

ASMC Admin Panel - Component Documentation

This document provides comprehensive documentation for all components in the ASMC Admin Panel, including their structure, props, usage, and examples.

Table of Contents

- [Component Architecture](#)
- [Common Components](#)
- [Admin Management Components](#)
- [Layout Components](#)
- [Form Components](#)
- [Data Display Components](#)
- [Navigation Components](#)
- [Utility Components](#)

Component Architecture

Component Structure Pattern

The admin panel follows a consistent component structure:

```
// Component Template
import React from 'react';
import { PropTypes } from 'prop-types';
import { ComponentName } from './ComponentName.styles';

const ComponentName = ({
  prop1,
  prop2,
  onAction,
  className,
  ...props
}) => {
  // Component logic

  return (
    <ComponentName className={className} {...props}>
      {/* Component JSX */}
    </ComponentName>
  );
};

ComponentName.propTypes = {
  prop1: PropTypes.string.isRequired,
  prop2: PropTypes.number,
  onAction: PropTypes.func,
  className: PropTypes.string,
};

ComponentName.defaultProps = {
  prop2: 0,
```

```
    onAction: () => {},
    className: '',
};

export default ComponentName;
```

Component Categories

1. **Common Components:** Reusable UI components
2. **Admin Components:** Business logic components
3. **Layout Components:** Structure and navigation
4. **Form Components:** Input and validation
5. **Data Display:** Tables, charts, and lists

Common Components

Input Component

Location: /src/components/Common/Input.jsx

A reusable input component with validation and styling.

```
import Input from '../Common/Input';

<Input
  name="memberName"
  label="Member Name"
  value={memberName}
  onChange={handleChange}
  error={errors.memberName}
  helperText="Enter the full name"
  required
  fullWidth
/>
```

Props:

- `name` (string): Input field name
- `label` (string): Input label
- `value` (string): Input value
- `onChange` (function): Change handler
- `error` (boolean): Error state
- `helperText` (string): Help text
- `required` (boolean): Required field
- `fullWidth` (boolean): Full width styling

Button Component

Location: /src/components/Common/Button.jsx

Customizable button component with various styles.

```
import Button from '../Common/Button';

<Button
```

```
variant="contained"
color="primary"
size="large"
onClick={handleSubmit}
disabled={isSubmitting}
startIcon={<SaveIcon />}
>
  Save Member
</Button>
```

Props:

- variant (string): 'contained', 'outlined', 'text'
- color (string): 'primary', 'secondary', 'error'
- size (string): 'small', 'medium', 'large'
- onClick (function): Click handler
- disabled (boolean): Disabled state
- startIcon (element): Start icon
- endIcon (element): End icon

Table Component

Location: /src/components/Common/Table.jsx

Data table with pagination, sorting, and filtering.

```
import Table from './Common/Table';

<Table
  data={members}
  columns={memberColumns}
  onEdit={handleEdit}
  onDelete={handleDelete}
  onPageChange={handlePageChange}
  loading={isLoading}
  pagination={{
    page: currentPage,
    count: totalPages,
    rowsPerPage: pageSize,
  }}
/>
```

Props:

- data (array): Table data
- columns (array): Column configuration
- onEdit (function): Edit handler
- onDelete (function): Delete handler
- onPageChange (function): Page change handler
- loading (boolean): Loading state
- pagination (object): Pagination configuration

Modal Component

Location: /src/components/Common/CommonModal.jsx

Reusable modal dialog component.

```
import CommonModal from './Common/CommonModal';

<CommonModal
  open={isOpen}
  onClose={handleClose}
  title="Edit Member"
  size="md"
>
  <MemberForm
    member={selectedMember}
    onSubmit={handleSubmit}
    onCancel={handleClose}
  />
</CommonModal>
```

Props:

- `open` (boolean): Modal visibility
- `onClose` (function): Close handler
- `title` (string): Modal title
- `size` (string): 'sm', 'md', 'lg', 'xl'
- `children` (element): Modal content

Select Component

Location: /src/components/Common/Select.jsx

Dropdown select component with search.

```
import Select from './Common/Select';

<Select
  name="status"
  label="Member Status"
  value={status}
  onChange={handleChange}
  options={statusOptions}
  multiple={false}
  searchable={true}
/>
```

Props:

- `name` (string): Field name
- `label` (string): Field label
- `value` (any): Selected value
- `onChange` (function): Change handler
- `options` (array): Select options
- `multiple` (boolean): Multiple selection
- `searchable` (boolean): Search functionality

Admin Management Components

Members Manager

Location: /src/components/admin/members-manager/

Complete member management system with CRUD operations.

