Frontend Project Details

Last updated by | Polane Mahloko | Feb 26, 2026 at 4:36 AM GMT+2

Final Details for specification

[Department of Informatics - Proposal 1_Sales_Automation.pdf]

Sales Automation API — Documentation

Base URL (Production): `https://sales-automation-bmdqg9b6a0d3ffem.southafricanorth-01.azurewebsites.net`

Base URL (Local): `http://localhost:5053`

Format: JSON (`Content-Type: application/json`)

Authentication: `Authorization: Bearer <token>` — required on all endpoints except `/auth/login` and `/auth/register`

Table of Contents

- Response Codes

Authentication

POST /api/auth/login

Log in and receive a JWT token.

Request

```
{
  "email": "admin@salesautomation.com",
  "password": "Admin@123"
}
```

Response `200`

```
{
  "token": "<jwt>",
  "userId": "...",
  "email": "admin@salesautomation.com",
  "firstName": "Admin",
  "lastName": "User",
  "roles": ["Admin"],
  "tenantId": "11111111-1111-1111-1111-111111111111",
  "expiresAt": "2026-02-25T16:00:00Z"
}
```

POST /api/auth/register

Create a new user account. Supports three organisation scenarios.

Scenario A — Create a new organisation

The registering user becomes the **Admin** of the new organisation.

```
{
  "email": "alice@acme.com",
  "password": "Pass@123",
  "firstName": "Alice",
  "lastName": "Smith",
  "tenantName": "Acme Corp"
}
```

Scenario B — Join an existing organisation

Requires the organisation's Tenant ID from an existing Admin.

```
{
  "email": "bob@acme.com",
  "password": "Pass@123",
  "firstName": "Bob",
  "lastName": "Jones",
  "tenantId": "63d7bdc1-d2c4-488c-98c7-15c8d0657d58",
  "role": "SalesManager"
}
```

Scenario C — Use the default shared tenant

No organisation specified — falls back to the system default tenant.

```
{
  "email": "carol@test.com",
  "password": "Pass@123",
  "firstName": "Carol",
  "lastName": "White",
  "role": "SalesRep"
}
```

Request fields

Field	Type	Required	Notes
`email`	string	Yes	Must be unique
`password`	string	Yes	Minimum 6 characters
`firstName`	string	Yes	
`lastName`	string	Yes	
`phoneNumber`	string	No	
`tenantName`	string	No	Creates a new organisation; caller becomes `Admin` . Mutually exclusive with `tenantId` .
`tenantId`	guid	No	Joins an existing organisation by ID.
`role`	string	No	Ignored when `tenantName` is provided. Defaults to `SalesRep` . Cannot be `Admin` when joining.

Role options: `SalesRep` · `SalesManager` · `BusinessDevelopmentManager`

Response `201` — same shape as login response.

GET /api/auth/me

Returns the currently authenticated user's profile and claims.

Multi-Tenancy

Every organisation registered in the system gets its own isolated workspace called a **tenant**. All data — clients, opportunities, proposals, contracts, activities — is fully scoped to the authenticated user's tenant.

What this means for API consumers

- You never need to pass a tenant ID in request bodies or query strings — it is determined automatically from your token.
- Attempting to access a resource that belongs to another tenant returns `404 Not Found` , not `403` . This prevents leaking the existence of cross-tenant records.
- All list endpoints only return records from your organisation.

Registration and tenant assignment

Registration scenario	Token contains	Data access
`tenantName` provided	New org's `tenantId`	Only that org's data
`tenantId` provided	Existing org's `tenantId`	Only that org's data
Neither provided	Default tenant `tenantId`	Shared default workspace

Default tenant (demo / testing)

```
Tenant ID:  11111111-1111-1111-1111-111111111111
Admin:      admin@salesautomation.com / Admin@123
```

Token structure

Every issued token carries the following claims used for data scoping and authorization:

Claim	Description
`sub`	User ID
`email`	User email
`firstName` / `lastName`	Display name
`tenantId`	Organisation scope for all requests
`role`	Role for access control decisions
`exp`	Expiry timestamp (Unix)

Roles & Permissions

Role	How to obtain	Access level
`Admin`	Register with `tenantName`, or assigned by another Admin	Full access — all endpoints including delete, approve, reject, assign, user management
`SalesManager`	Register with `"role":"SalesManager"` when joining	Full access including approve/reject proposals, assign opportunities, delete records
`BusinessDevelopmentManager`	Register with `"role":"BusinessDevelopmentManager"`	Create and manage opportunities, proposals, pricing requests, contracts, activities
`SalesRep`	Default when no role specified	Read own data, create activities and pricing requests, update assigned opportunities

> Role restrictions on specific endpoints are noted inline as **Admin only**, **Admin/SalesManager**, etc.

Clients

GET /api/clients

List all clients in your organisation.

Query parameters

Parameter	Type	Description
`pageNumber`	int	Default: `1`
`pageSize`	int	Default: `10`
`searchTerm`	string	Searches name, industry
`industry`	string	Filter by industry
`clientType`	int	See [ClientType]
`isActive`	bool	Filter active/inactive

GET /api/clients/{id}

Get a single client by ID.

