

# Lab4

Studente: Paola Caso - s257009

## Comandi (da eseguire nella directory principale)

1. Run `npm install`
2. Run `npm start`

La pagina è ora disponibile alla URL: <http://localhost:4200>

## Struttura dati

- Tabella studenti di un corso e datasource:

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

- Studenti presenti nella tabella visualizzata a schermo:

```
const STUDENTS: Student[] = [
  {id: 's257009', name: 'Caso', firstName: 'Paola', group: 'TeamProva'},
  {id: 's256665', name: 'Michele', firstName: 'Greco', group: 'TeamProva'},
  {id: 's238906', name: 'Bruno', firstName: 'Alberto', group: 'TeamProva'}];
```

- Studenti presenti nella lista di options dell'autocomplete:

```
const OPTIONS: Student[] = [
  {id: 's253309', name: 'Burlacu', firstName: 'Iustin', group: ''},
  {id: 's247948', name: 'Buffo', firstName: 'Matteo', group: ''},
  {id: 's222767', name: 'Massafra', firstName: 'Christian', group: 'TeamProva2'},
  {id: 's236564', name: 'Lamberti', firstName: 'Giovanni', group: ''},
  {id: 's378748', name: 'Agosta', firstName: 'Anna', group: ''}];
```

- Array che contiene gli studenti selezionati della tabella:

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

Ogni volta che viene selezionato/deselezionato uno studente nella tabella, l'oggetto `Student` corrispondente viene aggiunto/rimosso all'/dall' array `selection.selected`.

- Array che contiene l'header delle colonne della tabella da visualizzare:

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

## Checkbox master

La checkbox master è composta da tre stadi:

1. Stato iniziale: `[ ]` : nessuno studente è selezionato, l'array `selection.selected` è vuoto.
2. Stato indeterminato: `[-]` : qualche studente è selezionato, l'array `selection.selected` contiene uno o più oggetti `Student` (corrispondenti agli studenti selezionati).
3. Stato finale: `[V]` : se la checkbox master viene cliccata, vengono di conseguenza selezionati/deselezionati tutti gli studenti della tabella. Nel primo caso l'array `selection.selected` contiene tutti gli studenti della tabella, e lo stato della checkbox master è 'ALL CHECKED'.

## Eliminare uno studente dalla tabella

1. Selezioniamo i primi due studenti della tabella.
2. Due oggetti di tipo `Student`, corrispondenti agli studenti selezionati, vengono inseriti in `selection.selected`.
3. È possibile cliccare sul bottone "Delete selected" per eliminare dalla tabella gli studenti selezionati. Il metodo invocato è il seguente:

```

deleteSelected(){

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

```

## Aggiungere uno studente alla tabella:

1. Per inserire un nuovo studente nella tabella si utilizza l'apposito text-form dotato di autocompletamento: durante la scrittura del nome dello studente viene applicato un filtro dinamico che mostrerà soltanto quegli studenti che contengono nel proprio `name` la stringa digitata.
2. Cliccando su uno studente tra quelli presenti nella lista a tendina, verrà salvato l'oggetto `Student` selezionato in una variabile di appoggio presente in `app.component.ts`: `studentToAdd: Student`. Ciò avviene mediante l'invocazione del metodo:

```

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

```

dove `student` è l'opzione scelta nella lista.

3. Nel momento in cui verrà cliccato il bottone `Add`, l'oggetto `Student` verrà effettivamente aggiunto al `dataSource` della tabella, ed eliminato dall'array di `options` della lista, tramite l'invocazione del metodo:

```

commitStudent() {
    if (this.studentToAdd != null && !this.students.data.includes(this.studentToAdd)) {
        this.students.data.push(this.studentToAdd);
        const index = OPTIONS.indexOf(this.studentToAdd, 0);
        if (index > -1) {
            OPTIONS.splice(index, 1);
        }
        this.students._updateChangeSubscription();
    }
}

```

Si noti che, qualora si cliccasse più volte sul bottone `Add`, nessuna modifica verrebbe effettuata, in quanto il controllo preliminare `!this.students.data.includes(this.studentToAdd)` garantisce che lo studente non sia già presente all'interno della tabella.