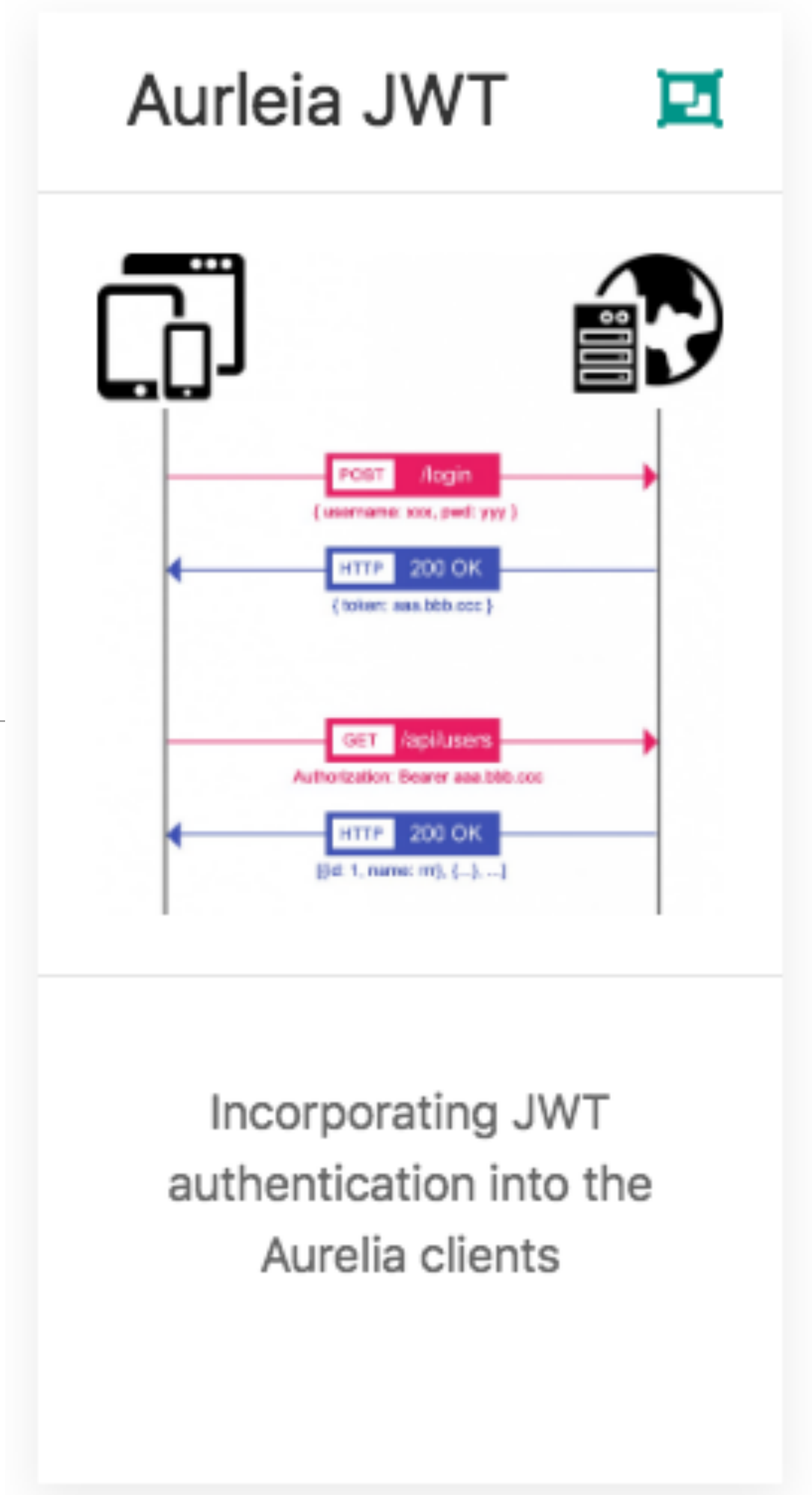
