Angular HttpClient for Sending Http Request Example

Step 1: Create New App

You can easily create your angular app using bellow command:

```
ng new my-new-app
```

Step 2: Import HttpClientModule

In this step, we need to import HttpClientModule to app.module.ts file. so let's import it as like bellow:

src/app/app.module.ts

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

import { NgModule } from '@angular/core';



import { AppComponent } from './app.component';

import { HttpClientModule } from '@angular/common/http';



@NgModule({

  declarations: [

    AppComponent

  ],
```

```
  imports: [
    BrowserModule,
    HttpClientModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

Step 3: Create Service for API

Here, we need to create service for http client request. we will create service file and write client http request code. this service will use in our component file. So let's create service and put bellow code:

```
ng g s services/post
```

Now let's add code as like bellow:

src/app/services/post.service.ts

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';


@Injectable({
  providedIn: 'root'
})
```

```
export class PostService {

  private url =
'http://jsonplaceholder.typicode.com/posts';



  constructor(private httpClient: HttpClient) { }



  getPosts(){

    return this.httpClient.get(this.url);

  }



  create(post){

    return this.httpClient.post(this.url,
JSON.stringify(post));

  }



}
```

Step 4: Use Service to Component

Now we have to use this services to our app component. So let's updated code as like bellow:

src/app/app.component.ts

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

import { PostService } from './services/post.service';
```

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

  constructor(private service:PostService) {}

  ngOnInit() {
      this.service.getPosts()
        .subscribe(response => {
          this.posts = response;
        });
  }

  createPost(input: HTMLInputElement){
    let post = {title: input.value};
    input.value = '';
```

```
    this.service.create(post)
      .subscribe((response: { id }) => {
        post['id'] = response.id;
        this.posts.splice(0,0, post);
      });
  }
}
```

Step 5: Updated View File

Now here, we will updated our html file. let's put bellow code:

src/app/app.component.html

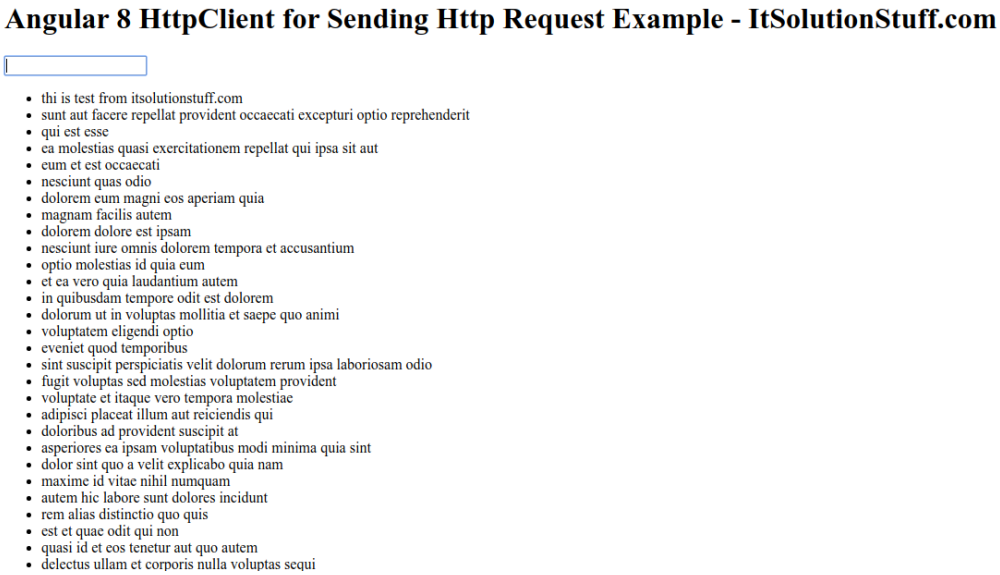
```
<h1>Angular 8 HttpClient for Sending Http Request Example
- ItSolutionStuff.com</h1>


<input
    (keyup.enter)="createPost(title)" #title
    type="text" class="form-control">


<ul class="list-group">
  <li
      *ngFor="let post of posts"
      class="list-group-item">
      {{ post.title }}
```

