

Enrichify Lead System Documentation

Overview

The Enrichify Lead System is a React-based application designed for managing and processing lead data. It provides functionality for viewing leads, uploading CSV files for processing, and exporting lead information in various formats.

Features

Core Functionality

- **Lead Management:** View, filter, and manage lead data
- **File Processing:** Upload and process CSV files for lead enrichment
- **Data Export:** Export leads as CSV or PDF files
- **DataZapp Integration:** Upload and process DataZapp files
- **Date Filtering:** Filter leads by date range
- **Bulk Operations:** Select multiple leads for batch operations

Architecture

Main Components

1. App Component

The main application component that manages the overall state and routing between different tabs.

Key State Variables:

- `currentPage`: Current pagination page
- `loading`: Loading state indicator
- `selectedUsers`: Array of selected user IDs
- `results`: Processed results from file uploads
- `startDate/endDate`: Date filter range
- `activeTab`: Current active tab ('leads' or 'datazapp')
- `users`: Array of lead data
- `totalCount`: Total number of leads

2. RenderDataZappTab Component

Handles DataZapp file uploads and processing.

Features:

- File upload with progress indication
- Automatic CSV download after processing
- Error handling and user feedback

API Endpoints

The application communicates with a backend server running on `http://localhost:5000`. Key endpoints include:

Lead Management

- `GET /leads` - Fetch paginated leads with optional date filters
- `GET /totalLeads` - Get total count of leads

File Processing

- `POST /enrichifystatcounter` - Process CSV files for lead enrichment
- `POST /upload-datazapp` - Process DataZapp files
- `POST /reuploadfile` - Re-upload files for processing

Functions Reference

File Upload and Processing

`handleFileUpload(event)`

Processes CSV file uploads for lead enrichment.

Parameters:

- `event`: File input change event

Process:

1. Creates FormData with selected file
2. Sends POST request to `/enrichifystatcounter`
3. Automatically downloads processed CSV
4. Updates lead data and counts

`handleDataZappUpload(event)`

Handles DataZapp file uploads and processing.

Parameters:

- `event`: File input change event

Features:

- Supports CSV and XLSX files
- Automatic filename generation with timestamp
- Success notifications and error handling

Data Export

`handleExportCSV()`

Exports selected leads to CSV format.

CSV Headers:

- First Name, Last Name, Email, Address, State
- Lead Source, Lead Quality, Credit Score
- Optional fields: Exit_url, Entry_url, URL, Title, Income, Phone

`handleExportLead(user)`

Exports individual lead data to PDF format using jsPDF.

PDF Features:

- Professional formatting
- Page break handling for long content
- URL wrapping for long links
- Generation timestamp

`convertToCSV(data)` / `convertToCSVTwo(data)`

Utility functions for converting JSON data to CSV format.

Features:

- Handles special characters and commas

- Quotes escaping
- Null/undefined value handling
- Object and array serialization

Data Management

`fetchLeads()`

Retrieves paginated lead data from the server.

Parameters (via state):

- `currentPage`: Page number for pagination
- `startDate/endDate`: Date filter range
- `usersPerPage`: Number of leads per page (default: 10)

`fetchCount()`

Retrieves total lead count from the server.

User Interface

`handleSelectUser(userId)` / `handleSelectAll()`

Manages lead selection for bulk operations.

Date Filtering

- `handleStartDateChange(e)`: Updates start date filter
- `handleEndDateChange(e)`: Updates end date filter
- `clearFilters()`: Resets all date filters

Component Structure

App

- |— Tab Navigation
 - | |— View Leads Tab
 - | |— DataZapp Upload Tab
- |— Filter Section
 - | |— Date Filters
 - | |— Action Buttons
 - | |— Results Counter
- |— Data Table (Leads Tab)
 - | |— Lead Information
 - | |— URL Links
 - | |— Export Options
 - | |— Selection Checkboxes
- |— DataZapp Upload (DataZapp Tab)
 - | |— File Upload
 - | |— Processing Status
 - | |— Download Options
- |— Popup Modal
 - | |— Leads Pulled Information

Dependencies

Required Libraries

- `react`: Core React library
- `axios`: HTTP client for API requests
- `jspdf`: PDF generation library

File Handling

- FormData API for file uploads
- Blob API for file downloads
- URL.createObjectURL for download links

Usage Guide

Viewing Leads

1. Navigate to the "View Leads" tab
2. Use date filters to narrow results
3. Select leads using checkboxes

4. Export selected leads or individual PDFs

Uploading Files

1. Click "Upload CSV File" for lead enrichment
2. Select CSV file from your system
3. Wait for processing completion
4. Processed file downloads automatically

DataZapp Processing

1. Navigate to "Upload DataZapp File" tab
2. Click "Upload & Process DataZapp File"
3. Select CSV or XLSX file
4. Processing results download automatically

Exporting Data

- **Individual PDF:** Click "Export Lead" for specific leads
- **Bulk CSV:** Select multiple leads and click "Export Selected as CSV"
- **Processed Data:** Downloads happen automatically after file processing

Error Handling

The application includes comprehensive error handling:

- File upload errors with user notifications
- API request error catching and user feedback
- Loading states during processing
- Input validation for file types

Styling

The application uses CSS classes for styling:

- `.business-container`, `.business-header`: Main layout
- `.tab-navigation`, `.tab-button`: Tab interface
- `.business-table`: Data table styling
- `.loading-widget`, `.spinner`: Loading indicators
- `.popup-overlay`, `.popup-content`: Modal dialogs

Best Practices

Performance

- Pagination for large datasets
- Loading states for user feedback
- Efficient state management

User Experience

- Clear error messages
- Progress indicators
- Intuitive navigation
- Responsive design considerations

Data Security

- File type validation
- Error boundary implementation
- Proper data sanitization in CSV exports

Troubleshooting

Common Issues

1. File Upload Failures

- Check file format (CSV/XLSX only)
- Verify server is running on localhost:5000
- Check file size limitations

2. CSV Export Issues

- Ensure leads are selected
- Check browser download settings
- Verify data formatting

3. PDF Generation Problems

- Long URLs may cause formatting issues
- Large datasets may impact performance
- Check jsPDF library compatibility

Development Notes

- Server must be running on `http://localhost:5000`
- File input elements are hidden and triggered programmatically
- CSV processing includes automatic character escaping
- PDF generation handles page breaks automatically

Future Enhancements

Potential improvements could include:

- Real-time processing status updates
- Advanced filtering options
- Batch PDF export functionality
- Data validation before upload
- User authentication and permissions
- Dashboard analytics and reporting