

System Interfaces Documentation

Generated: November 19, 2025
Total Interfaces: 45+
Architecture: RESTful API + Livewire UI Components

Interface Overview

The HRM Laravel Base system provides a comprehensive set of interfaces spanning RESTful API, web UI components, portal interfaces, and system integrations. The system follows API-first design principles with Livewire-powered reactive interfaces.

1. RESTful API Interfaces

1.1 Authentication & Security APIs

POST /api/login

Authenticates users and returns access tokens.

Request:

```
{
  "email": "user@example.com",
  "password": "password",
  "remember": false
}
```

Response:

```
{
  "user": {
    "id": 1,
    "name": "John Doe",
    "email": "user@example.com",
    "current_organization_id": 1
  },
  "token": "1|abc123token...",
  "abilities": []
}
```

POST /api/logout

Revokes authentication tokens.

Headers: `Authorization: Bearer {token}`

Response: `204 No Content`

POST /api/register

Registers new users (if enabled).

Request:

```
{
  "name": "John Doe",
  "email": "john@example.com",
  "password": "password",
  "password_confirmation": "password"
}
```

1.2 Organization Management APIs

GET /api/organizations

Lists user's organizations with pagination.

Response:

```
{
  "data": [
    {
      "id": 1,
      "name": "Acme Corp",
      "description": "Pharmaceutical company",
      "is_active": true,
      "created_at": "2025-01-01T00:00:00Z",
      "members_count": 15,
      "stores_count": 3
    }
  ],
  "links": {...},
  "meta": {...}
}
```

POST /api/organizations

Creates new organization.

Request:

```
{
  "name": "New Company",
  "description": "Company description"
}
```

GET /api/organizations/{id}

Retrieves organization details with relationships.

Response:

```
{
  "id": 1,
  "name": "Acme Corp",
  "description": "Pharmaceutical company",
  "is_active": true,
  "members": [...],
  "units": [...],
  "stores": [...],
  "created_at": "2025-01-01T00:00:00Z"
}
```

1.3 HR Management APIs

GET /api/employees

Lists employees with filtering and pagination.

Query Parameters:

- `search` - Search by name, email
- `department` - Filter by department
- `position` - Filter by position
- `status` - Filter by active/inactive

Response:

```
{
  "data": [
    {
      "id": 1,
      "first_name": "John",
      "last_name": "Doe",
      "email": "john@example.com",
      "position": {
        "title": "Software Developer",
        "department": "IT"
      },
      "is_active": true,
      "attendance_today": {
        "punch_in": "09:00",
        "punch_out": null,
        "status": "present"
      }
    }
  ]
}
```

POST /api/employees

Creates new employee record.

Request:

```
{
  "first_name": "Jane",
  "last_name": "Smith",
  "email": "jane@example.com",
  "position_id": 1,
  "shift_id": 1,
  "organization_unit_id": 1,
  "biometric_id": "BIO001",
  "date_of_birth": "1990-01-01",
  "gender": "female",
  "phone": "+1234567890",
  "address": "123 Main St"
}
```

GET /api/positions

Manages job positions and roles.

GET /api/shifts

Manages work shifts and schedules.

1.4 Financial Management APIs

GET /api/accounts

Manages chart of accounts.

Response:

```
{
  "data": [
    {
      "id": 1,
      "code": "1001",
      "name": "Cash Account",
      "type": "asset",
      "balance": 50000.00,
      "created_at": "2025-01-01T00:00:00Z"
    }
  ]
}
```

GET /api/journal-entries

Financial transaction management.

Query Parameters:

- `status` - draft, posted, void
- `date_from` - Filter by date range
- `date_to` - Filter by date range

POST /api/journal-entries

Creates journal entries with validation.

Request:

```
{
  "reference_number": "JE001",
  "entry_date": "2025-01-01",
  "description": "Office supplies purchase",
  "entries": [
    {
      "chart_of_account_id": 1,
      "type": "debit",
      "amount": 1000.00
    },
    {
      "chart_of_account_id": 2,
      "type": "credit",
      "amount": 1000.00
    }
  ]
}
```

GET /api/vouchers

Specialized voucher management.

Endpoints:

- `POST /api/vouchers/sales` - Sales vouchers
- `POST /api/vouchers/purchase` - Purchase vouchers
- `POST /api/vouchers/expense` - Expense vouchers
- `POST /api/vouchers/salary` - Salary vouchers

GET /api/reports/

Financial reporting endpoints.

