

# 7. HTTP

## HTTP

- GET
- POST
- PUT
- DELETE

## REST server

- ➡ getAll, get(id)
- ➡ postNew
- ➡ putExisting, putExistingOrNew
- ➡ delete(id)

## module

```
import { HttpClientModule }  
from '@angular/common/http';  
...  
imports: [  
  ...  
  HttpClientModule  
],
```

## costanti

```
export const C_GRANDEZZA_TITOLO: string = '80px';  
export const REST_URL_GET_ALL: string = 'http://localhost:8080/rest/getAll';  
export const REST_URL_POST_NEW: string = 'http://localhost:8080/rest/postNew';
```

# 7. HTTP

## RESTful server

- Scaricare ed installare Java 8+ <https://www.oracle.com>

- VERIFICA

```
java -version
```



v1.8 o superiore



- Scaricare il server rest

[https://github.com/softwarelma/binaries/tree/master/ers\\_boot](https://github.com/softwarelma/binaries/tree/master/ers_boot)

- Avviare il server rest

```
java -jar ers_boot-0.0.3.jar
```

- VERIFICA

<http://localhost:8080/rest/getAll>



## 7. HTTP

Lavoriamo in modalità **CORS (Cross-Origin Resource Sharing)**

Nel caso di Chrome/Chromium esiste un'ulteriore possibilità di blocco, **CORB (Cross-Origin Read Blocking)**

---

Headers del server rest

```
httpServletResponse.setHeader("Access-Control-Allow-Methods", "GET, POST, DELETE, PUT");  
httpServletResponse.setHeader("Access-Control-Allow-Origin", "*");
```

oppure

```
@CrossOrigin(origins = "http://localhost:4200")
```

---

altrimenti aggiungere nel browser un plugin/extension per  
**Access-Control-Allow-Origin**

# 7. HTTP

## COSA FAREMO?

Seguono delle modifiche al DAO più 4 componenti

- visualizza
- cerca
- riepilogo
- edit

Gli snippets contengono  
i sorgenti completi

## 7. HTTP

service articolo  
dao, 1/6

```
import { FiltroArticoloType } from '../types/filtro-articolo.type';
import { Injectable } from '@angular/core';
import { Articolo } from '../classes/articolo';
import { HttpClient, HttpHeaders, HttpParams } from
'@angular/common/http';
import { Observable, throwError } from 'rxjs';
import { catchError, map } from 'rxjs/operators';
import { REST_URL_GET_ALL, REST_URL_POST_NEW } from
'../consts/costanti';
@Injectable({
  providedIn: 'root'
})
export class ArticoloDaoService {
  private arrayArticoli: Articolo[];
  constructor(private httpClient: HttpClient) { }
  private handleError(error: any) {
    alert(error.message || error);
    return throwError(error.message || error);
  }
}
```

## 7. HTTP

service articolo  
dao, 2/6

```
getArticoloById(id: number): Articolo {  
    if (this.arrayArticoli == null) { return null; }  
    for (const articolo of this.arrayArticoli) {  
        if (articolo.id === id) { return articolo; }  
    }  
    return null;  
}  
  
getArticoloHome(component: any) {  
    this.getArrayArticoliFromRestProducer().subscribe(  
        response => {  
            component.articolo = this.arrayArticoli.length === 0 ?  
                null : this.arrayArticoli[this.arrayArticoli.length - 1];  
            component.articolo = component.articolo == null ?  
                new Articolo() : component.articolo;  
        }  
    );  
}  
  
getArrayArticoli(component: any) {  
    this.getArrayArticoliFromRestProducer().subscribe(  
        response => { component.arrayArticoli = this.arrayArticoli; }  
    );  
}
```