GET /api/clients/{id}/stats

Get aggregated statistics for a client — opportunity count, contract count, total contract value.

POST /api/clients

Create a new client.

Request

```
{
  "name": "Acme Corp",
  "industry": "Technology",
  "clientType": 2,
  "website": "https://acme.com",
  "billingAddress": "123 Main St",
  "taxNumber": "1234567890",
  "companySize": "50-100"
}
```

PUT /api/clients/{id}

Update an existing client.

DELETE /api/clients/{id}

Soft-delete a client. **Admin / SalesManager only.** Returns `204` .

Contacts

GET /api/contacts

List contacts with optional filters.

Query parameters: `clientId` , `searchTerm` , `pageNumber` , `pageSize`

GET /api/contacts/by-client/{clientId}

Get all contacts belonging to a specific client.

GET /api/contacts/{id}

Get a single contact.

POST /api/contacts

Create a contact linked to a client.

Request

```
{
  "clientId": "...",
  "firstName": "Jane",
  "lastName": "Doe",
  "email": "jane@acme.com",
  "phoneNumber": "+27 11 123 4567",
  "position": "Procurement Manager",
  "isPrimaryContact": true
}
```

PUT /api/contacts/{id}

Update a contact.

PUT /api/contacts/{id}/set-primary

Mark this contact as the primary contact for their client. Unmarks any previous primary.

DELETE /api/contacts/{id}

Soft-delete a contact. **Admin / SalesManager / BDM only.** Returns `204` .

Opportunities

GET /api/opportunities

List opportunities.

Query parameters: `clientId` , `stage` , `ownerId` , `searchTerm` , `pageNumber` , `pageSize`

GET /api/opportunities/my-opportunities

List opportunities assigned to the current user.

Query parameters: `stage` , `pageNumber` , `pageSize`

GET /api/opportunities/pipeline

Aggregated pipeline view grouped by stage.

Query parameters: `ownerId` (optional — filter to a specific rep)

Response includes: stage breakdown, opportunity counts, total value per stage, weighted pipeline value, conversion rate.

GET /api/opportunities/{id}

Get a single opportunity.

GET /api/opportunities/{id}/stage-history

Get the full stage change history for an opportunity.

POST /api/opportunities

Create an opportunity.

Request

```
{
  "title": "Big Deal",
  "clientId": "...",
  "contactId": "...",
  "estimatedValue": 50000,
  "currency": "ZAR",
  "stage": 1,
  "source": 3,
  "probability": 30,
  "expectedCloseDate": "2026-06-30",
  "description": "Initial engagement from referral"
}
```

PUT /api/opportunities/{id}

Update an opportunity.

PUT /api/opportunities/{id}/stage

Move an opportunity to a new stage. Records history with an optional reason.

Request

```
{
  "stage": 3,
  "reason": "Proposal sent"
}
```

POST /api/opportunities/{id}/assign

Assign an opportunity to a sales rep. **Admin / SalesManager only.**

Request: `{ "userId": "..." }`

DELETE /api/opportunities/{id}

Delete an opportunity. **Admin / SalesManager only.** Returns `204`.

Stage values — see [OpportunityStage]

Source values — see [OpportunitySource]

Proposals

GET /api/proposals

List proposals.

Query parameters: `clientId`, `opportunityId`, `status`, `pageNumber`, `pageSize`

GET /api/proposals/{id}

Get a proposal including all line items and calculated totals.

POST /api/proposals

Create a proposal. Line items can be included on creation.

Request

```json
{
  "opportunityId": "...",
  "clientId": "...",
  "title": "Q1 2026 Proposal",
  "description": "Full implementation package",
  "currency": "ZAR",
  "validUntil": "2026-03-31",
  "lineItems": [
    {
      "productServiceName": "Implementation",
      "description": "Phase 1 setup",
      "quantity": 1,
      "unitPrice": 80000,
      "discount": 5,
      "taxRate": 15
    }
  ]
}
```

Line item total formula: `(Quantity × UnitPrice × (1 − Discount/100)) × (1 + TaxRate/100)`

PUT /api/proposals/{id}

Update a proposal. Only allowed when status is `Draft`.

POST /api/proposals/{id}/line-items

Add a line item to an existing draft proposal.

PUT /api/proposals/{id}/line-items/{lineItemId}

Update a line item on a draft proposal.

DELETE /api/proposals/{id}/line-items/{lineItemId}

Remove a line item from a draft proposal.

PUT /api/proposals/{id}/submit

Submit a draft proposal for approval. Status changes to `Submitted`.

PUT /api/proposals/{id}/approve

Approve a submitted proposal. **Admin / SalesManager only.** Status changes to `Approved`.

PUT /api/proposals/{id}/reject

Reject a submitted proposal. **Admin / SalesManager only.** Status changes to `Rejected`.

Request: `{ "reason": "Pricing too high, revise and resubmit" }`

DELETE /api/proposals/{id}

Delete a draft proposal. **Admin / SalesManager only.** Returns `204`.

Status values — see [ProposalStatus]

Pricing Requests

GET /api/pricingrequests

List pricing requests.

