Technical Documentation of Ninja Clock API Integration

1. Code Guidelines

1.1. Naming

- Variables: Use camelCase (e.g., clockResponse, NINJA_CLOCK_URL).
- Files: Lowercase with hyphens (e.g., ninja-clock-fetcher.js).
- Interfaces: Use PascalCase (e.g., ClockQuery, FetchClockResponse).

1.2. Code Style

- Favor const over let wherever possible.
- Use modern JavaScript features, including Promises and arrow functions.
- Handle exceptions with promise-based try-catch (i.e., .then() and .catch()).

2. Project Overview & Function Descriptions

This project fetches the current time for a specific city using the Ninja Clock API:
- Constants:
  - NINJA_CLOCK_URL: The base endpoint URL of the Ninja Clock API.
  - NINJA_CLOCK_API_KEY: API key for authenticating the Ninja Clock API requests.
- Interfaces:
  - ClockQuery: Expects a single property, city, which defines the city for which we want to fetch the time.
  - ClockResponse: Represents the expected successful response from the Ninja Clock API.
  - FetchClockResponse: A union type that can be either a successful or failed API response.
- fetchClock({city}: ClockQuery):
  - Builds the full API request URL using the provided city.
  - Makes an asynchronous fetch request to the API, including passing the API key in headers.
  - Processes the API response. If the request was successful, it formats and returns the clock data for the city. Otherwise, it throws a custom error.

3. Technologies

- Fetch API: To request and fetch data from the Ninja Clock API.
- Custom Middleware: ErrorStatus for enhanced error management and messaging.

4. Description of Clock Retrieval

The Ninja Clock API integration offers time details for specific cities:
- City-specific Time:
  - Provide a city name (like "London") and fetch the current time, timezone, and other related data for that city.
- Error Handling:
  - In case the API request encounters issues, it uses a custom ErrorStatus to provide detailed error information.