Endpoints:

- GET /api/reports/trial-balance - Trial balance
- GET /api/reports/balance-sheet - Balance sheet
- GET /api/reports/income-statement - Income statement

Sample Response:

```
{
  "trial_balance": {
    "as_of_date": "2025-01-31",
    "accounts": [...],
    "total_debits": 150000.00,
    "total_credits": 150000.00,
    "is_balanced": true
  }
}
```

1.5 Inventory Management APIs

GET /api/inventory/items

Item catalog management.

Response:

```
{
  "data": [
    {
      "id": 1,
      "name": "Paracetamol 500mg",
      "code": "PAR001",
      "category": "Medicines",
      "unit": "strips",
      "cost_price": 25.50,
      "selling_price": 35.00,
      "current_stock": 150,
      "min_stock_level": 50,
      "stock_status": "adequate",
      "stores": [
        {
          "store_id": 1,
          "quantity": 100,
          "location": "Main Store"
        }
      ]
    }
  ]
}
```

GET /api/inventory/stores

Store management.

GET /api/inventory/transactions

Stock movement tracking.

Transaction Types:

- **IN** - Stock receipt
- **OUT** - Stock issue
- **TRANSFER** - Store transfer
- **ADJUST** - Stock adjustment

POST /api/inventory/transactions

Creates inventory transactions.

Request:


```
{
  "store_id": 1,
  "transaction_type": "IN",
  "reference_number": "GRN001",
  "notes": "Purchase from supplier",
  "items": [
    {
      "item_id": 1,
      "quantity": 100,
      "unit_cost": 25.50,
      "batch_number": "BATCH001",
      "expiry_date": "2026-12-31"
    }
  ]
}
```

2. Web UI Interfaces (Livewire Components)

2.1 Dashboard Interfaces

Dashboard Component

Real-time dashboard with KPIs and quick actions.

Features:

- Organization statistics
- Low stock alerts
- Recent transactions
- Quick action buttons
- Interactive charts

Data Binding:

```
public $stats = [
  'total_employees' => 0,
  'active_employees' => 0,
  'total_stores' => 0,
  'low_stock_items' => 0,
  'today_transactions' => 0
];
```

2.2 Organization Management UI

OrganizationTree Component

Drag-and-drop organizational structure.

Features:

- Hierarchical tree view
- Drag-drop reorganization
- Unit creation/editing
- Member assignment

Events:

- `unitMoved` - Unit repositioned
- `memberAssigned` - User assigned to unit
- `unitCreated` - New unit created

MemberManager Component

Organization member management.

Features:

- Member listing with search
 - Role assignment
 - Invitation system
 - Permission management
-

2.3 HR Management UI

EmployeeManagement Component

Complete employee CRUD interface.

Features:

- Employee listing with filters
- Employee creation wizard
- Profile editing
- Bulk operations
- Export functionality

AttendanceDashboard Component

Attendance tracking and management.

Features:

- Daily attendance view
 - Exception handling
 - Biometric sync
 - Regularization requests
-

2.4 Financial Management UI**ChartOfAccounts Component**

Account management interface.

Features:

- Account listing with hierarchy
- Account creation/editing
- Balance calculations
- Import/export

JournalEntry Component

Double-entry journal interface.

Features:

- Dynamic line items
- Auto-balance validation
- Account search
- Posting workflow

VoucherSystem Component

Specialized voucher creation.

Voucher Types:

- Sales voucher with customer details
 - Purchase voucher with vendor info
 - Expense voucher with categories
 - Salary voucher with payroll integration
-

2.5 Inventory Management UI

ItemManagement Component

Item catalog management.

Features:

- Item listing with search
- Item creation wizard
- Stock level indicators
- Category management

StoreManagement Component

Multi-store interface.

Features:

- Store listing
- Stock transfer between stores
- Store-specific inventory
- Location management

TransactionWizard Component

Multi-step transaction creation.

Steps:

1. Select transaction type
 2. Choose store/items
 3. Enter quantities
 4. Review and finalize
-

3. Portal Interfaces

3.1 Employee Portal

EmployeeDashboard Component

Self-service employee portal.

Features:

- Personal information view
- Attendance summary
- Leave balance
- Payslip access
- Quick actions

LeaveRequest Component

Leave application interface.

Features:

- Leave type selection
- Date picker with balance check
- Reason entry
- Status tracking

PayslipViewer Component

Payslip viewing and download.