Query parameters: `status`, `priority`, `assignedToId`, `pageNumber`, `pageSize`

GET /api/pricingrequests/{id}

Get a single pricing request.

GET /api/pricingrequests/pending

List unassigned pricing requests. **Admin / SalesManager only.**

GET /api/pricingrequests/my-requests

List pricing requests assigned to the current user.

POST /api/pricingrequests

Create a pricing request.

Request

```
{
  "title": "Custom Pricing for Client X",
  "description": "Requires volume discount analysis",
  "clientId": "...",
  "opportunityId": "...",
  "requestedById": "...",
  "priority": 3,
  "requiredByDate": "2026-03-15"
}
```

PUT /api/pricingrequests/{id}

Update a pricing request.

POST /api/pricingrequests/{id}/assign

Assign a pricing request to a user. **Admin / SalesManager only.** Status changes to `InProgress`.

Request: `{ "userId": "..." }`

PUT /api/pricingrequests/{id}/complete

Mark a pricing request as complete. Status changes to `Completed`.

Status values — see PricingRequestStatus

Priority values — see Priority

Contracts

GET /api/contracts

List contracts.

Query parameters: `clientId`, `status`, `pageNumber`, `pageSize`

GET /api/contracts/{id}

Get a single contract. Response includes computed fields `isExpiringSoon` and `daysUntilExpiry`.

GET /api/contracts/expiring

Get contracts expiring within a given number of days.

Query parameters: `daysUntilExpiry` (default: `90`)

GET /api/contracts/client/{clientId}

Get all contracts for a specific client.

POST /api/contracts

Create a contract.

Request

```
{
  "clientId": "...",
  "opportunityId": "...",
  "proposalId": "...",
  "title": "Annual SLA Agreement",
  "contractValue": 120000,
  "currency": "ZAR",
  "startDate": "2026-01-01",
  "endDate": "2026-12-31",
  "ownerId": "...",
  "renewalNoticePeriod": 90,
  "autoRenew": false,
  "terms": "Standard T&Cs apply"
}
```

PUT /api/contracts/{id}

Update a contract. Only allowed when status is `Draft` or `Active` .

PUT /api/contracts/{id}/activate

Activate a draft contract. **Admin / SalesManager only.** Status changes to `Active` .

PUT /api/contracts/{id}/cancel

Cancel a contract. **Admin / SalesManager only.** Status changes to `Cancelled`.

DELETE /api/contracts/{id}

Delete a contract. **Admin only.** Returns `204`.

POST /api/contracts/{contractId}/renewals

Create a renewal record for an active contract.

Request

```
{
  "proposedStartDate": "2027-01-01",
  "proposedEndDate": "2027-12-31",
  "proposedValue": 132000,
  "notes": "Annual CPI adjustment of 8%"
}
```

PUT /api/contracts/renewals/{renewalId}/complete

Complete a renewal. **Admin / SalesManager only.** Sets contract status to `Renewed`.

---

isExpiringSoon: `true` when status is `Active` AND `daysUntilExpiry` ≤ `renewalNoticePeriod`

Status values — see [ContractStatus]

Activities

GET /api/activities

List activities.

Query parameters: `assignedToId`, `type`, `status`, `relatedToType`, `relatedToId`, `pageNumber`, `pageSize`

GET /api/activities/{id}

Get a single activity.

GET /api/activities/my-activities

List activities assigned to the current user.

Query parameters: `status` , `pageNumber` , `pageSize`

GET /api/activities/upcoming

Activities due within the next N days.

Query parameters: `daysAhead` (default: `7`)

GET /api/activities/overdue

Activities that are `Scheduled` but past their due date.

POST /api/activities

Create an activity. Can be linked to any entity.

Request

```json
{
  "type": 2,
  "subject": "Discovery call with procurement",
  "description": "Initial needs assessment",
  "priority": 2,
  "dueDate": "2026-03-01T10:00:00",
  "assignedToId": "...",
  "relatedToType": 2,
  "relatedToId": "...",
  "duration": 60,
  "location": "Microsoft Teams"
}
```

PUT /api/activities/{id}

Update a scheduled activity.

PUT /api/activities/{id}/complete

Mark an activity as complete.

Request: `{ "outcome": "Client confirmed interest. Follow-up scheduled." }`

PUT /api/activities/{id}/cancel

Cancel a scheduled activity.

DELETE /api/activities/{id}

Delete an activity. **Admin / SalesManager only.** Returns `204` .

Type values — see [ActivityType]

Status values — see [ActivityStatus]

Priority values — see [Priority]

relatedToType values — see [RelatedToType]

Documents

GET /api/documents

List documents.

Query parameters: `relatedToType` , `relatedToId` , `category` , `pageNumber` , `pageSize`

GET /api/documents/{id}

Get document metadata.

POST /api/documents/upload

Upload a document. Uses `multipart/form-data` .

Form fields

Field	Type	Required	Notes
`file`	file	Yes	Max 50 MB
`documentCategory`	int	No	See DocumentCategory
`relatedToType`	int	No	Entity type to attach to
`relatedToId`	guid	No	Entity ID to attach to
`description`	string	No	

GET /api/documents/{id}/download

Download the file. Returns the file bytes with the correct `Content-Type` header.

