API Report File for "@angular/common_http"

> *Do not edit this file. It is a report generated by [API Extractor](https://api-extractor.com/).*

```ts
import * as i0 from '@angular/core';
import { InjectionToken } from '@angular/core';
import { ModuleWithProviders } from '@angular/core';
import { Observable } from 'rxjs';
import { XhrFactory as XhrFactory_2 } from '@angular/common';

// @public
export const HTTP_INTERCEPTORS: InjectionToken<HttpInterceptor[]>;

// @public
export abstract class HttpBackend implements HttpHandler {
    // (undocumented)
    abstract handle(req: HttpRequest<any>): Observable<HttpEvent<any>>;
}

// @public
export class HttpClient {
    constructor(handler: HttpHandler);
    delete(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string | number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'arraybuffer';
        withCredentials?: boolean;
        body?: any | null;
    }): Observable<ArrayBuffer>;
    delete(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string | number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'blob';
        withCredentials?: boolean;
        body?: any | null;
    }): Observable<Blob>;
    delete(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'text';
        withCredentials?: boolean;
        body?: any | null;
    }): Observable<string>;
    delete(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'arraybuffer';
        withCredentials?: boolean;
        body?: any | null;
    }): Observable<HttpEvent<ArrayBuffer>>;
    delete(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'blob';
        withCredentials?: boolean;
        body?: any | null;
    }): Observable<HttpEvent<Blob>>;
    delete(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'text';
        withCredentials?: boolean;
        body?: any | null;
    }): Observable<HttpEvent<string>>;
    delete(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
        body?: any | null;
    }): Observable<HttpEvent<Object>>;
    delete<T>(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | (string | number | boolean)
[];
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
        body?: any | null;
    }): Observable<HttpEvent<T>>;
    delete(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'arraybuffer';
        withCredentials?: boolean;
        body?: any | null;
    }): Observable<HttpResponse<ArrayBuffer>>;
    delete(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'blob';
        withCredentials?: boolean;
        body?: any | null;
    }): Observable<HttpResponse<Blob>>;
    delete(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'text';
        withCredentials?: boolean;
        body?: any | null;
    }): Observable<HttpResponse<string>>;
    delete(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
        body?: any | null;
    }): Observable<HttpResponse<Object>>;
    delete<T>(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
        body?: any | null;
    }): Observable<HttpResponse<T>>;
    delete(url: string, options?: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
        body?: any | null;
    }): Observable<Object>;
    delete<T>(url: string, options?: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
        body?: any | null;
    }): Observable<T>;
    get(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'arraybuffer';
        withCredentials?: boolean;
    }): Observable<ArrayBuffer>;
    get(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'blob';
        withCredentials?: boolean;
    }): Observable<Blob>;
    get(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'text';
        withCredentials?: boolean;
    }): Observable<string>;
    get(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'arraybuffer';
        withCredentials?: boolean;
    }): Observable<HttpEvent<ArrayBuffer>>;
    get(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string | number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'blob';
        withCredentials?: boolean;
    }): Observable<HttpEvent<Blob>>;
    get(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string | number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'text';
        withCredentials?: boolean;
    }): Observable<HttpEvent<string>>;
    get(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string | number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<HttpEvent<Object>>;
    get<T>(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string | number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<HttpEvent<T>>;
    get(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'arraybuffer';
        withCredentials?: boolean;
    }): Observable<HttpResponse<ArrayBuffer>>;
    get(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'blob';
        withCredentials?: boolean;
    }): Observable<HttpResponse<Blob>>;