MembersList Component

```
import MembersList from './members-manager/MembersList';

<MembersList
  members={members}
  loading={isLoading}
  onEdit={handleEdit}
  onDelete={handleDelete}
  onView={handleView}
  onFilter={handleFilter}
  pagination={pagination}
/>
```

MemberForm Component

```
import MemberForm from './members-manager/MemberForm';

<MemberForm
  member={selectedMember}
  onSubmit={handleSubmit}
  onCancel={handleCancel}
  mode={mode} // 'create' | 'edit' | 'view'
/>
```

MemberDetails Component

```
import MemberDetails from './members-manager/MemberDetails';

<MemberDetails
  member={member}
  onEdit={handleEdit}
  onDelete={handleDelete}
  showActions={true}
/>
```

Booking Manager

Location: /src/components/admin/booking-manager/

Hall and event booking management system.

BookingList Component

```
import BookingList from './booking-manager/BookingList';

<BookingList
```

```
bookings={bookings}
loading={isLoading}
onApprove={handleApprove}
onReject={handleReject}
onEdit={handleEdit}
filters={filters}
/>
```

BookingForm Component

```
import BookingForm from './booking-manager/BookingForm';

<BookingForm
  booking={selectedBooking}
  onSubmit={handleSubmit}
  onCancel={handleCancel}
  facilities={facilities}
  timeSlots={timeSlots}
/>
```

BookingCalendar Component

```
import BookingCalendar from './booking-manager/BookingCalendar';

<BookingCalendar
  bookings={bookings}
  onDateSelect={handleDateSelect}
  onBookingSelect={handleBookingSelect}
  view="month" // 'day' | 'week' | 'month'
/>
```

Events Manager

Location: /src/components/admin/events-manager/

Event creation and management system.

EventsList Component

```
import EventsList from './events-manager/EventsList';

<EventsList
  events={events}
  loading={isLoading}
  onEdit={handleEdit}
  onDelete={handleDelete}
  onPublish={handlePublish}
  filters={filters}
/>
```

EventForm Component

```
import EventForm from '../events-manager/EventForm';

<EventForm
  event={selectedEvent}
  onSubmit={handleSubmit}
  onCancel={handleCancel}
  mode={mode}
/>
```

Facility Manager

Location: /src/components/admin/facility-manager/

Facility information and management system.

FacilityList Component

```
import FacilityList from '../facility-manager/FacilityList';

<FacilityList
  facilities={facilities}
  loading={isLoading}
  onEdit={handleEdit}
  onDelete={handleDelete}
  onView={handleView}
/>
```

FacilityForm Component

```
import FacilityForm from '../facility-manager/FacilityForm';

<FacilityForm
  facility={selectedFacility}
  onSubmit={handleSubmit}
  onCancel={handleCancel}
  categories={categories}
/>
```

Staff Manager

Location: /src/components/admin/staff-manager/

Staff member management and role assignment.

StaffList Component

```
import StaffList from '../staff-manager/StaffList';

<StaffList
  staff={staff}
  loading={isLoading}
  onEdit={handleEdit}
  onDelete={handleDelete}
```

```
        onRoleChange={handleRoleChange}
    />
```

StaffForm Component

```
import StaffForm from '../staff-manager/StaffForm';

<StaffForm
  staff={selectedStaff}
  onSubmit={handleSubmit}
  onCancel={handleCancel}
  roles={availableRoles}
/>
```

Payment Manager

Location: /src/components/admin/payment-manager/

Payment processing and verification system.