DELETE /api/documents/{id}

Soft-delete a document. **Admin / SalesManager only.** Returns `204`.

Category values — see DocumentCategory

Notes

GET /api/notes

List notes, optionally scoped to a specific entity.

Query parameters: `relatedToType`, `relatedToId`, `pageNumber`, `pageSize`

GET /api/notes/{id}

Get a single note.

POST /api/notes

Add a note to any entity.

Request

```json
{
  "content": "Client requested revised pricing before end of Q1.",
  "relatedToType": 2,
  "relatedToId": "...",
  "isPrivate": false
}
```

PUT /api/notes/{id}

Update a note. Only the note creator can update it.

DELETE /api/notes/{id}

Delete a note. Returns `204`.

---

relatedToType values — see [RelatedToType](#)

Dashboard

All dashboard endpoints are automatically scoped to the current user's organisation.

GET /api/dashboard/overview

Full dashboard summary.

Response shape

```
{
  "opportunities": {
    "totalCount": 42,
    "wonCount": 8,
    "winRate": 19.05,
    "pipelineValue": 1250000
  },
  "pipeline": {
    "stages": [...],
    "weightedPipelineValue": 430000
  },
  "activities": {
    "upcomingCount": 12,
    "overdueCount": 3,
    "completedTodayCount": 5
  },
  "contracts": {
    "totalActiveCount": 15,
    "expiringThisMonthCount": 2,
    "totalContractValue": 3800000
  },
  "revenue": {
    "thisMonth": 180000,
    "thisQuarter": 520000,
    "thisYear": 1100000,
    "monthlyTrend": [...]
  }
}
```

GET /api/dashboard/pipeline-metrics

Pipeline stage breakdown with counts and values for charting.

GET /api/dashboard/sales-performance

Top N sales reps by performance. **Admin / SalesManager only.**

Query parameters: `topCount` (default: `5`)

GET /api/dashboard/activities-summary

Activity counts grouped by type and status.

GET /api/dashboard/contracts-expiring

Contracts expiring within N days.

Query parameters: `days` (default: `30`)

Reports

Admin / SalesManager only.

GET /api/reports/opportunities

Filtered opportunities report.

Query parameters: `startDate`, `endDate`, `stage`, `ownerId`

GET /api/reports/sales-by-period

Revenue grouped by a time period.

Query parameters: `startDate`, `endDate`, `groupBy` (`month` or `week`)

Reference: Enums

ClientType

| Value | Label |

|---|---|
`1`	Government
`2`	Private
`3`	Partner

OpportunityStage

Value	Label
1	Lead
2	Qualified
3	Proposal
4	Negotiation
5	Closed Won
6	Closed Lost

OpportunitySource

Value	Label
1	Inbound
2	Outbound
3	Referral
4	Partner
5	RFP

ProposalStatus

Value	Label	Transitions to
1	Draft	Submitted
2	Submitted	Approved, Rejected
3	Rejected	—
4	Approved	—

PricingRequestStatus

Value	Label
1	Pending
2	In Progress
3	Completed

ContractStatus

Value	Label
1	Draft
2	Active
3	Expired
4	Renewed
5	Cancelled

ActivityType

Value	Label
1	Meeting
2	Call
3	Email
4	Task
5	Presentation
6	Other

ActivityStatus

Value	Label
1	Scheduled
2	Completed
3	Cancelled

Priority

Value	Label
1	Low
2	Medium
3	High
4	Urgent

DocumentCategory

Value	Label
1	Contract
2	Proposal
3	Presentation
4	Quote
5	Report
6	Other

RelatedToType

Value	Label	Used in
1	Client	Activities, Documents, Notes
2	Opportunity	Activities, Documents, Notes
3	Proposal	Activities, Documents, Notes
4	Contract	Activities, Documents, Notes
5	Activity	Notes

Typical Workflows

Onboarding a new organisation

```
1. POST /api/auth/register  { "tenantName": "My Company" } ........ → Admin user created
2. POST /api/auth/register  { "tenantId": "...", "role": "SalesManager" }
3. POST /api/auth/register  { "tenantId": "...", "role": "SalesRep" }
```

Full sales cycle

```
1.  POST /api/auth/login
2.  POST /api/clients ..................... → create client
3.  POST /api/contacts .................... → add primary contact
4.  POST /api/opportunities ............... → open opportunity (stage: Lead)
5.  PUT  /api/opportunities/{id}/stage .... → advance stages
6.  POST /api/pricingrequests ............. → request pricing
7.  POST /api/pricingrequests/{id}/assign . → assign to team member
8.  POST /api/proposals ................... → create proposal with line items
9.  PUT  /api/proposals/{id}/submit ....... → submit for approval
10. PUT  /api/proposals/{id}/approve ...... → approve (Admin/SalesManager)
11. POST /api/contracts ................... → create contract
12. PUT  /api/contracts/{id}/activate ..... → activate contract
13. GET  /api/dashboard/overview .......... → review metrics
```