## 7. HTTP

service articolo  
dao, 3/6

```
getArrayArticoliFromRestProducer(): Observable<Articolo[]> {  
    return this.httpClient.get<Articolo[]>(REST_URL_GET_ALL)  
        .pipe(  
            map(  
                (response: any) => {  
                    let listArticolo: Articolo[];  
                    if (response.error == null) {  
                        listArticolo = response.listArticolo;  
                    } else {  
                        listArticolo = [];  
                        alert(response.error);  
                    }  
                    return this.fixInstances(listArticolo);  
                },  
                catchError(this.handleError)  
            )  
        );  
}
```

## 7. HTTP

service articolo  
dao, 4/6

```
getArrayArticoliFiltrati(filtro: FiltroArticoloType): Array<Articolo> {  
    const arrayArticoliFiltrati: Array<Articolo> = [];  
    const filtroLower: string = filtro.testo.toLowerCase().toString();  
    for (const articolo of this.arrayArticoli) {  
        if (articolo.titolo.toLowerCase().includes(filtroLower)) {  
            arrayArticoliFiltrati.push(articolo);  
        } else if (articolo.arraySottotitolo[1].toLowerCase().includes(filtroLower)) {  
            arrayArticoliFiltrati.push(articolo);  
        } else if (articolo.testo.toLowerCase().includes(filtroLower)) {  
            arrayArticoliFiltrati.push(articolo);  
        }  
    }  
    return arrayArticoliFiltrati;  
}
```



## 7. HTTP

service articolo  
dao, 5/6

```
fixInstances(arrayArticoliJson: Articolo[]): Articolo[] {  
    const arrayArticoli: Array<Articolo> = [];  
    let obj: any;  
    for (obj of arrayArticoliJson) {  
        const articolo: Articolo = this.newInstance(obj);  
        arrayArticoli.push(articolo);  
    }  
    this.arrayArticoli = arrayArticoli;  
    return this.arrayArticoli;  
}  
  
newInstance(obj: any): Articolo {  
    const articolo: Articolo = new Articolo();  
    articolo.id = obj.id;  
    articolo.titolo = obj.titolo;  
    articolo.immagine = obj.immagine;  
    articolo.nomeImmagine = obj.nomeImmagine;  
    articolo.arraySottotitolo = obj.arraySottotitolo;  
    articolo.testo = obj.testo;  
    articolo.dataPubblicazione = obj.dataPubblicazione;  
    return articolo;  
}
```

## 7. HTTP

service articolo  
dao, 6/6

```
postNewConsumer(articoloDaInserire: Articolo) {  
    this.postNewProducer(articoloDaInserire).subscribe(  
        (articoloInserito: Articolo) => {  
            const articolo: Articolo = this.newInstance(articoloInserito);  
            this.arrayArticoli.push(articolo);  
            alert('Articolo inserito con successo');  
        }  
    );  
}  
  
postNewProducer(articoloDaInserire: Articolo): Observable<Articolo> {  
    return this.httpClient.post<Articolo>(  
        REST_URL_POST_NEW,  
        articoloDaInserire  
    ).pipe(  
        map(  
            (articoloInserito: any) => articoloInserito.articolo,  
            catchError(this.handleError)  
        )  
    );  
}
```

## 7. HTTP

comp  
visualizza,  
1/3

```
import { ArticoloDaoService } from '../services/articolo-dao.service';
import { Component, OnInit, Input } from '@angular/core';
import { C_GRANDEZZA_TITOLO } from '../consts/costanti';
import { Articolo } from '../classes/articolo';
import { ActivatedRoute } from '@angular/router';

@Component({
  selector: 'app-visualizza',
  templateUrl: './visualizza.component.html',
  styleUrls: ['./visualizza.component.css']
})
export class VisualizzaComponent implements OnInit {
  @Input() articolo: Articolo;

  stileTitolo = { 'font-size': C_GRANDEZZA_TITOLO, 'font-weight': 'bold',
    'margin': '0' };

  classeSottotitolo: string = 'sottotitolo';
  articoloParam: Articolo;

  constructor(private dao: ArticoloDaoService,
    private activatedRoute: ActivatedRoute) {
    const id: number = +this.activatedRoute.snapshot.paramMap.get('id');
    this.articoloParam = id == null ? null : this.dao.getArticoloById(id);
  }
}
```

