

# FactorAuthenticationService

August 17, 2026

## FactorAuthenticationService

**Kind:** Class

**Source:** Frontend/src/app/services/security/factor-authentication.service.ts (line 58)

**Part of:** Frontend

FactorAuthenticationService gathers device and location context for factor-authentication flows. It creates a DeviceInfo fingerprint, reads the user's available location data, and exposes IP-based location lookup as an Observable<LocationInfo> .

### Methods

| Method                    | Signature                   | Returns                  |
|---------------------------|-----------------------------|--------------------------|
| generateDeviceFingerprint | generateDeviceFingerprint() | DeviceInfo               |
| getUserLocation           | getUserLocation()           | any                      |
| getLocationFromIP         | getLocationFromIP()         | Observable<LocationInfo> |

### Where it refuses work

- FactorAuthenticationService stops the work with an early return when !navigator.geolocation .
- FactorAuthenticationService stops the work with an early return when /iPad|Tablet/i.test(userAgent) .

### Diagram

```
graph LR
  Client[Application Component] --> Service[FactorAuthenticationService]
  Service --> Fingerprint[Fingerprint[generateDeviceFingerprint]]
  Service --> BrowserLocation[BrowserLocation[getUserLocation]]
  Service --> IPLocation[IPLocation[getLocationFromIP]]
```

```
Fingerprint --> DeviceInfo[DeviceInfo]
BrowserLocation --> LocationData[Location Data]
IPLocation --> LocationInfo[Observable LocationInfo]
```

## Usage

```
import { Component, inject, OnInit } from '@angular/core';
import { FactorAuthenticationService } from
  '../services/security/factor-authentication.service';

@Component({
  selector: 'app-factor-authentication',
  template: `<!-- Factor authentication UI -->`,
})
export class FactorAuthenticationComponent implements OnInit {
  private readonly factorAuthentication =
    inject(FactorAuthenticationService);

  ngOnInit(): void {
    const deviceInfo = this.factorAuthentication.generateDeviceFingerprint();
    const browserLocation = this.factorAuthentication.getUserLocation();

    this.factorAuthentication.getLocationFromIP().subscribe({
      next: (ipLocation) => {
        console.log({ deviceInfo, browserLocation, ipLocation });
      },
      error: (error) => {
        console.error('Unable to resolve location from IP.', error);
      },
    });
  }
}
```

## AI Coding Instructions

- Call `generateDeviceFingerprint()` when authentication context needs current device metadata.
- Treat `getUserLocation()` as optional or unavailable, since browser location access may require user permission.
- Subscribe to `getLocationFromIP()` and handle both successful and failed `Observable` responses.
- Keep location and fingerprint data within the security flow; do not log sensitive values in production.

## Used by

2 references from 2 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

## Imported by (2)

- `LoginComponent` — `Frontend/src/app/components/security/login/login.component.ts :1`
- `EnhanceLoginResponse` — `Frontend/src/app/services/security/auth.service.ts :1`