    get(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'text';
        withCredentials?: boolean;
    }): Observable<HttpResponse<string>>;
    get(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<HttpResponse<Object>>;
    get<T>(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<HttpResponse<T>>;
    get(url: string, options?: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<Object>;
    get<T>(url: string, options?: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<T>;
    head(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
```

```
        };
        reportProgress?: boolean;
        responseType: 'arraybuffer';
        withCredentials?: boolean;
    }): Observable<ArrayBuffer>;
    head(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'blob';
        withCredentials?: boolean;
    }): Observable<Blob>;
    head(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'text';
        withCredentials?: boolean;
    }): Observable<string>;
    head(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'arraybuffer';
        withCredentials?: boolean;
    }): Observable<HttpEvent<ArrayBuffer>>;
    head(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'blob';
        withCredentials?: boolean;
    }): Observable<HttpEvent<Blob>>;
    head(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'text';
        withCredentials?: boolean;
    }): Observable<HttpEvent<string>>;
    head(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<HttpEvent<Object>>;
    head<T>(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<HttpEvent<T>>;
    head(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string | number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'arraybuffer';
        withCredentials?: boolean;
    }): Observable<HttpResponse<ArrayBuffer>>;
    head(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string | number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'blob';
        withCredentials?: boolean;
    }): Observable<HttpResponse<Blob>>;
    head(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string | number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'text';
        withCredentials?: boolean;
    }): Observable<HttpResponse<string>>;
    head(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string | number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<HttpResponse<Object>>;
    head<T>(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<HttpResponse<T>>;
    head(url: string, options?: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<Object>;
    head<T>(url: string, options?: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<T>;
    jsonp(url: string, callbackParam: string): Observable<Object>;
    jsonp<T>(url: string, callbackParam: string): Observable<T>;
    options(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'arraybuffer';
        withCredentials?: boolean;
    }): Observable<ArrayBuffer>;
    options(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'blob';
        withCredentials?: boolean;
    }): Observable<Blob>;
    options(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'text';
        withCredentials?: boolean;
    }): Observable<string>;
    options(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'arraybuffer';
        withCredentials?: boolean;
    }): Observable<HttpEvent<ArrayBuffer>>;
    options(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string | number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'blob';
        withCredentials?: boolean;
    }): Observable<HttpEvent<Blob>>;
    options(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string | number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'text';
        withCredentials?: boolean;
    }): Observable<HttpEvent<string>>;
    options(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string | number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<HttpEvent<Object>>;
    options<T>(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string | number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<HttpEvent<T>>;
    options(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'arraybuffer';
        withCredentials?: boolean;
    }): Observable<HttpResponse<ArrayBuffer>>;
    options(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'blob';
        withCredentials?: boolean;
    }): Observable<HttpResponse<Blob>>;
    options(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'text';
        withCredentials?: boolean;
    }): Observable<HttpResponse<string>>;
    options(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<HttpResponse<Object>>;
    options<T>(url: string, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<HttpResponse<T>>;
    options(url: string, options?: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<Object>;
    options<T>(url: string, options?: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<T>;