PaymentList Component

```
import PaymentList from '../payment-manager/PaymentList';

<PaymentList
  payments={payments}
  loading={isLoading}
  onVerify={handleVerify}
  onRefund={handleRefund}
  onExport={handleExport}
  filters={filters}
/>
```

PaymentDetails Component

```
import PaymentDetails from '../payment-manager/PaymentDetails';

<PaymentDetails
  payment={payment}
  onVerify={handleVerify}
  onRefund={handleRefund}
  showActions={true}
/>
```

Layout Components

Header Component

Location: /src/components/layout/Header.jsx

Application header with navigation and user menu.

```
import Header from '../layout/Header';

<Header
  user={currentUser}
  onLogout={handleLogout}
  onProfileClick={handleProfileClick}
  notifications={notifications}
/>
```

Features:

- User profile dropdown
- Notification center
- Search functionality
- Theme toggle
- Logout functionality

Sidebar Component

Location: /src/components/layout/Sidebar.jsx

Navigation sidebar with menu items.

```
import Sidebar from '../layout/Sidebar';

<Sidebar
  menuItems={menuItems}
  selectedItem={selectedItem}
  onItemSelected={handleItemSelected}
  collapsed={isCollapsed}
  onToggle={handleToggle}
/>
```

Features:

- Collapsible navigation
- Icon and text labels
- Active state indication
- Permission-based visibility

MenuList Component

Location: /src/components/layout/MenuList.jsx

Dynamic menu list with permission filtering.

```
import MenuList from '../layout/MenuList';

<MenuList
  modules={modules}
  userPermissions={permissions}
  onItemClick={handleItemClick}
  level={0}
/>
```

Breadcrumbs Component

Location: /src/components/layout/Breadcrumbs.jsx

Navigation breadcrumbs for page hierarchy.

```
import Breadcrumbs from '../layout/Breadcrumbs';

<Breadcrumbs
  items={breadcrumbItems}
  onItemClick={handleItemClick}
  separator="/"
/>
```

Form Components

DatePicker Component

Location: /src/components/Common/DatePicker.jsx

Date selection component with validation.

```
import DatePicker from '../Common/DatePicker';

<DatePicker
  name="startDate"
  label="Start Date"
  value={startDate}
  onChange={handleDateChange}
  minDate={new Date()}
  maxDate={maxDate}
  required
/>
```

TimePicker Component

Location: /src/components/Common/TimePicker.jsx

Time selection component.

```
import TimePicker from '../Common/TimePicker';

<TimePicker
  name="startTime"
  label="Start Time"
  value={startTime}
  onChange={handleTimeChange}
  format="24h"
/>
```

MultipleSelect Component

Location: /src/components/Common/MultipleSelectArray.jsx

Multiple selection component for arrays.

```
import MultipleSelectArray from '../Common/MultipleSelectArray';

<MultipleSelectArray
  name="categories"
  label="Categories"
  value={categories}
  onChange={handleChange}
  options={categoryOptions}
  maxHeight={300}
/>
```

FileUpload Component

Location: /src/components/Common/UploadFile.jsx

File upload component with validation.

```
import UploadFile from '../Common/UploadFile';

<UploadFile
  name="profileImage"
  label="Profile Image"
  value={profileImage}
  onChange={handleFileChange}
  accept="image/*"
  maxSize={5 * 1024 * 1024} // 5MB
  onError={handleUploadError}
/>
```

Data Display Components

CardSummary Component

Location: /src/components/Common/CardSummary.jsx

Summary card for displaying key metrics.

```
import CardSummary from '../Common/CardSummary';

<CardSummary
  title="Total Members"
  value={totalMembers}
  icon={}
  color="primary"
  trend={+12}
  trendLabel="vs last month"
/>
```

TableFilter Component

Location: /src/components/Common/TableFilter.jsx

Advanced filtering component for tables.

```
import TableFilter from './Common/TableFilter';

<TableFilter
  filters={filterConfig}
  values={filterValues}
  onChange={handleFilterChange}
  onReset={handleFilterReset}
/>
```