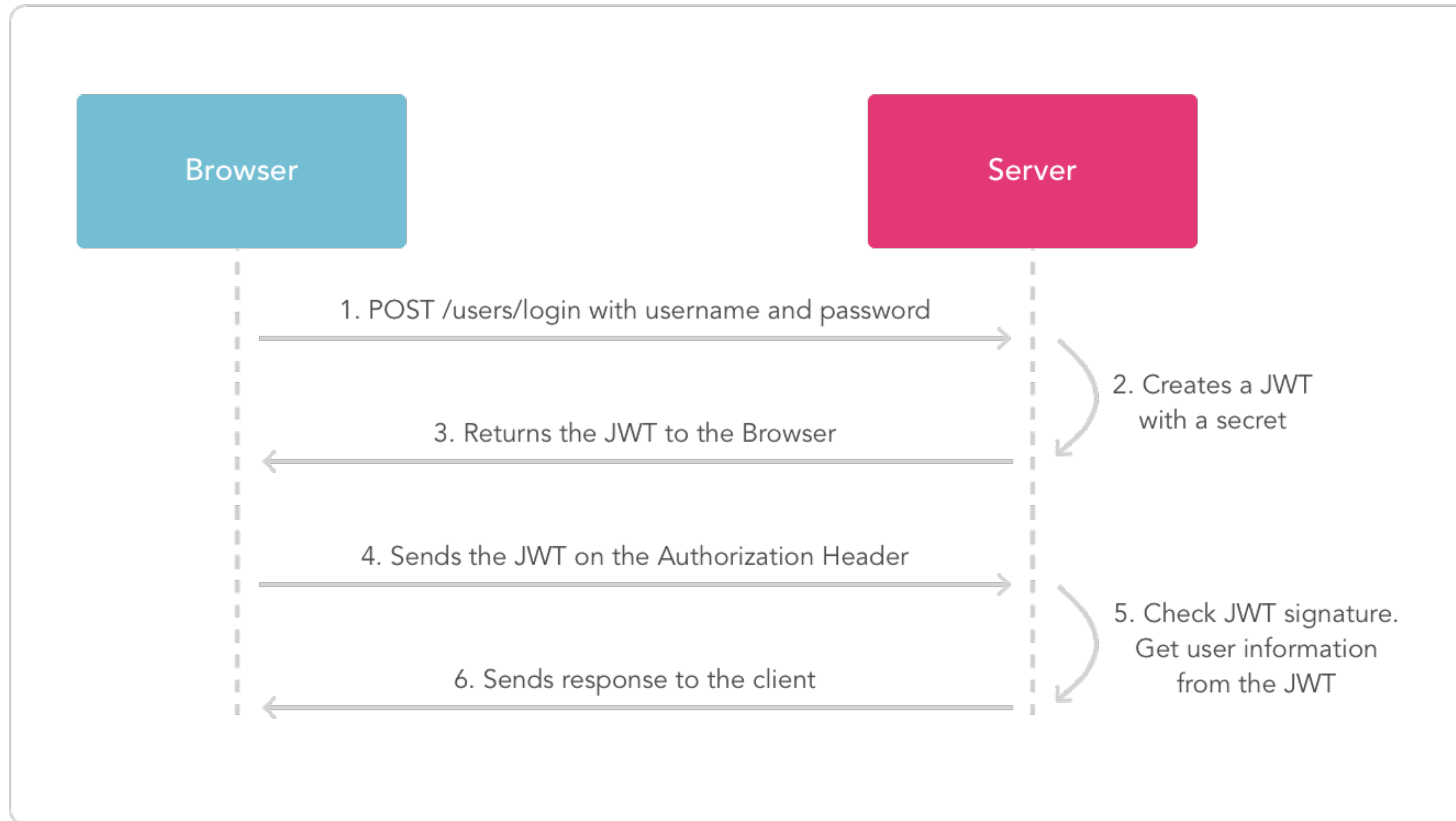


