Frontend Code Documentation

Source code documentation of NBP Crasher application focusing on frontend implementation and usages

Table of contents

[Project Information 3](#_Toc188034543)

[Frontend Purpose 3](#_Toc188034544)

[Used Technologies 3](#_Toc188034545)

[Code documentation 4](#_Toc188034546)

[Application Configuration and Constants 4](#_Toc188034547)

[Models and Types 4](#_Toc188034548)

[BaseService 5](#_Toc188034549)

[StatsService 5](#_Toc188034550)

# Project Information

* **Project Name:** NBP Crasher
* **Project Manager:** Przemysław Kowalski
* **Report Date:** 17.01.2025
* **Project Duration:** 08.12.2024 – 31.01.2025

# Frontend Purpose

The goal of the frontend for the NBP Crasher project is to provide a GUI (Graphical User Interface – Presentation Layer) which displays data in user-friendly interface & communicates with our API for obtaining statistical measures based on currency exchange rates retrieved from an external API ([NBP API](https://api.nbp.pl/)).

# Used Technologies

This part of system was developed using easily available technologies.

The language used is [JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript) – standard ECMAScript 2024

UI was developed with:

* framework [Angular](https://angular.dev/) – version 17.3.0,
* [TypeScript](https://www.typescriptlang.org/) – version 5.4.2,
* [Bootstrap](https://getbootstrap.com/) – version 5.3.3,
* [DevExtreme Angular components](https://js.devexpress.com/Angular/) – version 23.1.13.

Tests written with [Karma](https://karma-runner.github.io/latest/index.html) test runner (version 5.1.0) & [Jasmine](https://jasmine.github.io/) testing framework (version 5.1.0).

For final deployment we used contenerization using [Docker](https://www.docker.com/)

# Code documentation

## Application Configuration and Constants

**appConfig:** Application configuration with providers.

**appName:** Application name.

**defaultCurrencyFrom:** Default "from" currency.

**defaultCurrencyTo:** Default "to" currency.

**minDateFrom:** Minimum date selectable (Based on NBP API docs).

**timeFrames:** Array of available time frames.

**defaultTimeFrameIndex:** Default time frame index (1W).

**defaultTimeFrame:** Default time frame (1W).

**dateFormat:** Date format.

**blankData:** Placeholder for blank data.

## Models and Types

**SupportedCurrency:** Supported currencies type definition.

**Currency:** Currency type with name property.

**supportedCurrencies:** Array of supported currencies.

**TimeFrame:** Time frame type extending DxTabsItem with optional dateFrom and dateTo properties.

**Statistics:** Statistics type definition.

**Session:** Session type definition.

**ChangeDistribution:** Change distribution type definition.

**Data:** Data type definition, aggregating statistics, sessions, and changes distribution.

## BaseService

**apiErrors:** Base API errors.

**notificationOverride:** Override settings for notifications.

**defaultError:** Default error configuration.

**generateParams(filters?: unknown): HttpParams** Generates HttpParams from given filters. Parameters:

* filters (optional): Filters to be converted into HttpParams.

Returns:

* HttpParams: The generated HttpParams.

**catchCustomError(error: HttpErrorResponse): Observable<void>** Handles custom HTTP errors and displays appropriate notifications. Parameters:

* error: The HTTP error response to handle.

Returns:

* Observable<void>: An observable that throws the processed error message.

## StatsService

**baseRoutePath:** Base route path for the stats API.

**get(filters: GetStatsRequest): Promise<Data>** Fetches statistics data based on given filters. Parameters:

* filters: Filters to apply to the stats request.

Returns:

* Promise<Data>: A promise that resolves to the fetched data.