

TSQL.APP Universal React Client

Complete Documentation

Version 1.0.2 • Client 251005.1 • React 17.0.2

Read Documentation

Executive Summary

The TSQL.APP Universal React Client is a fully **application-agnostic**, metadata-driven frontend that can render *any* TSQL.APP application dynamically from pure JSON metadata sent by your SQL Server backend.

No hardcoded forms, no per-app builds — one single React client powers unlimited enterprise applications.

Core Principles

- **Zero Hardcoding** — UI is 100% defined by backend JSON
- **Metadata-Driven** — Views, fields, validation, actions all come from SQL
- **Universal** — Same client works for every TSQL.APP application
- **Backend Controls Everything** — Server decides layout, behavior and permissions

Startup & Configuration

The client discovers its backend automatically:

- Checks for a static base URL (for CDN builds)
- Reads cached API URL from cookie
- Queries the public discovery service
- Falls back to localhost or a default endpoint

Custom CSS is loaded dynamically from the same discovery endpoint.

State Management

Uses a single centralized reducer with React Context (no Redux). All UI changes go through dispatched actions such as `json`, `save`, `filter`, `pathname`, etc.

Routing System

Custom history-based routing (not React Router) driven by `pathname` actions and the browser History API. Navigation updates URL and triggers fresh data fetches automatically.

Authentication

OAuth 2.0 with automatic token refresh. Tokens are stored securely in http-only cookies. A fallback identity provider ensures high availability.

API Communication

All requests include rich context headers (current path, selected record, client UUID, etc.) so the backend always knows exactly what the user is doing.

GET for navigation/filtering • POST for saves, actions and file uploads.

View Rendering

Backend sends an array of views (Listview, Detailview, Editdetailview, Modalview, etc.). The client maps each `comp` value to the corresponding React component and injects the associated data and field definitions.

Key Components

- **Listview** — sortable, filterable, multi-select tables
- **Detailview / Editdetailview** — tabbed forms with dynamic fields
- **Action Menu** — buttons driven by backend actions array
- **Toasts & Modals** — notifications and popups

Monaco Code Editor Integration

Full-featured T-SQL editor with custom language support, live autocomplete for variables, keywords and database objects, and instant save on Ctrl+S / F5.

Technologies & Dependencies

React 17 • Bootstrap 4 • Monaco Editor • react-dropzone • react-datepicker • recharts • many lightweight utilities.

Advanced Features

- Multi-row selection with action support
- Powerful filtering syntax in URL
- Excel/CSV export (up to 100k rows)
- File drag-and-drop upload
- Print & screenshot actions
- Designer mode (column reorder, field rename)
- Per-app/user custom CSS injection
- Global keyboard shortcuts
- Mobile-aware layout

Conclusion

This universal client completely separates frontend development from business logic. Backend developers can deliver rich, consistent, enterprise-grade web applications using only T-SQL — with one shared, continuously improved React client powering them all.

One client. Infinite applications.

Document Version: 1.0 • Last Updated: 2025-10-13 • Client Version: 251005.1

← Back to [tsql.app](#)