## 7. HTTP

comp  
visualizza,  
2/3

```
ngOnInit() {  
  if (this.articoloParam != null) {  
    this.articolo = this.articoloParam;  
    this.articoloParam = null;  
    return;  
  }  
  this.dao.getArticoloHome(this);  
}  
visualizzaArticolo(articolo: Articolo) {  
  this.articolo = articolo;  
}  
}
```

## 7. HTTP

comp visualizza, 3/3

```
<ng-content></ng-content>
<div *ngIf="articolo && articolo.titolo">
  <p align="center" [ngStyle]="stileTitolo">{{articolo.titolo}}</p>
  <p align="center"></p>
  <div *ngFor="let sottotitolo of articolo.arraySottotitolo">
    <p align="right" *ngIf="sottotitolo.toUpperCase().includes('FONTE:');
      else sottotitoloUpper">{{sottotitolo}}</p>
    <ng-template #sottotitoloUpper>
      <p align="center" [ngClass]="classeSottotitolo">{{sottotitolo | uppercase}}</p>
    </ng-template>
  </div>
  <p align="right">Lunghezza articolo: {{articolo.testo.length / 7 | number: '0.0-0'}} parole</p>
  <p [innerHTML]="articolo.testo | pipeTestoBr"></p>
  <p align="right">Pubblicato il {{articolo.dataPubblicazione | date: 'dd/MM/yyyy HH:mm:ss'}}</p>
</div>
```

## 7. HTTP

comp cerca

```
import { Component, OnInit, ViewChild } from '@angular/core';
import { Articolo } from 'src/app/classes/articolo';
import { ArticoloDaoService } from './../../services/articolo-dao.service';
import { VisualizzaComponent } from '../visualizza/visualizza.component';
@Component({
  selector: 'app-cerca',
  templateUrl: './cerca.component.html',
  styleUrls: ['./cerca.component.css']
})
export class CercaComponent implements OnInit {
  articolo: Articolo;
  @ViewChild(VisualizzaComponent, { static: true }) visualizza:
  VisualizzaComponent;
  constructor(private dao: ArticoloDaoService) { }
  ngOnInit() { this.dao.getArticoloHome(this); }
  visualizzaArticolo(articolo: Articolo) {
    // this.articolo = articolo;
    this.visualizza.visualizzaArticolo(articolo);
  }
}
```

## 7. HTTP

comp riepilogo

```
import { Articolo } from './../../classes/articolo';
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { ArticoloDaoService } from './../../services/articolo-dao.service';
@Component({
  selector: 'app-riepilogo',
  templateUrl: './riepilogo.component.html',
  styleUrls: ['./riepilogo.component.css']
})
export class RiepilogoComponent implements OnInit {
  arrayArticoli: Array<Articolo> = [];
  constructor(private dao: ArticoloDaoService, private router: Router) { }
  ngOnInit() { this.dao.getArrayArticoli(this); }
  visualizzaArticolo(articolo: Articolo) {
    this.router.navigate(['visualizza', articolo.id]);
  }
  modificaArticolo(articolo: Articolo) {
    this.router.navigate(['modifica', articolo.id]);
  }
}
```

## 7. HTTP

comp edit,  
1/5

```
import { ArticoloDaoService } from '../services/articolo-dao.service';
import { Articolo } from 'src/app/classes/articolo';
import { Component, OnInit } from '@angular/core';
import { ActivatedRoute } from '@angular/router';

@Component({
  selector: 'app-edit',
  templateUrl: './edit.component.html',
  styleUrls: ['./edit.component.css']
})
export class EditComponent implements OnInit {
  inserimentoModifica: string = 'Inserimento/Modifica';
  inserisciSalvaModifiche: string = 'Inserisci/Salva modifiche';
  titoloModel: string;
  files: any[];
  urlImmagine: string;
  nomeImmagine: string;
  sottotitoloModel: string;
  arraySottotitolo: Array<string> = [];
  testoModel: string;
  id: number;
  articolo: Articolo;
```



