

Route Listing Table Enhancement

Date: November 28, 2025

Version: 0.37

Status:  Deployed to Production

Overview

Added a comprehensive account listing table beneath the Routes map that displays all filtered accounts in a sortable, scrollable table format. This enhancement provides route planners with a detailed view of all accounts for the selected technician, territory, and day filters.

Feature Details

What Was Added

Account Listing Table beneath the Routes map that displays:

- **Account Number** - Unique customer identifier
- **Customer Name** - Full customer name
- **Street Address** - Service location street address
- **City** - City name
- **State** - Always "AZ" (Arizona)
- **ZIP Code** - 5-digit ZIP code
- **Days of Service** - Comma-separated list of service days

Key Features

1. Dynamic Filtering

- Table automatically updates based on selected filters:
 - Technician (required)
 - Territory (optional)
 - Day of Service (optional)

2. Smart Display

- Only appears when a technician is selected and has routes
- Shows account count in header (e.g., "Account Listing (74 accounts)")
- Accounts sorted alphabetically by account number

3. Optimized UX

- Sticky table header stays visible while scrolling
- Maximum height of 600px with vertical scrolling
- Hover effect on rows for better readability
- Responsive column widths for optimal viewing

4. Data Quality

- Handles missing data gracefully (shows "N/A" or "Unknown")

- Validates arrays for days of service
 - Uses accurate geocoded addresses (ROOFTOP-level precision)
-

Technical Implementation

Files Modified

```
/nextjs_space/components/routes-map-view.tsx
```

- Added Table component imports
- Created new table section after map Card
- Conditional rendering based on selectedTechnician and filteredRoutes
- Implemented sorting by account number
- Added sticky header with z-index for scroll behavior

Component Structure

```
/* Account Listing Table */
{selectedTechnician && filteredRoutes.length > 0 && (
  <Card className="p-4">
    <h3 className="text-lg font-semibold mb-4">
      Account Listing ({filteredRoutes.length} accounts)
    </h3>
    <div className="rounded-md border overflow-auto max-h-[600px]">
      <Table>
        <TableHeader className="sticky top-0 bg-background z-10">
          {/* Column headers */}
        </TableHeader>
        <TableBody>
          {filteredRoutes
            .sort((a, b) => a.customerNumber.localeCompare(b.customerNumber))
            .map((route) => (
              <TableRow key={route.customerNumber}>
                {/* Row data */}
              </TableRow>
            )))
        </TableBody>
      </Table>
    </div>
  </Card>
)}
```

Styling Details

- **Table Container:** Rounded border with max-height constraint
- **Header:** Sticky positioning with background color
- **Columns:** Fixed widths for consistent layout
- Account #: 120px
- Customer Name: min 200px
- Street Address: min 250px
- City: 140px
- State: 80px
- ZIP Code: 90px
- Days of Service: min 150px

- **Rows:** Hover effect for better UX
-

Use Cases

1. Route Planning

- View all stops for a technician in table format
- Quickly scan addresses and service days
- Export data for route optimization tools

2. Territory Analysis

- Filter by territory to see cross-territory assignments
- Identify geographic clusters for efficiency improvements

3. Service Scheduling

- Filter by day to see specific day assignments
- Validate service day distribution
- Identify scheduling conflicts

4. Account Verification

- Verify customer information accuracy
- Cross-reference with CRM systems
- Audit address data quality

Testing

Test Scenarios

1. No Technician Selected

- Table does not appear
- Only map controls visible

2. Single Technician Selected

- Table appears below map
- Shows all accounts for that technician
- Header shows correct count

3. Territory Filter Applied

- Table updates to show only selected territory
- Count updates correctly

4. Day Filter Applied

- Table shows only accounts with that service day
- Count reflects filtered results

5. Combined Filters

- All filters work together correctly
- Table updates in real-time

6. **Data Quality**

- Missing data shows “N/A” or “Unknown”
- Days of service display correctly
- Sorting works alphabetically

7. **Scroll Behavior**

- Header stays visible while scrolling
- Table scrolls smoothly
- No layout shifts

Browser Compatibility

- Chrome/Edge (Chromium)
 - Firefox
 - Safari
 - Mobile browsers (responsive)
-

Deployment

Production URL: <https://phoenixnewlocations.abacusai.app>

Deployment Status: Live

Build Results:

✓ Compiled successfully			
✓ Generating static pages (5/5)			
Route (app)	Size	First Load	JS
` f /	81.3 kB	168 kB	

Checkpoint: “Added account listing table to Routes view”

Performance Considerations

1. Memoization

- filteredRoutes already memoized in parent component
- Sorting happens only when filteredRoutes changes

2. Rendering Optimization

- Table only renders when data exists
- Conditional rendering prevents unnecessary DOM updates

3. Scroll Performance

- Max height prevents excessive DOM rendering
 - Sticky header uses CSS (no JavaScript overhead)
-

Future Enhancements (Optional)

1. Export Functionality

- Add CSV/Excel export button
- Include current filter settings in export

2. Column Sorting

- Click column headers to sort
- Multi-column sort support

3. Search/Filter

- Add quick search box
- Filter by account number or name

4. Row Selection

- Select multiple accounts
- Bulk actions (e.g., print route sheet)

5. Print View

- Optimized print layout
 - Include technician info and date
-

Related Documentation

- [Performance Improvements Summary](#) (PERFORMANCE_IMPROVEMENTS_SUMMARY.md)
 - [Route View Fixes Summary](#) (ROUTE_VIEW_FIXES_SUMMARY.md)
 - [Geocoding Fix Summary](#) (GEOCODING_FIX_SUMMARY.md)
 - [Customer Data Fix Summary](#) (CUSTOMER_DATA_FIX_SUMMARY.md)
-

Summary

The Route Listing Table enhancement provides route planners with a comprehensive, filterable view of all accounts for a selected technician. The table integrates seamlessly with existing filters, maintains high performance, and handles data quality issues gracefully. This feature completes the Route Info tool suite, offering both visual (map) and tabular (list) views of route data.