ExportData Component

Location: /src/components/Common/ExportData.jsx

Data export functionality.

```
import ExportData from './Common/ExportData';

<ExportData
  data={tableData}
  columns={exportColumns}
  filename="members"
  formats={['csv', 'excel', 'pdf']}
  onExport={handleExport}
/>
```

Navigation Components

MultiLevelMenu Component

Location: /src/components/layout/MultiLevelMenu.jsx

Multi-level navigation menu.

```
import MultiLevelMenu from './layout/MultiLevelMenu';

<MultiLevelMenu
  items={menuItems}
  onItemClick={handleItemClick}
  level={0}
  maxLevel={3}
/>
```

SingleLevelMenu Component

Location: /src/components/layout/SingleLevelMenu.jsx

Single-level navigation menu.

```
import SingleLevelMenu from './layout/SingleLevelMenu';

<SingleLevelMenu
  items={menuItems}
```

```
    selectedItem={selectedItem}
    onClick={handleItemClick}
/>
```

Utility Components

HasPermission Component

Location: /src/components/Common/HasPermission.jsx

Permission-based component rendering.

```
import HasPermission from './Common/HasPermission';

<HasPermission permission="members:write">
  <CreateMemberButton />
</HasPermission>

<HasPermission permission="members:read">
  <MembersList />
</HasPermission>
```

RequirePermission Component

Location: /src/components/Common/RequirePermission.jsx

Permission requirement wrapper.

```
import RequirePermission from './Common/RequirePermission';

<RequirePermission
  permission="admin:access"
  fallback={<>AccessDenied />}>
  <AdminPanel />
</RequirePermission>
```

SearchRecords Component

Location: /src/components/Common/SearchRecords.jsx

Search functionality component.

```
import SearchRecords from './Common/SearchRecords';

<SearchRecords
  placeholder="Search members..."
  onSearch={handleSearch}
  loading={isSearching}
  results={searchResults}
  onResultClick={handleResultClick}
/>
```

Snackbar Component

Location: /src/components/Common/Snackbar.jsx

Notification snackbar component.

```
import Snackbar from './Common/Snackbar';

<Snackbar
  open={showNotification}
  message="Member created successfully"
  severity="success"
  onClose={handleClose}
  autoHideDuration={6000}
/>
```

Transition Component

Location: /src/components/Common/Transition.jsx

Animation transition wrapper.

```
import Transition from './Common/Transition';

<Transition type="fade" in={isVisible}>
  <div>Animated content</div>
</Transition>
```

Component Usage Examples

Complete Form Example

```
import React, { useState } from 'react';
import {
  Box,
  Grid,
  Paper,
  Typography,
} from '@mui/material';
import {
  Input,
  Select,
  DatePicker,
  Button,
  CommonModal,
} from '../Common';

const MemberForm = ({ member, onSubmit, onCancel }) => {
  const [formData, setFormData] = useState(member || {});
  const [errors, setErrors] = useState({});

  const handleChange = (name, value) => {
    setFormData(prev => ({
      ...prev,
```

```

        [name]: value
    }));
};

const handleSubmit = () => {
    // Validation logic
    const newErrors = validateForm(formData);

    if (Object.keys(newErrors).length === 0) {
        onSubmit(formData);
    } else {
        setErrors(newErrors);
    }
};

return (
    <Paper sx={{ p: 3 }}>
        <Typography variant="h6" gutterBottom>
            Member Information
        </Typography>

        <Grid container spacing={2}>
            <Grid item xs={12} md={6}>
                <Input
                    name="name"
                    label="Full Name"
                    value={formData.name || ''}
                    onChange={(e) => handleChange('name', e.target.value)}
                    error={!errors.name}
                    helperText={errors.name}
                    required
                />
            </Grid>

            <Grid item xs={12} md={6}>
                <Input
                    name="email"
                    label="Email"
                    type="email"
                    value={formData.email || ''}
                    onChange={(e) => handleChange('email', e.target.value)}
                    error={!errors.email}
                    helperText={errors.email}
                    required
                />
            </Grid>

            <Grid item xs={12} md={6}>
                <Select
                    name="status"
                    label="Status"
                    value={formData.status || ''}