Contract renewal

```
1. GET  /api/contracts/expiring?daysUntilExpiry=90   → identify expiring contracts
2. POST /api/contracts/{id}/renewals                  → create renewal record
3. PUT  /api/contracts/renewals/{renewalId}/complete  → complete renewal
```

Response Codes

Code	Meaning
`200`	Success
`201`	Resource created
`204`	Deleted (no content returned)
`400`	Validation error or invalid request (e.g. invalid role, tenant not found)
`401`	Missing or expired token
`403`	Insufficient role for this action
`404`	Resource not found — also returned when accessing another tenant's resource

---

Sales Automation API — v2.0# Sales Automation API — Documentation

Base URL (Production): `https://sales-automation-bmdqg9b6a0d3ffem.southafricanorth-01.azurewebsites.net`

Base URL (Local): `http://localhost:5053`

Format: JSON (`Content-Type: application/json`)

Authentication: `Authorization: Bearer <token>` — required on all endpoints except `/auth/login` and `/auth/register`

Table of Contents

Authentication

POST /api/auth/login

Log in and receive a JWT token.

Request

```
{
  "email": "admin@salesautomation.com",
  "password": "Admin@123"
}
```

Response `200`

```json
{
  "token": "<jwt>",
  "userId": "...",
  "email": "admin@salesautomation.com",
  "firstName": "Admin",
  "lastName": "User",
  "roles": ["Admin"],
  "tenantId": "11111111-1111-1111-1111-111111111111",
  "expiresAt": "2026-02-25T16:00:00Z"
}
```

POST /api/auth/register

Create a new user account. Supports three organisation scenarios.

Scenario A — Create a new organisation

The registering user becomes the **Admin** of the new organisation.

```json
{
  "email": "alice@acme.com",
  "password": "Pass@123",
  "firstName": "Alice",
  "lastName": "Smith",
  "tenantName": "Acme Corp"
}
```

Scenario B — Join an existing organisation

Requires the organisation's Tenant ID from an existing Admin.

```
{
  "email": "bob@acme.com",
  "password": "Pass@123",
  "firstName": "Bob",
  "lastName": "Jones",
  "tenantId": "63d7bdc1-d2c4-488c-98c7-15c8d0657d58",
  "role": "SalesManager"
}
```

Scenario C — Use the default shared tenant

No organisation specified — falls back to the system default tenant.

```
{
  "email": "carol@test.com",
  "password": "Pass@123",
  "firstName": "Carol",
  "lastName": "White",
  "role": "SalesRep"
}
```

Request fields

| Field | Type | Required | Notes |

|---|---|---|---|
`email`	string	Yes	Must be unique
`password`	string	Yes	Minimum 6 characters
`firstName`	string	Yes	
`lastName`	string	Yes	
`phoneNumber`	string	No	
`tenantName`	string	No	Creates a new organisation; caller becomes `Admin`. Mutually exclusive with `tenantId`.
`tenantId`	guid	No	Joins an existing organisation by ID.
`role`	string	No	Ignored when `tenantName` is provided. Defaults to `SalesRep`. Cannot be `Admin` when joining.

Role options: `SalesRep` · `SalesManager` · `BusinessDevelopmentManager`

Response `201` — same shape as login response.

GET /api/auth/me

Returns the currently authenticated user's profile and claims.

Multi-Tenancy

Every organisation registered in the system gets its own isolated workspace called a **tenant**. All data — clients, opportunities, proposals, contracts, activities — is fully scoped to the authenticated user's tenant.

What this means for API consumers

- You never need to pass a tenant ID in request bodies or query strings — it is determined automatically from your token.
- Attempting to access a resource that belongs to another tenant returns `404 Not Found`, not `403`. This prevents leaking the existence of cross-tenant records.
- All list endpoints only return records from your organisation.

Registration and tenant assignment

Registration scenario	Token contains	Data access
`tenantName` provided	New org's `tenantId`	Only that org's data
`tenantId` provided	Existing org's `tenantId`	Only that org's data
Neither provided	Default tenant `tenantId`	Shared default workspace

Default tenant (demo / testing)

```
Tenant ID:  11111111-1111-1111-1111-111111111111
Admin:      admin@salesautomation.com / Admin@123
```

Token structure

Every issued token carries the following claims used for data scoping and authorization:

Claim	Description
`sub`	User ID
`email`	User email
`firstName` / `lastName`	Display name
`tenantId`	Organisation scope for all requests
`role`	Role for access control decisions
`exp`	Expiry timestamp (Unix)

Roles & Permissions

Role	How to obtain	Access level
`Admin`	Register with `tenantName` , or assigned by another Admin	Full access — all endpoints including delete, approve, reject, assign, user management
`SalesManager`	Register with `"role":"SalesManager"` when joining	Full access including approve/reject proposals, assign opportunities, delete records
`BusinessDevelopmentManager`	Register with `"role":"BusinessDevelopmentManager"`	Create and manage opportunities, proposals, pricing requests, contracts, activities
`SalesRep`	Default when no role specified	Read own data, create activities and pricing requests, update assigned opportunities

> Role restrictions on specific endpoints are noted inline as **Admin only**, **Admin/SalesManager**, etc.