Features:

- Payslip listing
 - Detailed breakdown view
 - PDF download
 - Email payslip
-

3.2 Manager Portal

ManagerDashboard Component

Team management interface.

Features:

- Team overview
- Attendance summary
- Leave approvals
- Team performance

TeamAttendance Component

Team attendance management.

Features:

- Team attendance view
 - Exception handling
 - Bulk regularization
 - Export reports
-

4. Mobile & Responsive Interfaces

4.1 Mobile-Optimized Components

MobileAttendance Component

Kiosk-style attendance interface.

Features:

- Large touch targets
- QR code scanning
- Biometric integration
- Offline support

MobileStockCount Component

Mobile stock counting.

Features:

- Barcode scanning
- Offline data collection
- Sync capabilities
- Photo capture

5. Integration Interfaces

5.1 Biometric Integration

BiometricSync Interface

Device integration framework.

Endpoints:

- `POST /api/biometric/sync` - Sync attendance data
- `GET /api/biometric/devices` - Device management
- `POST /api/biometric/register` - Register employee

Data Format:

```
{
  "device_id": "BIO001",
  "records": [
    {
      "employee_id": "EMP001",
      "timestamp": "2025-01-01T09:00:00Z",
      "type": "punch_in",
      "device_serial": "SN123456"
    }
  ]
}
```

5.2 Email Integration

Notification Interface

Email notification system.

Events:

- Leave approval/rejection
- Payslip generation
- Low stock alerts
- System notifications

5.3 File Storage Integration

DocumentManagement Interface

File upload and management.

Features:

- Employee documents
- Transaction attachments
- Report exports
- Image uploads

Supported Formats:

- PDF, DOC, DOCX
 - JPG, PNG, GIF
 - XLS, XLSX, CSV
-

6. API Response Standards

6.1 Success Responses

Standard Format:

```
{
  "success": true,
  "data": {...},
  "message": "Operation completed successfully",
  "meta": {
    "timestamp": "2025-01-01T12:00:00Z",
    "version": "v1"
  }
}
```

Paginated Response:

```
{
  "data": [...],
  "links": {
    "first": "...",
    "last": "...",
    "prev": "...",
    "next": "..."
  },
  "meta": {
    "current_page": 1,
    "from": 1,
    "last_page": 10,
    "per_page": 15,
    "to": 15,
    "total": 150
  }
}
```

6.2 Error Responses

Validation Error:

```
{
  "success": false,
  "message": "Validation failed",
  "errors": {
    "email": ["The email field is required."],
    "password": ["Password must be at least 8 characters."]
  }
}
```

Authorization Error:

```
{
  "success": false,
  "message": "Unauthorized",
  "error": "INSUFFICIENT_PERMISSIONS"
}
```

Not Found Error:

```
{
  "success": false,
  "message": "Resource not found",
  "error": "RESOURCE_NOT_FOUND"
}
```

7. Real-time Features

7.1 Livewire Real-time Updates

Wire:Click Events

Interactive UI updates without page refresh.

Wire:Model Live Binding

Real-time form validation and updates.

Wire:Poll

Periodic data refresh for dashboards.

7.2 WebSocket Events (Future)

Planned real-time features:

- Live attendance updates
 - Real-time stock levels
 - Instant notifications
 - Collaborative editing
-

8. Security & Authentication

8.1 API Authentication

Sanctum Token Authentication

Bearer token-based API access.

Session-based Web Authentication

Laravel session management for web interfaces.

8.2 Authorization

Role-based Access Control

Permission checking at route and model level.

Organization-based Data Isolation

Multi-tenant data separation.

9. Performance & Optimization

9.1 API Optimization

Eager Loading

Preventing N+1 query problems.

Pagination

Efficient large dataset handling.

Caching

Frequently accessed data caching.

9.2 UI Optimization

Lazy Loading

On-demand component loading.

Computed Properties

Efficient reactive calculations.

Wire:Loading States

User feedback during operations.

10. Future Interface Enhancements

10.1 Planned Features

GraphQL API

Alternative query interface for complex data requirements.

Mobile App API

Dedicated mobile application endpoints.

Third-party Integrations

- *Accounting software integration*
- *Payment gateway APIs*
- *Cloud storage services*

Advanced Analytics

- *Custom report builder API*
- *Data visualization endpoints*
- *Predictive analytics interface*

This interface documentation reflects the current system capabilities as of November 19, 2025. system provides comprehensive API coverage with modern reactive UI components.*