    patch(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'arraybuffer';
        withCredentials?: boolean;
    }): Observable<ArrayBuffer>;
    patch(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'blob';
        withCredentials?: boolean;
    }): Observable<Blob>;
    patch(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'text';
        withCredentials?: boolean;
    }): Observable<string>;
    patch(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'arraybuffer';
        withCredentials?: boolean;
    }): Observable<HttpEvent<ArrayBuffer>>;
    patch(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'blob';
        withCredentials?: boolean;
    }): Observable<HttpEvent<Blob>>;
    patch(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'text';
        withCredentials?: boolean;
    }): Observable<HttpEvent<string>>;
    patch(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<HttpEvent<Object>>;
    patch<T>(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
```

```
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<HttpEvent<T>>;
    patch(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string | number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'arraybuffer';
        withCredentials?: boolean;
    }): Observable<HttpResponse<ArrayBuffer>>;
    patch(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string | number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'blob';
        withCredentials?: boolean;
    }): Observable<HttpResponse<Blob>>;
    patch(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string | number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'text';
        withCredentials?: boolean;
    }): Observable<HttpResponse<string>>;
    patch(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<HttpResponse<Object>>;
    patch<T>(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<HttpResponse<T>>;
    patch(url: string, body: any | null, options?: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<Object>;
    patch<T>(url: string, body: any | null, options?: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<T>;
    post(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'arraybuffer';
        withCredentials?: boolean;
    }): Observable<ArrayBuffer>;
    post(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'blob';
        withCredentials?: boolean;
    }): Observable<Blob>;
    post(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'text';
        withCredentials?: boolean;
    }): Observable<string>;
    post(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'arraybuffer';
        withCredentials?: boolean;
    }): Observable<HttpEvent<ArrayBuffer>>;
    post(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'blob';
        withCredentials?: boolean;
    }): Observable<HttpEvent<Blob>>;
    post(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'text';
        withCredentials?: boolean;
    }): Observable<HttpEvent<string>>;
    post(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<HttpEvent<Object>>;
    post<T>(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<HttpEvent<T>>;
    post(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'arraybuffer';
        withCredentials?: boolean;
    }): Observable<HttpResponse<ArrayBuffer>>;
    post(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'blob';
        withCredentials?: boolean;
    }): Observable<HttpResponse<Blob>>;
    post(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'text';
        withCredentials?: boolean;
    }): Observable<HttpResponse<string>>;
    post(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<HttpResponse<Object>>;
    post<T>(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<HttpResponse<T>>;
    post(url: string, body: any | null, options?: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<Object>;
    post<T>(url: string, body: any | null, options?: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<T>;
    put(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'arraybuffer';
        withCredentials?: boolean;
    }): Observable<ArrayBuffer>;
    put(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'blob';
        withCredentials?: boolean;
    }): Observable<Blob>;
    put(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'text';
        withCredentials?: boolean;
    }): Observable<string>;
    put(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
```

```
        };
        reportProgress?: boolean;
        responseType: 'arraybuffer';
        withCredentials?: boolean;
    }): Observable<HttpEvent<ArrayBuffer>>;
    put(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'blob';
        withCredentials?: boolean;
    }): Observable<HttpEvent<Blob>>;
    put(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'text';
        withCredentials?: boolean;
    }): Observable<HttpEvent<string>>;
