To pass the color from a configuration file, you can create a service that loads the configuration data and exposes it to the rest of your application. Here's an example of how you could modify the previous solution to use a configuration file:

1. Create a configuration file, `config.json`, that contains the color you want to use:

```json
{
  "headerColor": "#ff0000"
}
```

2. Create a service that loads the configuration data:

```typescript
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';

@Injectable({
  providedIn: 'root'
})
export class ConfigService {
  private configData: any;

  constructor(private http: HttpClient) { }

  public loadConfig(): Promise<any> {
    return this.http.get('/assets/config.json')
      .toPromise()
      .then((data) => {
        this.configData = data;
      });
  }

  public get headerColor(): string {
    return this.configData.headerColor;
  }
}
```

This service uses Angular's `HttpClient` to load the configuration file and exposes the `headerColor` property, which returns the color defined in the file.

3. Modify your component to use the `ConfigService` to get the color:

```typescript
import { Component, OnInit } from '@angular/core';
import { ConfigService } from './config.service';
```

```
@Component({
  selector: 'app-my-component',
  template: `
    <header [ngStyle]="{'background-color': headerColor}">
      <!-- Header content goes here -->
    </header>
  `,
})
export class MyComponent implements OnInit {
  headerColor: string;

  constructor(private configService: ConfigService) { }

  ngOnInit(): void {
    this.headerColor = this.configService.headerColor;
  }
}
```

This code injects the `ConfigService` into the component and uses it to get the `headerColor` property, which is then used in the `ngStyle` directive to set the background color of the header.

4. Add the `ConfigService` to your application's providers:

```typescript
import { HttpClientModule } from '@angular/common/http';
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { ConfigService } from './config.service';
import { AppComponent } from './app.component';

@NgModule({
  declarations: [
    AppComponent,
  ],
  imports: [
    BrowserModule,
    HttpClientModule,
  ],
  providers: [
    ConfigService,
  ],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

This code adds the `ConfigService` to the providers list in your application's module, so that it can be injected into components.

When the `MyComponent` is initialized, it uses the `ConfigService` to get the `headerColor` property, which is then used in the `ngStyle` directive to set the background color of the header. The `headerColor` property is dynamically updated whenever the configuration file changes, so the header color will automatically update without requiring a page reload.