Jwt in Aurelia



JWT Authentication



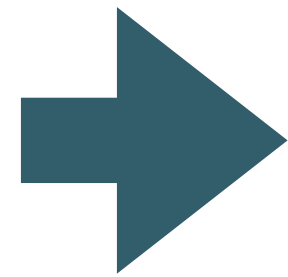
Review DonationService

Currently retrieves all
Candidates, Users &
Donations

```
constructor(  
  private httpClient: HttpClient,  
  private ea: EventAggregator,  
  private au: Aurelia,  
  private router: Router  
) {  
  httpClient.configure(http => {  
    http.withBaseUrl('http://localhost:3000');  
  });  
  this.getCandidates();  
  this.getUsers();  
  this.getDonations();  
}
```

Revised Constructor

```
constructor(  
  private httpClient: HttpClient,  
  private ea: EventAggregator,  
  private au: Aurelia,  
  private router: Router  
) {  
  httpClient.configure(http => {  
    http.withBaseUrl('http://localhost:3000');  
  });  
  this.getCandidates();  
  this.getUsers();  
  this.getDonations();  
}
```



```
constructor(  
  private httpClient: HttpClient,  
  private ea: EventAggregator,  
  private au: Aurelia,  
  private router: Router  
) {  
  httpClient.configure(http => {  
    http.withBaseUrl('http://localhost:3000');  
  });  
}
```

Simplified, no longer retrieve
any information

Revised Login Method

- Post method - should now generate jwt in response
- Set token as header in all subsequent requests
- Once authenticated we only retrieve candidates

```
async login(email: string, password: string) {
  const response = await this.httpClient.post('/api/users/authenticate', {
    email: email,
    password: password
  });
  const status = await response.content;
  if (status.success) {
    this.httpClient.configure(configuration => {
      configuration.withHeader('Authorization', 'bearer ' + status.token);
    });
    await this.getCandidates();
    this.changeRouter(PLATFORM.moduleName('app'));
    return true;
  } else {
    return false;
  }
}
```


Revised Logout Method

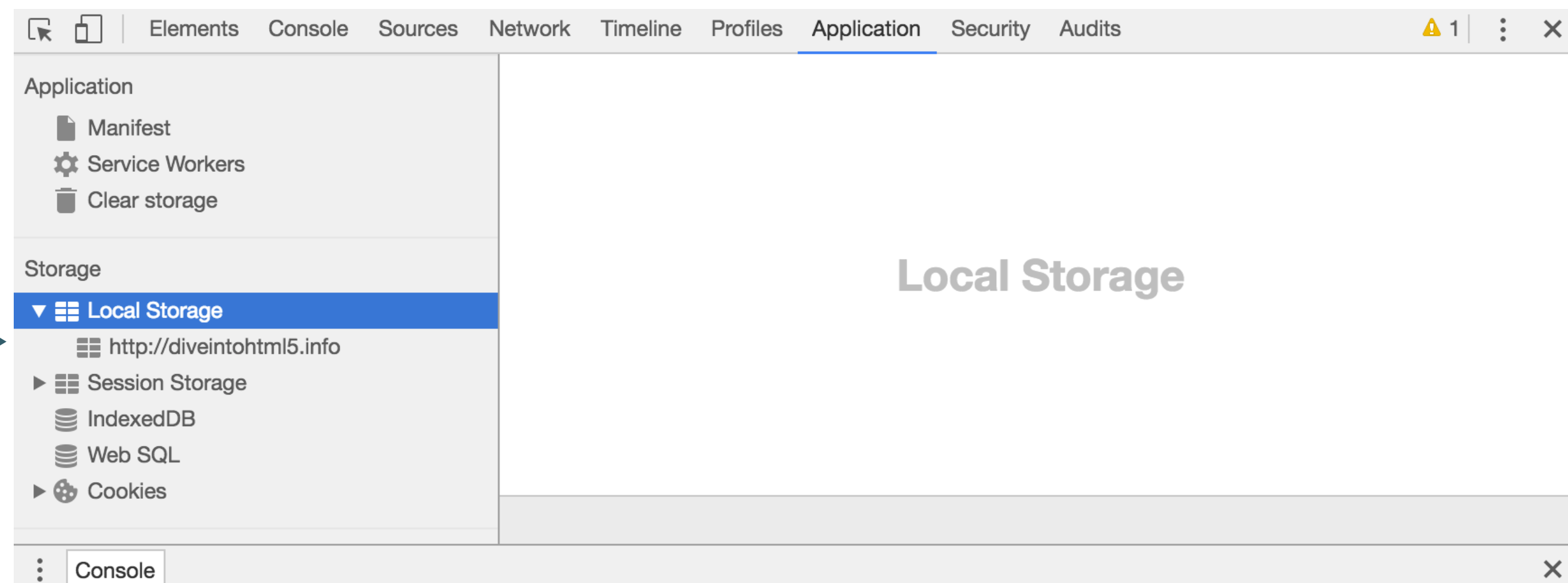
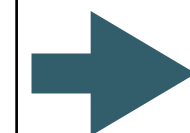
- Clear the Authorization header

```
logout() {  
  this.httpClient.configure(configuration => {  
    configuration.withHeader('Authorization', '');  
  });  
  this.changeRouter(PLATFORM.moduleName('start'));  
}
```

