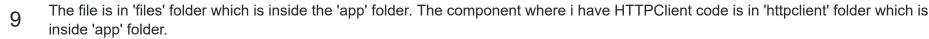
Angular 4: How to read content of text file with HTTPClient

Asked 1 year, 10 months ago Active 30 days ago Viewed 38k times



I have a .txt file in my Angular 4 project directory and I want to read its content. How to do it? Below is the code which I employed.





Meaning 'files' folder and 'httpclient' folder are children.



The code is shown below. It not working as i m getting 404 error - 'GET http://localhost:4200/files/1.txt 404 (Not Found)'

javascript





asked Nov 1 '17 at 10:59

yogihosting 2,316 1 16 36

Have you tried a relative URL? (i.e. http.get('files/1.txt') -- no initial slash) Or placing the files in the assets folder, the content of which - I

<u>angular.io/guide/http#requesting-non-json-data</u> – yogihosting Nov 1 '17 at 11:12

your file path is correct ? - Chandru Nov 1 '17 at 11:20 🎤

yes file path is correct. Even if i move the text file to the same folder of teh component and do this.http.get('./1.txt') i get the same error. - yogihosting Nov 1 '17 at 11:23

1 @yogihosting it's not working because when the CLI builds the project with ng serve it only processes .ts, .html and .css files. Static assets like .txt files do not belong in your code and they are dropped by ng serve . − AngularChef Nov 2 '17 at 10:32 ✓

5 Answers



Try like this:

19

```
this.http.get('app/files/1.txt').subscribe(data => {
    console.log(data.text());
})
```



The CLI can't access docments inside the app directory your project. if you move to text document you can access the text file like assets/1.txt.

if you want to access document inside the app directory you need to add path in assets array in the .angular-cli.json

.angular-cli.json

```
"assets": [
   "assets",
   "app", /* add this line to access document inside the app directory */
   "favicon.ico"
]
```

here below is my example try like this:

```
this.http.get('app/home/1.txt').subscribe(data => {
    console.log('data', data.text());
})
```



Yes assets files are accessible but How I can assess files kept in separate folder? - yogihosting Nov 1 '17 at 11:32

you need to add app in the angular-cli.json file inside the assets array . then you can access it. and import is re-run the project – Chandru Nov 1 '17 at 11:33 /

I added app to the angular-cli.json, its still not working. - yogihosting Nov 1 '17 at 11:38

@yogihosting After add app inside the assets array, you need to Re-run the project or Restart the project – Chandru Nov 1 '17 at 11:39

you are just great! I have to restart my angular project. I was using vs code. So i closed it and reopened it and ran the app once more. Now I am able to asses it. The code here is this.http.get('app/files/1.txt', {responseType: 'text'}) – yogihosting Nov 1 '17 at 11:46



Angular 6/7

10

{ responseType: 'text' as 'json'}



for now works

```
this.http.get("app/files/1.txt", { responseType: 'text' as 'json'}).subscribe(data => {
    console.log(data.text());
})
```

Refer to this ticket on GitHub for the complete discussion.





Hypenate

answered Nov 3 '18 at 13:22



645 1 13 24



Just one correction to the previous answer: add "responseType: 'text'" to the options:

3 this.http.get("app/files/1.txt", "{ responseType: 'text' }").subscribe(data => {

answered Apr 16 '18 at 10:11





I was able to get a local file(JSON) with Angular 6 project.

1

Project strucuture:



```
|-src
|--app
|---core
|---...
|---app.module.ts
|--assets
|...
```

Steps:

1) Inside angular.json I've added the folder where I put the file:

```
"projects": {
   "my-project": {
    "root": "",
    "sourceRoot": "src", // ** my project root is "src" folder
    "projectType": "application",
    "architect": {
       "build": {
         "builder": "@angular-devkit/build-angular:browser",
         "options": {
          // ...
           "assets": [
            "src",
                         // <--- ADDED THIS
            "src/assets",
            "src/favicon.ico",
            "src/manifest.json",
            "src/robots.txt",
             "src/web.config"
          1
```

```
import { HttpClient } from '@angular/common/http'
import { Injectable } from '@angular/core'
@Injectable({
  providedIn: 'root'
})
export class EnvironmentService {
  constructor(private readonly httpClient: HttpClient) {
  public init(): Promise<any> {
      return new Promise(resolve =>
         this. httpClient
         .get('file.json')
         .first()
         .subscribe((data: any) => {
           // data from JSON
            resolve()
         })
  }
```

Bonus:

In case if you want to read custom environment variables(besides what standard Angular one provides) you can add above service into App.module.ts:

```
export function init_app(appLoadService: EnvironmentService): () => Promise<any> {
    return () => appLoadService.init()
}

@NgModule({
    // ...
    providers: [
        EnvironmentService,
        { provide: APP_INITIALIZER, useFactory: init_app, deps: [ EnvironmentService ],
    multi: true },
    ],
    bootstrap: [ AppComponent ]
})
export class AppModule {
}
```

answered Jun 20 '18 at 5:37



Working with a Angular 6 app as well, followed your solution, but wasn't able to retrieve the JSON successfully. I created: <u>Angular 6: HttpClient Get JSON File 404 Error</u> To clarify my situation. – TYMG Jun 21 '18 at 17:07

Shared an answer in your post: stackoverflow.com/a/50986804/806202 I hope it will help. - Arsen Khachaturyan Jun 22 '18 at 11:20



This is tested in Angular 6

1 Create a service using



```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
//Read text file
export interface ResultJson{

@Injectable({
    providedIn: 'root'
})
export class TestService {
    urlEncoded = '/Test.text';
    getText(){
        return this.http.get(this.urlEncoded, {responseType: 'text'});
    }
}
```

and call the service in your component like below

```
resultJSON: ResultJson;
ResultJsonString : any;
this
   .testService
   .getText()
   .subscribe((data:ResultJson) => {
        this.ResultJsonString = data;
}
```

edited Aug 29 '18 at 20:53



answered Aug 29 '18 at 20:31