    put(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<HttpEvent<Object>>;
    put<T>(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<HttpEvent<T>>;
    put(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'arraybuffer';
        withCredentials?: boolean;
    }): Observable<HttpResponse<ArrayBuffer>>;
    put(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'blob';
        withCredentials?: boolean;
    }): Observable<HttpResponse<Blob>>;
    put(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'text';
        withCredentials?: boolean;
    }): Observable<HttpResponse<string>>;
    put(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<HttpResponse<Object>>;
    put<T>(url: string, body: any | null, options: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<HttpResponse<T>>;
    put(url: string, body: any | null, options?: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<Object>;
    put<T>(url: string, body: any | null, options?: {
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<T>;
    request<R>(req: HttpRequest<any>): Observable<HttpEvent<R>>;
    request(method: string, url: string, options: {
        body?: any;
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'arraybuffer';
        withCredentials?: boolean;
    }): Observable<ArrayBuffer>;
    request(method: string, url: string, options: {
        body?: any;
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'blob';
        withCredentials?: boolean;
    }): Observable<Blob>;
    request(method: string, url: string, options: {
        body?: any;
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'text';
        withCredentials?: boolean;
    }): Observable<string>;
    request(method: string, url: string, options: {
        body?: any;
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        observe: 'events';
        reportProgress?: boolean;
        responseType: 'arraybuffer';
        withCredentials?: boolean;
    }): Observable<HttpEvent<ArrayBuffer>>;
    request(method: string, url: string, options: {
        body?: any;
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'blob';
        withCredentials?: boolean;
    }): Observable<HttpEvent<Blob>>;
    request(method: string, url: string, options: {
        body?: any;
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'events';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'text';
        withCredentials?: boolean;
    }): Observable<HttpEvent<string>>;
    request(method: string, url: string, options: {
        body?: any;
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        reportProgress?: boolean;
        observe: 'events';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<HttpEvent<any>>;
    request<R>(method: string, url: string, options: {
        body?: any;
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        reportProgress?: boolean;
        observe: 'events';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<HttpEvent<R>>;
    request(method: string, url: string, options: {
        body?: any;
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'arraybuffer';
        withCredentials?: boolean;
    }): Observable<HttpResponse<ArrayBuffer>>;
    request(method: string, url: string, options: {
        body?: any;
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'blob';
        withCredentials?: boolean;
    }): Observable<HttpResponse<Blob>>;
    request(method: string, url: string, options: {
        body?: any;
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        observe: 'response';
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        reportProgress?: boolean;
        responseType: 'text';
        withCredentials?: boolean;
    }): Observable<HttpResponse<string>>;
    request(method: string, url: string, options: {
        body?: any;
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        reportProgress?: boolean;
        observe: 'response';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<HttpResponse<Object>>;
    request<R>(method: string, url: string, options: {
        body?: any;
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        reportProgress?: boolean;
        observe: 'response';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        responseType?: 'json';
        withCredentials?: boolean;
    }): Observable<HttpResponse<R>>;
    request(method: string, url: string, options?: {
        body?: any;
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        responseType?: 'json';
        reportProgress?: boolean;
        withCredentials?: boolean;
    }): Observable<Object>;
    request<R>(method: string, url: string, options?: {
        body?: any;
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        observe?: 'body';
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        responseType?: 'json';
        reportProgress?: boolean;