Clients

GET /api/clients

List all clients in your organisation.

Query parameters

Parameter	Type	Description
`pageNumber`	int	Default: `1`
`pageSize`	int	Default: `10`
`searchTerm`	string	Searches name, industry
`industry`	string	Filter by industry
`clientType`	int	See [ClientType]
`isActive`	bool	Filter active/inactive

GET /api/clients/{id}

Get a single client by ID.

GET /api/clients/{id}/stats

Get aggregated statistics for a client — opportunity count, contract count, total contract value.

POST /api/clients

Create a new client.

Request

```
{
  "name": "Acme Corp",
  "industry": "Technology",
  "clientType": 2,
  "website": "https://acme.com",
  "billingAddress": "123 Main St",
  "taxNumber": "1234567890",
  "companySize": "50-100"
}
```

PUT /api/clients/{id}

Update an existing client.

DELETE /api/clients/{id}

Soft-delete a client. **Admin / SalesManager only.** Returns `204`.

Contacts

GET /api/contacts

List contacts with optional filters.

Query parameters: `clientId`, `searchTerm`, `pageNumber`, `pageSize`

GET /api/contacts/by-client/{clientId}

Get all contacts belonging to a specific client.

GET /api/contacts/{id}

Get a single contact.

POST /api/contacts

Create a contact linked to a client.

Request

```
{
  "clientId": "...",
  "firstName": "Jane",
  "lastName": "Doe",
  "email": "jane@acme.com",
  "phoneNumber": "+27 11 123 4567",
  "position": "Procurement Manager",
  "isPrimaryContact": true
}
```

PUT /api/contacts/{id}

Update a contact.

PUT /api/contacts/{id}/set-primary

Mark this contact as the primary contact for their client. Unmarks any previous primary.

DELETE /api/contacts/{id}

Soft-delete a contact. **Admin / SalesManager / BDM only.** Returns `204` .

Opportunities

GET /api/opportunities

List opportunities.

Query parameters: `clientId`, `stage`, `ownerId`, `searchTerm`, `pageNumber`, `pageSize`

GET /api/opportunities/my-opportunities

List opportunities assigned to the current user.

Query parameters: `stage`, `pageNumber`, `pageSize`

GET /api/opportunities/pipeline

Aggregated pipeline view grouped by stage.

Query parameters: `ownerId` (optional — filter to a specific rep)

Response includes: stage breakdown, opportunity counts, total value per stage, weighted pipeline value, conversion rate.

GET /api/opportunities/{id}

Get a single opportunity.

GET /api/opportunities/{id}/stage-history

Get the full stage change history for an opportunity.

POST /api/opportunities

Create an opportunity.

Request

```
{
  "title": "Big Deal",
  "clientId": "...",
  "contactId": "...",
  "estimatedValue": 50000,
  "currency": "ZAR",
  "stage": 1,
  "source": 3,
  "probability": 30,
  "expectedCloseDate": "2026-06-30",
  "description": "Initial engagement from referral"
}
```

PUT /api/opportunities/{id}

Update an opportunity.

PUT /api/opportunities/{id}/stage

Move an opportunity to a new stage. Records history with an optional reason.

Request

```
{
  "stage": 3,
  "reason": "Proposal sent"
}
```

POST /api/opportunities/{id}/assign

Assign an opportunity to a sales rep. **Admin / SalesManager only.**

Request: `{ "userId": "..." }`

DELETE /api/opportunities/{id}

Delete an opportunity. **Admin / SalesManager only.** Returns `204`.

Stage values — see [OpportunityStage]

Source values — see [OpportunitySource]

Proposals

GET /api/proposals

List proposals.

Query parameters: `clientId`, `opportunityId`, `status`, `pageNumber`, `pageSize`

GET /api/proposals/{id}

Get a proposal including all line items and calculated totals.

POST /api/proposals

Create a proposal. Line items can be included on creation.

Request

```json
{
  "opportunityId": "...",
  "clientId": "...",
  "title": "Q1 2026 Proposal",
  "description": "Full implementation package",
  "currency": "ZAR",
  "validUntil": "2026-03-31",
  "lineItems": [
    {
      "productServiceName": "Implementation",
      "description": "Phase 1 setup",
      "quantity": 1,
      "unitPrice": 80000,
      "discount": 5,
      "taxRate": 15
    }
  ]
}
```


Line item total formula: `(Quantity × UnitPrice × (1 − Discount/100)) × (1 + TaxRate/100)`

PUT /api/proposals/{id}

Update a proposal. Only allowed when status is `Draft`.

POST /api/proposals/{id}/line-items

Add a line item to an existing draft proposal.

PUT /api/proposals/{id}/line-items/{lineItemId}

Update a line item on a draft proposal.

DELETE /api/proposals/{id}/line-items/{lineItemId}

Remove a line item from a draft proposal.

PUT /api/proposals/{id}/submit

Submit a draft proposal for approval. Status changes to `Submitted`.

PUT /api/proposals/{id}/approve

Approve a submitted proposal. **Admin / SalesManager only.** Status changes to `Approved`.

PUT /api/proposals/{id}/reject

Reject a submitted proposal. **Admin / SalesManager only.** Status changes to `Rejected`.

