Student" per eliminare gli studenti. Il metodo invocato è il seguente:

```
deleteStudents() {
  const newDS: Student[] = [];
  this.dataSource.data.forEach(s => {
    if (!this.selection.isSelected(s)) {
      newDS.push(s);
    }
  });
  delete this.dataSource;
  this.dataSource = new MatTableDataSource<Student>(newDS);
  this.selection.clear();
}
```

Aggiungere uno studente alla tabella:

1. Per l'inserimento di uno studente si fa uno dell'apposito text-box dotato di autocompletamento, ossia durante la scrittura del nome dello studente viene applicato un filtro dinamico che mostrerà soltanto gli studenti il cui nome combacia, almeno parzialmente, con quanto digitato finora.
2. Cliccando su uno studente di quelli disponibili nel menu verrà caricato l'object student in una variabile temporanea presente in `app.component : student : Student`. Ciò avviene tramite l'invocazione del metodo:

```
addStudent(option: Student) {
  this.student = option;
}
```

3. Nel momento in cui verrà cliccato il bottone `Add Student` l'object student verrà pushato nel `dataSource` tramite l'invocazione del metodo:

```
addStudentToDB() {
  if (!this.dataSource.data.includes(this.student)) {
    STUDENT_DATA.push(this.student);
    this.dataSource._updateChangeSubscription();
    const index = options.indexOf(this.student, 0);
    if (index > -1) {
      options.splice(index, 1);
    }
  }
}
```

Si noti che qualora venisse premuto più volte il button "Add Student" nessuna modifica verrebbe effettuata, in quanto il controllo preliminare `!this.dataSource.data.includes(this.student)` garantisce che lo studente non sia già presente all'interno della tabella.