        withCredentials?: boolean;
    }): Observable<R>;
    request(method: string, url: string, options?: {
        body?: any;
        headers?: HttpHeaders | {
            [header: string]: string | string[];
        };
        context?: HttpContext;
        params?: HttpParams | {
            [param: string]: string | number | boolean | ReadonlyArray<string |
number | boolean>;
        };
        observe?: 'body' | 'events' | 'response';
        reportProgress?: boolean;
        responseType?: 'arraybuffer' | 'blob' | 'json' | 'text';
        withCredentials?: boolean;
    }): Observable<any>;
    // (undocumented)
    static ɵfac: i0.ɵɵFactoryDeclaration<HttpClient, never>;
    // (undocumented)
    static ɵprov: i0.ɵɵInjectableDeclaration<HttpClient>;
}

// @public
export class HttpClientJsonpModule {
    // (undocumented)
    static ɵfac: i0.ɵɵFactoryDeclaration<HttpClientJsonpModule, never>;
    // (undocumented)
    static ɵinj: i0.ɵɵInjectorDeclaration<HttpClientJsonpModule>;
    // (undocumented)
    static ɵmod: i0.ɵɵNgModuleDeclaration<HttpClientJsonpModule, never, never,
never>;
}
```

```
// @public
export class HttpClientModule {
    // (undocumented)
    static ɵfac: i0.ɵɵFactoryDeclaration<HttpClientModule, never>;
    // (undocumented)
    static ɵinj: i0.ɵɵInjectorDeclaration<HttpClientModule>;
    // (undocumented)
    static ɵmod: i0.ɵɵNgModuleDeclaration<HttpClientModule, never, [typeof
HttpClientXsrfModule], never>;
}

// @public
export class HttpClientXsrfModule {
    static disable(): ModuleWithProviders<HttpClientXsrfModule>;
    static withOptions(options?: {
        cookieName?: string;
        headerName?: string;
    }): ModuleWithProviders<HttpClientXsrfModule>;
    // (undocumented)
    static ɵfac: i0.ɵɵFactoryDeclaration<HttpClientXsrfModule, never>;
    // (undocumented)
    static ɵinj: i0.ɵɵInjectorDeclaration<HttpClientXsrfModule>;
    // (undocumented)
    static ɵmod: i0.ɵɵNgModuleDeclaration<HttpClientXsrfModule, never, never,
never>;
}

// @public
export class HttpContext {
    delete(token: HttpContextToken<unknown>): HttpContext;
    get<T>(token: HttpContextToken<T>): T;
    has(token: HttpContextToken<unknown>): boolean;
    // (undocumented)
    keys(): IterableIterator<HttpContextToken<unknown>>;
    set<T>(token: HttpContextToken<T>, value: T): HttpContext;
}

// @public
export class HttpContextToken<T> {
    constructor(defaultValue: () => T);
    // (undocumented)
    readonly defaultValue: () => T;
}

// @public
export interface HttpDownloadProgressEvent extends HttpProgressEvent {
    partialText?: string;
    // (undocumented)
    type: HttpEventType.DownloadProgress;
}
```

```
// @public
export class HttpErrorResponse extends HttpResponseBase implements Error {
    constructor(init: {
        error?: any;
        headers?: HttpHeaders;
        status?: number;
        statusText?: string;
        url?: string;
    });
    // (undocumented)
    readonly error: any | null;
    // (undocumented)
    readonly message: string;
    // (undocumented)
    readonly name = "HttpErrorResponse";
    readonly ok = false;
}

// @public
export type HttpEvent<T> = HttpSentEvent | HttpHeaderResponse | HttpResponse<T> |
HttpProgressEvent | HttpUserEvent<T>;

// @public
export enum HttpEventType {
    DownloadProgress = 3,
    Response = 4,
    ResponseHeader = 2,
    Sent = 0,
    UploadProgress = 1,
    User = 5
}

// @public
export abstract class HttpHandler {
    // (undocumented)
    abstract handle(req: HttpRequest<any>): Observable<HttpEvent<any>>;
}

// @public
export class HttpHeaderResponse extends HttpResponseBase {
    constructor(init?: {
        headers?: HttpHeaders;
        status?: number;
        statusText?: string;
        url?: string;
    });
    clone(update?: {
        headers?: HttpHeaders;
        status?: number;
        statusText?: string;
        url?: string;
    }): HttpHeaderResponse;
    // (undocumented)
    readonly type: HttpEventType.ResponseHeader;
}

// @public
export class HttpHeaders {
    constructor(headers?: string | {
        [name: string]: string | string[];
    });
    append(name: string, value: string | string[]): HttpHeaders;
    delete(name: string, value?: string | string[]): HttpHeaders;
    get(name: string): string | null;
    getAll(name: string): string[] | null;
    has(name: string): boolean;
    keys(): string[];
    set(name: string, value: string | string[]): HttpHeaders;
}

// @public
export interface HttpInterceptor {
    intercept(req: HttpRequest<any>, next: HttpHandler): Observable<HttpEvent<any>>;
}

// @public
export interface HttpParameterCodec {
    // (undocumented)
    decodeKey(key: string): string;
    // (undocumented)
    decodeValue(value: string): string;
    // (undocumented)
    encodeKey(key: string): string;
    // (undocumented)
    encodeValue(value: string): string;
}

// @public