Request: `{ "reason": "Pricing too high, revise and resubmit" }`

DELETE /api/proposals/{id}

Delete a draft proposal. **Admin / SalesManager only.** Returns `204`.

Status values — see [ProposalStatus]

Pricing Requests

GET /api/pricingrequests

List pricing requests.

Query parameters: `status`, `priority`, `assignedToId`, `pageNumber`, `pageSize`

GET /api/pricingrequests/{id}

Get a single pricing request.

GET /api/pricingrequests/pending

List unassigned pricing requests. **Admin / SalesManager only.**

GET /api/pricingrequests/my-requests

List pricing requests assigned to the current user.

POST /api/pricingrequests

Create a pricing request.

Request

```json
{
  "title": "Custom Pricing for Client X",
  "description": "Requires volume discount analysis",
  "clientId": "...",
  "opportunityId": "...",
  "requestedById": "...",
  "priority": 3,
  "requiredByDate": "2026-03-15"
}
```

PUT /api/pricingrequests/{id}

Update a pricing request.

POST /api/pricingrequests/{id}/assign

Assign a pricing request to a user. **Admin / SalesManager only.** Status changes to `InProgress`.

Request: `{ "userId": "..." }`

PUT /api/pricingrequests/{id}/complete

Mark a pricing request as complete. Status changes to `Completed`.

Status values — see PricingRequestStatus

Priority values — see Priority

Contracts

GET /api/contracts

List contracts.

Query parameters: `clientId`, `status`, `pageNumber`, `pageSize`

GET /api/contracts/{id}

Get a single contract. Response includes computed fields `isExpiringSoon` and `daysUntilExpiry`.

GET /api/contracts/expiring

Get contracts expiring within a given number of days.

Query parameters: `daysUntilExpiry` (default: `90`)

GET /api/contracts/client/{clientId}

Get all contracts for a specific client.

POST /api/contracts

Create a contract.

Request

```
{
  "clientId": "...",
  "opportunityId": "...",
  "proposalId": "...",
  "title": "Annual SLA Agreement",
  "contractValue": 120000,
  "currency": "ZAR",
  "startDate": "2026-01-01",
  "endDate": "2026-12-31",
  "ownerId": "...",
  "renewalNoticePeriod": 90,
  "autoRenew": false,
  "terms": "Standard T&Cs apply"
}
```

PUT /api/contracts/{id}

Update a contract. Only allowed when status is `Draft` or `Active` .

PUT /api/contracts/{id}/activate

Activate a draft contract. **Admin / SalesManager only.** Status changes to `Active` .

PUT /api/contracts/{id}/cancel

Cancel a contract. **Admin / SalesManager only.** Status changes to `Cancelled`.

DELETE /api/contracts/{id}

Delete a contract. **Admin only.** Returns `204`.

POST /api/contracts/{contractId}/renewals

Create a renewal record for an active contract.

Request

```
{
  "proposedStartDate": "2027-01-01",
  "proposedEndDate": "2027-12-31",
  "proposedValue": 132000,
  "notes": "Annual CPI adjustment of 8%"
}
```

PUT /api/contracts/renewals/{renewalId}/complete

Complete a renewal. **Admin / SalesManager only.** Sets contract status to `Renewed`.

---

isExpiringSoon: `true` when status is `Active` AND `daysUntilExpiry` ≤ `renewalNoticePeriod`

Status values — see ContractStatus

Activities

GET /api/activities

List activities.

Query parameters: `assignedToId`, `type`, `status`, `relatedToType`, `relatedToId`, `pageNumber`, `pageSize`

GET /api/activities/{id}

Get a single activity.

GET /api/activities/my-activities

List activities assigned to the current user.

Query parameters: `status` , `pageNumber` , `pageSize`

GET /api/activities/upcoming

Activities due within the next N days.

Query parameters: `daysAhead` (default: `7`)

GET /api/activities/overdue

Activities that are `Scheduled` but past their due date.

POST /api/activities

Create an activity. Can be linked to any entity.

Request

```
{
  "type": 2,
  "subject": "Discovery call with procurement",
  "description": "Initial needs assessment",
  "priority": 2,
  "dueDate": "2026-03-01T10:00:00",
  "assignedToId": "...",
  "relatedToType": 2,
  "relatedToId": "...",
  "duration": 60,
  "location": "Microsoft Teams"
}
```

PUT /api/activities/{id}

Update a scheduled activity.

PUT /api/activities/{id}/complete

Mark an activity as complete.

Request: `{ "outcome": "Client confirmed interest. Follow-up scheduled." }`

PUT /api/activities/{id}/cancel

Cancel a scheduled activity.

DELETE /api/activities/{id}

Delete an activity. **Admin / SalesManager only.** Returns `204`.

Type values — see [ActivityType]

Status values — see [ActivityStatus]

Priority values — see [Priority]

relatedToType values — see [RelatedToType]

Documents

GET /api/documents

List documents.

Query parameters: `relatedToType`, `relatedToId`, `category`, `pageNumber`, `pageSize`

GET /api/documents/{id}

Get document metadata.

