How to store token in Local or Session Storage in Angular 2?

51

23

I want to use Local or session storage to save authentication token in `angular 2.0.0`. I use `angular2-localstorage` but it works only **angular 2.0.0-rc.5** and when I used it in 2.0.0 it through me **Type** error. I want to use default local storage of Angular 2.0.0.

angular   local-storage

edited Oct 3 '16 at 22:39   asked Oct 3 '16 at 21:09
halfer   Rahul dev
15.1k   7   60   122   **547**   2   9   21

Check out Angular 2 Cool Storage – Dave V Oct 3 '16 at 21:12

1   Is it best way to store in local storage? – Rahul dev   Oct 3 '16 at 21:18

9 Answers

88

Save to local storage

```
localStorage.setItem('currentUser', JSON.stringify({ token: token, name: name }));
```

Load from local storage

```
var currentUser = JSON.parse(localStorage.getItem('currentUser'));
var token = currentUser.token; // your token
```

For more I suggest you go through this tutorial: Angular 2 JWT Authentication Example & Tutorial

If I type in `JSON.parse(localStorage.getItem('currentUser')).token` I can see the token in console. How can I avoid this in production ? – Zaker Mar 20 '18 at 13:36 ✏

---

5   @Zaker Don't store your token in local storage. – Archibald May 9 '18 at 16:00

@Archibald Can you please suggest me where to store the token. Any link describing will also be helpful. Thanks – Zaker May 10 '18 at 5:00

Use cookies for that, with the `HttpOnly` and `Secure` flags: stormpath.com/blog/... – ilopezluna Aug 16 '18 at 10:46

@Zaker If the browser ever has the token, then it is never safe from the user. The only thing you can do is never send anything to the browser you wouldn't want the user to see. That's different than preventing a XSS. – Andrew T Finnell Dec 14 '18 at 20:43

▲

33

▼

That totally depends of what do you need exactly. If you just need to store and retrieve a token in order to use it in your http requests, i suggest you to simply create a service and use it in your whole project.

example of basic integration:

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

@Injectable()
export class TokenManager {

    private tokenKey:string = 'app_token';

    private store(content:Object) {
        localStorage.setItem(this.tokenKey, JSON.stringify(content));
    }

    private retrieve() {
        let storedToken:string = localStorage.getItem(this.tokenKey);
        if(!storedToken) throw 'no token found';
        return storedToken;
    }

    public generateNewToken() {
        let token:string = '...';//custom token generation;
        let currentTime:number = (new Date()).getTime() + ttl;
        this.store({ttl: currentTime, token});
```

```
            let storedToken = JSON.parse(this.retrieve());
            if(storedToken.ttl < currentTime) throw 'invalid token found';
            token = storedToken.token;
        }
        catch(err) {
            console.error(err);
        }
        return token;

    }

}
```

However if you need to use the local storage more often, by using the stored values in your app views for example. You can use one of the libraries that provides a wrapper of the webstorages like you did with angular2-localstorage.

I created some months ago ng2-webstorage that should respond to your needs. It provides two ng2 services and two decorators to sync the webstorage's values and the service/component's attributes.

```
import {Component} from '@angular/core';
import {LocalStorageService, LocalStorage} from 'ng2-webstorage';

@Component({
    selector: 'foo',
    template: `
        <section>{{boundValue}}</section>
        <section><input type="text" [(ngModel)]="attribute"/></section>
        <section><button (click)="saveValue()">Save</button></section>
    `,
})
export class FooComponent {

    @LocalStorage()
    boundValue; // attribute bound to the localStorage
    value;

    constructor(private storage:LocalStorageService) {
        this.localSt.observe('boundValue')// triggers the callback each time a new value
is set
            .subscribe((newValue) => console.log('new value', newValue));
```

```
    }
```

we can store session storage like that

21 store token should be like

```
    localStorage.setItem('user', JSON.stringify({ token: token, username: username }));
```

Store Session in to sessionStorage

You can store both string and array into session storage

String Ex.

```
    let key = 'title';
    let value = 'session';
    sessionStorage.setItem(key, value);
```

Array Ex.

```
    let key = 'user';
    let value = [{'name':'shail','email':'example@gmail.com'}];

    value = JSON.stringify(value);

    sessionStorage.setItem(key, value);
```

Delete saved session from sessionStorage by key

```
sessionStorage.removeItem('key');
```

Delete all saved sessions from sessionStorage

```
sessionStorage.clear();
```

2. store Local storage should be like

Store items in to localStorage

You can store both string and array into location storage

String Ex.

```
let key = 'title';
let value = 'session';
localStorage.setItem(key, value);
```

Array Ex.

```
let key = 'user';
let value = [{'name':'shail','email':'example@gmail.com'}];

value = JSON.stringify(value);

localStorage.setItem(key, value);
```

Get stored items from localStorage by key

```
const item = localStorage.getItem('key');
```

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up　　　OR SIGN IN WITH　　　G Google　　　Facebook ✕

```
localStorage.clear();
```

edited Jun 27 '18 at 14:42   answered Sep 22 '17 at 10:36

    kyle   Shailesh Ladumor

    698   5   13   **3,642** 22 39

---

1   by both methods, we can access token by `this.sessionStorage.getItem('token')` , remove token by `this.sessionStorage.removeItem('token')` . – Vikas Rana Nov 22 '18 at 11:57 ✏

3

As a general rule, the token should not be stored on the `localStorage` neither the `sessionStorage` . Both places are accessible from JS and the JS should not care about the authentication token.

IMHO The token should be stored on a cookie with the `HttpOnly` and `Secure` flag as suggested here: https://stormpath.com/blog/where-to-store-your-jwts-cookies-vs-html5-web-storage

answered Aug 16 '18 at 10:52

    ilopezluna

    2,403 5 27 55

---

2   I agree with you, local storage is not secure enough. – Mohammad Alghanem Apr 3 at 9:59

0

Here is the best practice: https://github.com/PillowPillow/ng2-webstorage

I was use it in AngularJs, now with Angular2. Wery usefull.

edited Sep 28 '17 at 10:24   answered Sep 28 '17 at 9:49

    Anton Pegov

    504 7 20

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up   OR SIGN IN WITH   G Google   Facebook✕

0

```
@NgModule({
    providers: [
        { provide: Storage, useValue: localStorage }
    ],
    imports:[],
    declarations:[]
})
```

And then you can use DI to get it wherever you need it.

```
@Injectable({
    providedIn:'root'
})
export class StateService {
    constructor(private storage: Storage) { }
}
```

answered May 30 '18 at 17:22

[dwbartz](#)
838   10   13

0

```
import { Injectable,OpaqueToken } from '@angular/core';
export const localStorage = new OpaqueToken('localStorage');
```

Place these lines in the top of the file, it should resolve the issue.

answered Jun 16 '18 at 4:38

[Deependra Kushwah](#)
28   2   9

```
var arr=[{"username":"sai","email":"sai@example.com,"}]
```

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up   OR SIGN IN WITH   **G** Google   Facebook ✕

2   Thank you for adding an answer to this question. Could you edit to add some explanation why your answer is a good one and help learners understand it. This will help you get reputation votes. – Brian Tompsett - 汤莱恩 Aug 16 '18 at 10:44

Simple example:

0

```
var userID = data.id;

localStorage.setItem('userID',JSON.stringify(userID));
```

edited Sep 5 '18 at 20:09   answered Jun 27 '18 at 13:29

Costamilam   Navin Kumar
119   6   **9**   1