

Meet The Overflow, a newsletter by developers, for developers. Fascinating questions, illuminating answers, and entertaining links from around the web. [Learn more](#)

How to make a string constant in angular 4?

Asked 2 years, 4 months ago Active 6 months ago Viewed 39k times

▲ In my service, I am using a `http` post. I want to set the URL as a constant.

14 `return this.http.get(this.config.API_URL+'users', options).map(res=>res.json());`

▼ I tried with a service:



4

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

@Injectable()
export class AppService {
  API_URL :String;

  constructor() {
    this.API_URL = 'some url';
  }
}
```

Is there any other method to make a constant value in Angular4 ?



edited Nov 16 '17 at 6:59



[Sangwin Gawande](#)
4,349 7 29 55

asked May 11 '17 at 13:11



[aish](#)
433 1 5 11

2 Possible duplicate of [Define global constants in Angular 2](#) – [elvin](#) May 11 '17 at 13:46

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

▲ I'm not sure if i understand your question but if you want to create constants, you can do it in a different class and import it.

37

▼

```
export class Constants {  
  public static get HOME_URL(): string { return "sample/url/"; }  
}
```



sample.component.ts

```
import { Constants } from "../constants";  
@Component({  
  
  })  
  export class SampleComponent {  
    constructor() {  
      let url = Constants.HOME_URL;  
    }  
  }  
}
```

answered May 11 '17 at 13:42



Cory

725 8 16

how to call like this? `const USER_AUTH_API_URL = '/api-url';` – [Velusamy Venkatraman](#) Jul 14 at 13:52 ✎



You can simply export a constant using es6/typescript modules if that's all you need:

33

constants.ts:

▼

```
export const API_URL: string = 'api/url';
```

And import where needed:

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

...

```
return this.http.get(API_URL+'users', options)
    .map(res=>res.json());
```

or if you have many constants you can import them all:

```
import * as constants from './constants.ts';

...

return this.http.get(constants.API_URL+'users', options)
    .map(res=>res.json());
```

If you want to make your constants configurable per application on startup, you can use providers. Check out the top answer in this link: [how do I get angular2 dependency injection to work with value providers](#)

edited May 23 '17 at 12:26



Community ♦

1 1

answered May 11 '17 at 14:01



Eeks33

1,635 6 15

1 If the constant happens to be an object, you can wrap it in a `Object.freeze({ ... })` to make it a real constant. — Jeffrey Roosendaal Mar 27 '18 at 14:32

Define app constant as public static readonly as below

3

```
public static readonly constName = 'Consts Value'
```

/app-constants.ts

```
export class Constants {

    public static readonly routeAuthRegister = '/auth/register';
    public static readonly routeAuthLogin = '/auth/login';
    public static readonly routeAuthRecovery = '/auth/forgot-password';

}
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

It's better to keep your base URI's in environment files. and define your environments in apps in .angular-cli.json. i here attache screen shot below for define custom app environments .

```
import { environment } from '../environments/environment';

export class ApiURIs {

    public static readonly apiURL: string = environment.apiEndpoint;
    public static readonly apiV1: string = environment.apiEndpoint + '/v1';
    public static readonly apiV2: string = environment.apiEndpoint + '/v2';


    public static readonly login: string = ApiEndpoints.apiV1 + '/login';
    public static readonly register: string = ApiEndpoints.apiV1 + '/register';
    public static readonly signup: string = ApiEndpoints.apiV2 + '/register';

}
```

Usage of constants

/core/auth.service.ts

```
import { ApiURIs } from './api-constants';

@Injectable()
export class AuthService {
    .....

    signUpUser(data) {
        return this.https.post(`${ApiURIs.signup}`, data);
    }

    .....
}
```

/auth/login.component.ts

```
export class LoginComponent implements OnInit {
    .....
```

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

```

}
.....
}

```

Define your different API URI's for different environment

The screenshot shows the VS Code interface with the Explorer on the left and two TypeScript files open in the editor.

EXPLORER:

- OPEN EDITORS: 1 UNSAVED
- HEYU
 - aws
 - dev
 - TS environment.prod.ts
 - TS environment.ts
 - qa1
 - sandbox
 - test
 - TS environment.prod.ts
 - TS environment.ts
 - icons
 - ★ favicon.ico
 - <> index.html
 - TS main.ts
 - () manifest.json
 - TS polyfills.ts
 - 🔗 styles.scss
 - TS test.ts
 - () tsconfig.app.json
 - () tsconfig.spec.json
 - TS typings.d.ts
 - () .angular-cli.json
 - () .angulardoc.json
 - ⚙ editorconfig
 - () google-services.json

TS environment.ts (Left Panel):

```

6 export const environment = {
7   production: false,
8   aws_deploy: true,
9   basic_auth_token: 'open_to',
10  apiEndpoint: '...',
11  // apiEndpoint: '...',
12  // apiEndpoint: '...',
13  secret: 'upay local key',
14  xApiKey: '...',
15  web: '...',
16  ios: '...',
17  android: '...',
18 },
19 appVersions: {
20   web: '12.3.0',
21   ios: '12.3.0',
22   android: '12.3.0'
23 },
24 firebase: {
25   apiKey: 'AIzaSyBzMgn4X1K...',
26   authDomain: 'upay-dev-287...',
27   databaseURL: 'https://up...',
28   projectId: 'upay-dev-287...',
29   storageBucket: 'upay-dev...',
30   messagingSenderId: '36651...'
31 },
32 smsAlias: 'UPAYTEST',
33 googleMapsKey: 'AIzaSyBT5l...',
34 };

```

TS environment.ts (Right Panel):

```

5 },
6 "apps": [
7   { ...
37   },
38   {
39     "root": "src",
40     "outDir": "dist",
41     "assets": [ ...
46   ],
47   "serviceWorker": true,
48   "index": "index.html",
49   "main": "main.ts",
50   "polyfills": "polyfills.ts",
51   "test": "test.ts",
52   "tsconfig": "tsconfig.app.json",
53   "testTsconfig": "tsconfig.spec.json",
54   "prefix": "heyu",
55   "styles": [ ...
59   ],
60   "scripts": [ ...
62   ],
63   "environmentSource": "environments/environment.ts",
64   "environments": {
65     "dev": "environments/dev/environment.ts",
66     "prod": "environments/dev/environment.prod.ts"
67   }
68 },
69 {
70   "root": "src",
71   "outDir": "dist",

```

edited Mar 28 at 17:04

answered Mar 28 at 16:05



Chanuka Asanka

1,468 1 17 29

By using our site, you acknowledge that you have read and understand our Cookie Policy, Privacy Policy, and our Terms of Service.

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).