export class HttpParams {
    constructor(options?: HttpParamsOptions);
    append(param: string, value: string | number | boolean): HttpParams;
    appendAll(params: {
        [param: string]: string | number | boolean | ReadonlyArray<string | number |
boolean>;
    }): HttpParams;
    delete(param: string, value?: string | number | boolean): HttpParams;
    get(param: string): string | null;
    getAll(param: string): string[] | null;
    has(param: string): boolean;
    keys(): string[];
    set(param: string, value: string | number | boolean): HttpParams;
    toString(): string;
}
```

```
// @public
export interface HttpParamsOptions {
    encoder?: HttpParameterCodec;
    fromObject?: {
        [param: string]: string | number | boolean | ReadonlyArray<string | number |
boolean>;
    };
    fromString?: string;
}

// @public
export interface HttpProgressEvent {
    loaded: number;
    total?: number;
    type: HttpEventType.DownloadProgress | HttpEventType.UploadProgress;
}

// @public
export class HttpRequest<T> {
    constructor(method: 'DELETE' | 'GET' | 'HEAD' | 'JSONP' | 'OPTIONS', url:
string, init?: {
        headers?: HttpHeaders;
        context?: HttpContext;
        reportProgress?: boolean;
        params?: HttpParams;
        responseType?: 'arraybuffer' | 'blob' | 'json' | 'text';
        withCredentials?: boolean;
    });
    constructor(method: 'POST' | 'PUT' | 'PATCH', url: string, body: T | null,
init?: {
        headers?: HttpHeaders;
        context?: HttpContext;
        reportProgress?: boolean;
        params?: HttpParams;
        responseType?: 'arraybuffer' | 'blob' | 'json' | 'text';
        withCredentials?: boolean;
    });
    constructor(method: string, url: string, body: T | null, init?: {
        headers?: HttpHeaders;
        context?: HttpContext;
        reportProgress?: boolean;
        params?: HttpParams;
        responseType?: 'arraybuffer' | 'blob' | 'json' | 'text';
        withCredentials?: boolean;
    });
    readonly body: T | null;
    // (undocumented)
    clone(): HttpRequest<T>;
    // (undocumented)
    clone(update: {
        headers?: HttpHeaders;
        context?: HttpContext;
        reportProgress?: boolean;
        params?: HttpParams;
        responseType?: 'arraybuffer' | 'blob' | 'json' | 'text';
        withCredentials?: boolean;
        body?: T | null;
        method?: string;
        url?: string;
        setHeaders?: {
            [name: string]: string | string[];
        };
        setParams?: {
            [param: string]: string;
        };
    }): HttpRequest<T>;
    // (undocumented)
    clone<V>(update: {
        headers?: HttpHeaders;
        context?: HttpContext;
        reportProgress?: boolean;
        params?: HttpParams;
        responseType?: 'arraybuffer' | 'blob' | 'json' | 'text';
        withCredentials?: boolean;
        body?: V | null;
        method?: string;
        url?: string;
        setHeaders?: {
            [name: string]: string | string[];
        };
        setParams?: {
            [param: string]: string;
        };
    }): HttpRequest<V>;
    readonly context: HttpContext;
    detectContentTypeHeader(): string | null;
    readonly headers: HttpHeaders;
    readonly method: string;
    readonly params: HttpParams;
    readonly reportProgress: boolean;
    readonly responseType: 'arraybuffer' | 'blob' | 'json' | 'text';
    serializeBody(): ArrayBuffer | Blob | FormData | string | null;
    // (undocumented)
    readonly url: string;
    readonly urlWithParams: string;
    readonly withCredentials: boolean;
}

// @public
export class HttpResponse<T> extends HttpResponseBase {
    constructor(init?: {
        body?: T | null;
        headers?: HttpHeaders;
        status?: number;
        statusText?: string;
        url?: string;
    });
    readonly body: T | null;
    // (undocumented)
    clone(): HttpResponse<T>;
    // (undocumented)
    clone(update: {
        headers?: HttpHeaders;
        status?: number;
        statusText?: string;
        url?: string;
    }): HttpResponse<T>;
    // (undocumented)
    clone<V>(update: {
        body?: V | null;
        headers?: HttpHeaders;
        status?: number;
        statusText?: string;
        url?: string;
    }): HttpResponse<V>;
    // (undocumented)
    readonly type: HttpEventType.Response;
}

// @public
export abstract class HttpResponseBase {
    constructor(init: {
        headers?: HttpHeaders;
        status?: number;
        statusText?: string;
        url?: string;
    }, defaultStatus?: number, defaultStatusText?: string);
    readonly headers: HttpHeaders;
    readonly ok: boolean;
    readonly status: number;
    readonly statusText: string;
    readonly type: HttpEventType.Response | HttpEventType.ResponseHeader;
    readonly url: string | null;
}

// @public
export interface HttpSentEvent {
    // (undocumented)
    type: HttpEventType.Sent;
}

// @public