```
  </li>
</ul>
```

Now we are ready to run our example, you can run by following command:

```
ng serve
```

you will see layout as bellow:



Angular Http Post Request Example

Step 1: Create New App

You can easily create your angular app using bellow command:

```
ng new my-new-app
```

Step 2: Import HttpClientModule

In this step, we need to import HttpClientModule to app.module.ts file. so let's import it as like bellow:

src/app/app.module.ts

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

import { NgModule } from '@angular/core';



import { AppComponent } from './app.component';

import { HttpClientModule } from '@angular/common/http';



@NgModule({

  declarations: [

    AppComponent

  ],

  imports: [

    BrowserModule,

    HttpClientModule

  ],

  providers: [],

  bootstrap: [AppComponent]

})

export class AppModule { }
```

Step 3: Create Service for API

Here, we need to create service for http client request. we will create service file and write client http request code. this service will use in our component file. So let's create service and put bellow code:

```
ng g s services/post
```

Now let's add code as like bellow:

src/app/services/post.service.ts

```
import { Injectable } from '@angular/core';

import { HttpClient } from '@angular/common/http';



@Injectable({

  providedIn: 'root'

})

export class PostService {

  private url =
'http://jsonplaceholder.typicode.com/posts';



  constructor(private httpClient: HttpClient) { }



  getPosts(){

    return this.httpClient.get(this.url);

  }
```

```
  create(post){

    return this.httpClient.post(this.url,
JSON.stringify(post));

  }


}
```

Step 4: Use Service to Component

Now we have to use this services to our app component. So let's updated code as like bellow:

src/app/app.component.ts

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

import { PostService } from './services/post.service';


@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.css']

})

export class AppComponent implements OnInit {

  posts;
```

```
  constructor(private service:PostService) {}


  ngOnInit() {
      this.service.getPosts()
        .subscribe(response => {
          this.posts = response;
        });
  }


  createPost(input: HTMLInputElement){
    let post = {title: input.value};
    input.value = '';


    this.service.create(post)
      .subscribe((response: { id }) => {
        post['id'] = response.id;
        this.posts.splice(0,0, post);
      });
  }
}
```

Step 5: Updated View File

Now here, we will updated our html file. let's put bellow code:

src/app/app.component.html

```
<h1>Angular 8 HttpClient for Sending Http Request Example
- ItSolutionStuff.com</h1>



<input

    (keyup.enter)="createPost(title)" #title

    type="text" class="form-control">



<ul class="list-group">

  <li

      *ngFor="let post of posts"

      class="list-group-item">

      {{ post.title }}

  </li>

</ul>
```

Now we are ready to run our example, you can run by following command:
Read Also: Angular 9/8 HttpClient for Sending Http Request Example

```
ng serve
```

you will see layout as bellow:

Angular 8 HttpClient for Sending Http Request Example - ItSolutionStuff.com

- thi is test from itsolutionstuff.com
- sunt aut facere repellat provident occaecati excepturi optio reprehenderit
- qui est esse
- ea molestias quasi exercitationem repellat qui ipsa sit aut
- eum et est occaecati
- nesciunt quas odio
- dolorem eum magni eos aperiam quia
- magnam facilis autem
- dolorem dolore est ipsam
- nesciunt iure omnis dolorem tempora et accusantium
- optio molestias id quia eum
- et ea vero quia laudantium autem
- in quibusdam tempore odit est dolorem
- dolorum ut in voluptas mollitia et saepe quo animi
- voluptatem eligendi optio
- eveniet quod temporibus
- sint suscipit perspiciatis velit dolorum rerum ipsa laboriosam odio
- fugit voluptas sed molestias voluptatem provident
- voluptate et itaque vero tempora molestiae
- adipisci placeat illum aut reiciendis qui
- doloribus ad provident suscipit at
- asperiores ea ipsam voluptatibus modi minima quia sint
- dolor sint quo a velit explicabo quia nam
- maxime id vitae nihil numquam
- autem hic labore sunt dolores incidunt
- rem alias distinctio quo quis
- est et quae odit qui non
- quasi id et eos tenetur aut quo autem
- delectus ullam et corporis nulla voluptas sequi