**Overview**

This documentation provides guidelines for maintaining and debugging the codebase related to tracking train services. The primary components include the TrackingArea component, useTrackingState hook, and the TrackPage component.

**Key Components**

1. TrackingArea Component

* File Path: src/app/track/TrackingArea.tsx
* Purpose: Displays tracking information for a specific train service.

**Props:**

* serviceToTrack: An object of type Journey, containing details about the train service.

**Key Functions:**

* getGlyphFromStatus: Returns a visual representation based on the service status.
* getDescriptionFromStatus: Provides a textual description of the service status.

2. *useTrackingState* Hook

* File Path: src/lib/hooks/useTrackState.tsx
* Purpose: Manages the state of the tracking information, including fetching updates from the server.

**Key Variables:**

* currentTrackingState: Holds the current state of the tracking data.
* timeTilRefresh: Determines how often the tracking data should be refreshed.

**Key Functions:**

* execute: Calls the server action to fetch the latest tracking data.
* setInterval: Periodically fetches new data based on the timeTilRefresh.

3. *TrackPage* Component

* File Path: src/app/track/page.tsx
* Purpose: Serves as the main page for tracking services, handling user interactions and displaying the TrackingArea component.

**Props:**

* searchParams: Contains query parameters from the URL, specifically the train information.

**Key Functions:**

* dissectTrainInfoFromUrl: Parses the train information from the URL.

**Debugging Guidelines**

**Common Errors**

1. Missing Properties in Props:
   * Error: Property 'updateKey' is missing in type...
   * Solution: Ensure the updateKey is included in the hidden state of currentTrackingState. This is set in the useTrackingState hook.
2. Incorrect Function Arguments:
   * Error: Expected 3 arguments, but got 1.
   * Solution: Check the function calls for getGlyphFromStatus and getDescriptionFromStatus. Ensure they are called with the correct number of arguments.

Debugging Steps

1. Console Logging:  
   Use console.log statements to track the flow of data and identify where the state may not be updating as expected.
2. Network Requests:  
   Monitor network requests in the browser’s developer tools to ensure the correct data is being fetched from the server.
3. Error Handling:  
   Implement robust error handling in the getServiceListCA function to catch and log errors when fetching data.

Maintenance Tips

* Code Comments: Keep comments updated to reflect any changes in logic or functionality.
* Type Definitions: Ensure TypeScript types are correctly defined and updated as data structures evolve.
* Testing: Regularly run tests to ensure components behave as expected, especially after making changes to the state management logic.