export const enum HttpStatusCode {
    // (undocumented)
    Accepted = 202,
    // (undocumented)
    AlreadyReported = 208,
    // (undocumented)
    BadGateway = 502,
    // (undocumented)
    BadRequest = 400,
    // (undocumented)
    Conflict = 409,
    // (undocumented)
    Continue = 100,
    // (undocumented)
    Created = 201,
    // (undocumented)
    EarlyHints = 103,
    // (undocumented)
    ExpectationFailed = 417,
    // (undocumented)
    FailedDependency = 424,
    // (undocumented)
    Forbidden = 403,
    // (undocumented)
    Found = 302,
    // (undocumented)
    GatewayTimeout = 504,
    // (undocumented)
    Gone = 410,
    // (undocumented)
    HttpVersionNotSupported = 505,
    // (undocumented)
    ImATeapot = 418,
    // (undocumented)
    ImUsed = 226,
    // (undocumented)
    InsufficientStorage = 507,
    // (undocumented)
    InternalServerError = 500,
    // (undocumented)
    LengthRequired = 411,
    // (undocumented)
    Locked = 423,
    // (undocumented)
    LoopDetected = 508,
    // (undocumented)
    MethodNotAllowed = 405,
    // (undocumented)
    MisdirectedRequest = 421,
    // (undocumented)
    MovedPermanently = 301,
    // (undocumented)
    MultipleChoices = 300,
    // (undocumented)
    MultiStatus = 207,
    // (undocumented)
    NetworkAuthenticationRequired = 511,
    // (undocumented)
    NoContent = 204,
    // (undocumented)
    NonAuthoritativeInformation = 203,
    // (undocumented)
    NotAcceptable = 406,
    // (undocumented)
    NotExtended = 510,
    // (undocumented)
    NotFound = 404,
    // (undocumented)
    NotImplemented = 501,
    // (undocumented)
    NotModified = 304,
    // (undocumented)
    Ok = 200,
    // (undocumented)
    PartialContent = 206,
    // (undocumented)
    PayloadTooLarge = 413,
    // (undocumented)
    PaymentRequired = 402,
    // (undocumented)
    PermanentRedirect = 308,
    // (undocumented)
    PreconditionFailed = 412,
    // (undocumented)
    PreconditionRequired = 428,
    // (undocumented)
    Processing = 102,
    // (undocumented)
    ProxyAuthenticationRequired = 407,
    // (undocumented)
    RangeNotSatisfiable = 416,
    // (undocumented)
    RequestHeaderFieldsTooLarge = 431,
    // (undocumented)
    RequestTimeout = 408,
    // (undocumented)
    ResetContent = 205,
    // (undocumented)
    SeeOther = 303,
    // (undocumented)
    ServiceUnavailable = 503,
    // (undocumented)
    SwitchingProtocols = 101,
    // (undocumented)
    TemporaryRedirect = 307,
    // (undocumented)
    TooEarly = 425,
    // (undocumented)
    TooManyRequests = 429,
    // (undocumented)
    Unauthorized = 401,
    // (undocumented)
    UnavailableForLegalReasons = 451,
    // (undocumented)
    UnprocessableEntity = 422,
    // (undocumented)
    UnsupportedMediaType = 415,
    // (undocumented)
    Unused = 306,
    // (undocumented)
    UpgradeRequired = 426,
    // (undocumented)
    UriTooLong = 414,
    // (undocumented)
    UseProxy = 305,
    // (undocumented)
    VariantAlsoNegotiates = 506
}

// @public
export interface HttpUploadProgressEvent extends HttpProgressEvent {
    // (undocumented)
    type: HttpEventType.UploadProgress;
}

// @public
export class HttpUrlEncodingCodec implements HttpParameterCodec {
    decodeKey(key: string): string;
    decodeValue(value: string): string;
    encodeKey(key: string): string;
    encodeValue(value: string): string;
}

// @public
export interface HttpUserEvent<T> {
    // (undocumented)
    type: HttpEventType.User;
}

// @public
export class HttpXhrBackend implements HttpBackend {
    constructor(xhrFactory: XhrFactory_2);
    handle(req: HttpRequest<any>): Observable<HttpEvent<any>>;
    // (undocumented)
    static ɵfac: i0.ɵɵFactoryDeclaration<HttpXhrBackend, never>;
    // (undocumented)
    static ɵprov: i0.ɵɵInjectableDeclaration<HttpXhrBackend>;
}

// @public
```

```
export abstract class HttpXsrfTokenExtractor {
    abstract getToken(): string | null;
}

// @public
export class JsonpClientBackend implements HttpBackend {
    constructor(callbackMap: JsonpCallbackContext, document: any);
    handle(req: HttpRequest<never>): Observable<HttpEvent<any>>;
    // (undocumented)
    static ɵfac: i0.ɵɵFactoryDeclaration<JsonpClientBackend, never>;
    // (undocumented)
    static ɵprov: i0.ɵɵInjectableDeclaration<JsonpClientBackend>;
}

// @public
export class JsonpInterceptor {
    constructor(jsonp: JsonpClientBackend);
    intercept(req: HttpRequest<any>, next: HttpHandler): Observable<HttpEvent<any>>;
    // (undocumented)
    static ɵfac: i0.ɵɵFactoryDeclaration<JsonpInterceptor, never>;
    // (undocumented)
    static ɵprov: i0.ɵɵInjectableDeclaration<JsonpInterceptor>;
}

// @public @deprecated
export type XhrFactory = XhrFactory_2;

// @public @deprecated
export const XhrFactory: typeof XhrFactory_2;

// (No @packageDocumentation comment for this package)
```