## 7. HTTP

comp edit,  
2/5

```
constructor(private dao: ArticoloDaoService,  
private activatedRoute: ActivatedRoute) {  
  this.id = +this.activatedRoute.snapshot.paramMap.get('id');  
}  
ngOnInit() {  
  this.inserimentoModifica = this.id === 0 ? 'Inserimento' : 'Modifica';  
  this.inserisciSalvaModifiche = this.id === 0 ? 'Inserisci' : 'Salva modifiche';  
  if (this.id !== 0) {  
    this.articolo = this.dao.getArticoloById(this.id);  
    this.titoloModel = this.articolo.titolo;  
    this.urlImmagine = this.articolo.immagine;  
    this.nomeImmagine = this.articolo.nomeImmagine;  
    this.arraySottotitolo = Object.assign([], this.articolo.arraySottotitolo);  
    this.testoModel = this.articolo.testo;  
  }  
}
```

## 7. HTTP

comp edit,  
3/5

```
salva() {  
    if (!this.isArticoloValido()) { return; }  
    if (this.id === 0) { this.articolo = new Articolo(); }  
    this.articolo.titolo = this.titoloModel;  
    this.articolo.immagine = this.urlImmagine;  
    this.articolo.nomeImmagine = this.nomeImmagine;  
    this.articolo.arraySottotitolo = this.arraySottotitolo;  
    this.articolo.testo = this.testoModel;  
    if (this.id === 0) {  
        this.dao.postNewConsumer(this.articolo);  
        this.articolo = new Articolo();  
        this.titoloModel = null;  
        this.files = null;  
        this.urlImmagine = null;  
        this.nomeImmagine = null;  
        this.sottotitoloModel = null;  
        this.arraySottotitolo = [];  
        this.testoModel = null;  
        this.id = 0;  
    }  
}
```

## 7. HTTP

comp edit,  
4/5

```
aggiungiSottotitolo() {  
  if (this.sottotitoloModel == null || this.sottotitoloModel.length < 10) {  
    alert('Introdurre un sottotitolo di almeno 10 caratteri');  
    return;  
  }  
  this.arraySottotitolo.push(this.sottotitoloModel);  
  this.sottotitoloModel = null;  
}  
  
caricaImmagine(event) {  
  this.files = event.target.files;  
  const reader: FileReader = new FileReader();  
  if (this.files[0] == null) { return; }  
  reader.readAsDataURL(this.files[0]);  
  reader.onload = (event2: any) => {  
    this.urlImmagine = event2.target.result;  
    // this.urlImmagine = reader.result as string;  
    this.nomeImmagine = this.files[0].name;  
  };  
}
```

## 7. HTTP

comp edit,  
5/5

```
isArticoloValido(): boolean {  
    if (this.titoloModel == null || this.titoloModel.length < 4) {  
        alert('Introdurre un titolo di almeno 4 caratteri'); return false;  
    }  
    if (this.urlImmagine == null || this.urlImmagine.length === 0) {  
        alert('Scegliere un\'immagine'); return false;  
    }  
    if (this.nomeImmagine == null || this.nomeImmagine.length === 0) {  
        alert('Scegliere un\'immagine'); return false;  
    }  
    if (this.arraySottotitolo == null || this.arraySottotitolo.length < 2) {  
        alert('Introdurre almeno 2 sottotitoli'); return false;  
    }  
    if (this.testoModel == null || this.testoModel.length < 20) {  
        alert('Introdurre un testo di almeno 20 caratteri'); return false;  
    }  
    return true;  
}
```