```

```

        onChange={(value) => handleChange('status', value)}
        options={statusOptions}
        required
      />
    </Grid>

    <Grid item xs={12} md={6}>
      <DatePicker
        name="joinDate"
        label="Join Date"
        value={formData.joinDate}
        onChange={(date) => handleChange('joinDate', date)}
        required
      />
    </Grid>
  </Grid>

  <Box sx={{ mt: 3, display: 'flex', gap: 2, justifyContent: 'flex-end' }}>
    <Button variant="outlined" onClick={onCancel}>
      Cancel
    </Button>
    <Button variant="contained" onClick={handleSubmit}>
      Save Member
    </Button>
  </Box>
</Paper>
);
};

export default MemberForm;

```

Data Table Example

```

import React, { useState, useEffect } from 'react';
import {
  Box,
  Typography,
  IconButton,
} from '@mui/material';
import {
  Edit as EditIcon,
  Delete as DeleteIcon,
  Visibility as ViewIcon,
} from '@mui/icons-material';
import {
  Table,
  TableFilter,
  ExportData,
  Button,
} from '../Common';

const MembersTable = () => {

```

```

const [members, setMembers] = useState([]);
const [loading, setLoading] = useState(false);
const [filters, setFilters] = useState({});
const [pagination, setPagination] = useState({
  page: 0,
  rowsPerPage: 10,
  count: 0,
});

const columns = [
  {
    id: 'name',
    label: 'Name',
    sortable: true,
  },
  {
    id: 'email',
    label: 'Email',
    sortable: true,
  },
  {
    id: 'status',
    label: 'Status',
    sortable: true,
  },
  {
    id: 'joinDate',
    label: 'Join Date',
    sortable: true,
    format: (value) => new Date(value).toLocaleDateString(),
  },
  {
    id: 'actions',
    label: 'Actions',
    sortable: false,
    render: (member) => (
      <Box>
        <IconButton onClick={() => handleView(member)}>
          <ViewIcon />
        </IconButton>
        <IconButton onClick={() => handleEdit(member)}>
          <EditIcon />
        </IconButton>
        <IconButton onClick={() => handleDelete(member)}>
          <DeleteIcon />
        </IconButton>
      </Box>
    ),
  },
];
;

const handleEdit = (member) => {

```

```

    // Edit logic
};

const handleDelete = (member) => {
    // Delete logic
};

const handleView = (member) => {
    // View logic
};

const handleFilterChange = (newFilters) => {
    setFilters(newFilters);
    // Refetch data with new filters
};

const handlePageChange = (newPage, newRowsPerPage) => {
    setPagination(prev => ({
        ...prev,
        page: newPage,
        rowsPerPage: newRowsPerPage,
    }));
    // Refetch data with new pagination
};

return (
    <Box>
        <Box sx={{ mb: 2, display: 'flex', justifyContent: 'space-between' }}>
            <Typography variant="h6">Members</Typography>
            <ExportData
                data={members}
                columns={columns}
                filename="members"
                formats={['csv', 'excel']}
            />
        </Box>

        <TableFilter
            filters={filterConfig}
            values={filters}
            onChange={handleFilterChange}
        />

        <Table
            data={members}
            columns={columns}
            loading={loading}
            pagination={pagination}
            onPageChange={handlePageChange}
            onEdit={handleEdit}
            onDelete={handleDelete}
        />

```

```
        </Box>
    );
};

export default MembersTable;
```

Component Development Guidelines

1. Component Structure

- Use functional components with hooks
- Implement PropTypes for type checking
- Include default props where appropriate
- Follow consistent naming conventions

2. State Management

- Use local state for component-specific data
- Use Redux for global state
- Use RTK Query for server state

3. Styling

- Use Material-UI theme system
- Implement responsive design
- Follow consistent spacing and typography

4. Accessibility

- Include proper ARIA labels
- Ensure keyboard navigation
- Maintain color contrast ratios

5. Performance

- Use React.memo for expensive components
- Implement proper key props for lists
- Avoid unnecessary re-renders

Version: 1.0.0

Last Updated: January 2025

Maintainer: ASMC Development Team