- A html5 standard way for web pages to store named key/value pairs locally, within the client web browser.
- Like cookies, this data persists even after you navigate away from the web site, close your browser tab, exit your browser.
- Unlike cookies, this data is never transmitted to the remote web server (unless you send it manually)

Local Storage

Inspect it using
developer tools



Storing Tokens in Local Storage

- donation name value pair created in local storage



```
async login(email: string, password: string) {  
  ...  
  if (status.success) {  
    this.httpClient.configure(configuration => {  
      configuration.withHeader('Authorization', 'bearer ' + status.token);  
    });  
    localStorage.donation = JSON.stringify(response.content);  
    await this.getCandidates();  
  }  
  ...  
}
```

	Key	Value
Application	donation	{"success":true,"token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjU4Mzk0YmY1YzJiZTAwYWI5Mm1xZmM2MSIsImVtYWlsIjoibW"}
Manifest		
Service Workers		
Clear storage		
Storage		
Local Storage		
http://localhost:9000		
Session Storage		
IndexedDB		
Web SQL		
Cookies		

Check LocalStorage for Tokens

- If token is found:
- set the token as an 'Authorization' header for subsequent api requests

```
checkIsAuthenticated() {  
  let authenticated = false;  
  if (localStorage.donation !== 'null') {  
    authenticated = true;  
    this.httpClient.configure(http => {  
      const auth = JSON.parse(localStorage.donation);  
      http.withHeader('Authorization', 'bearer ' + auth.token);  
    });  
    this.changeRouter(PLATFORM.moduleName('app'));  
  }  
}
```

On App Startup...

- Check to see if token is present...
- ... if it is, bypass login/signup routes and go straight to 'home' router

```
@inject(DonationService)
export class Start {
  router: Router;
  constructor(private ds: DonationService) {}

  configureRouter(config: RouterConfiguration, router: Router) {
    config.map([
      {
        route: ['', 'login'],
        name: 'Login',
        moduleId: PLATFORM.moduleName('views/login'),
        nav: true,
        title: 'Login'
      },
      {
        route: 'signup',
        name: 'signup',
        moduleId: PLATFORM.moduleName('views/signup'),
        nav: true,
        title: 'Signup'
      }
    ]);
    this.router = router;
  }

  attached() {
    this.ds.checkIsAuthenticated();
  }
}
```