POST /api/documents/upload

Upload a document. Uses `multipart/form-data`.

Form fields

Field	Type	Required	Notes
`file`	file	Yes	Max 50 MB
`documentCategory`	int	No	See [DocumentCategory]
`relatedToType`	int	No	Entity type to attach to
`relatedToId`	guid	No	Entity ID to attach to
`description`	string	No	

GET /api/documents/{id}/download

Download the file. Returns the file bytes with the correct `Content-Type` header.

DELETE /api/documents/{id}

Soft-delete a document. **Admin / SalesManager only.** Returns `204`.

Category values — see [DocumentCategory]

Notes

GET /api/notes

List notes, optionally scoped to a specific entity.

Query parameters: `relatedToType`, `relatedToId`, `pageNumber`, `pageSize`

GET /api/notes/{id}

Get a single note.

POST /api/notes

Add a note to any entity.

Request

```json
{
  "content": "Client requested revised pricing before end of Q1.",
  "relatedToType": 2,
  "relatedToId": "...",
  "isPrivate": false
}
```

PUT /api/notes/{id}

Update a note. Only the note creator can update it.

DELETE /api/notes/{id}

Delete a note. Returns `204`.

---

relatedToType values — see [RelatedToType](#)

Dashboard

All dashboard endpoints are automatically scoped to the current user's organisation.

GET /api/dashboard/overview

Full dashboard summary.

Response shape

```json
{
  "opportunities": {
    "totalCount": 42,
    "wonCount": 8,
    "winRate": 19.05,
    "pipelineValue": 1250000
  },
  "pipeline": {
    "stages": [...],
    "weightedPipelineValue": 430000
  },
  "activities": {
    "upcomingCount": 12,
    "overdueCount": 3,
    "completedTodayCount": 5
  },
  "contracts": {
    "totalActiveCount": 15,
    "expiringThisMonthCount": 2,
    "totalContractValue": 3800000
  },
  "revenue": {
    "thisMonth": 180000,
    "thisQuarter": 520000,
    "thisYear": 1100000,
    "monthlyTrend": [...]
  }
}
```

GET /api/dashboard/pipeline-metrics

Pipeline stage breakdown with counts and values for charting.

GET /api/dashboard/sales-performance

Top N sales reps by performance. **Admin / SalesManager only.**

Query parameters: `topCount` (default: `5`)

GET /api/dashboard/activities-summary

Activity counts grouped by type and status.

GET /api/dashboard/contracts-expiring

Contracts expiring within N days.

Query parameters: `days` (default: `30`)

Reports

Admin / SalesManager only.

GET /api/reports/opportunities

Filtered opportunities report.

Query parameters: `startDate`, `endDate`, `stage`, `ownerId`

GET /api/reports/sales-by-period

Revenue grouped by a time period.

Query parameters: `startDate`, `endDate`, `groupBy` (`month` or `week`)

Reference: Enums

ClientType

Value	Label
`1`	Government
`2`	Private
`3`	Partner

OpportunityStage

Value	Label
1	Lead
2	Qualified
3	Proposal
4	Negotiation
5	Closed Won
6	Closed Lost

OpportunitySource

Value	Label
1	Inbound
2	Outbound
3	Referral
4	Partner
5	RFP

ProposalStatus

Value	Label	Transitions to
1	Draft	Submitted
2	Submitted	Approved, Rejected
3	Rejected	—
4	Approved	—

PricingRequestStatus

Value	Label
1	Pending
2	In Progress
3	Completed

ContractStatus

Value	Label
1	Draft
2	Active
3	Expired
4	Renewed
5	Cancelled

ActivityType

Value	Label
1	Meeting
2	Call
3	Email
4	Task
5	Presentation
6	Other

ActivityStatus

Value	Label
1	Scheduled
2	Completed
3	Cancelled

Priority

Value	Label
1	Low
2	Medium
3	High
4	Urgent

DocumentCategory

Value	Label
1	Contract
2	Proposal
3	Presentation
4	Quote
5	Report
6	Other

RelatedToType

Value	Label	Used in
1	Client	Activities, Documents, Notes
2	Opportunity	Activities, Documents, Notes
3	Proposal	Activities, Documents, Notes
4	Contract	Activities, Documents, Notes
5	Activity	Notes

Typical Workflows

Onboarding a new organisation

```
1. POST /api/auth/register  { "tenantName": "My Company" } ········ → Admin user created
2. POST /api/auth/register  { "tenantId": "...", "role": "SalesManager" }
3. POST /api/auth/register  { "tenantId": "...", "role": "SalesRep" }
```

Full sales cycle

```
1.  POST /api/auth/login
2.  POST /api/clients                       → create client
3.  POST /api/contacts                      → add primary contact
4.  POST /api/opportunities                 → open opportunity (stage: Lead)
5.  PUT  /api/opportunities/{id}/stage      → advance stages
6.  POST /api/pricingrequests               → request pricing
7.  POST /api/pricingrequests/{id}/assign   → assign to team member
8.  POST /api/proposals                     → create proposal with line items
9.  PUT  /api/proposals/{id}/submit         → submit for approval
10. PUT  /api/proposals/{id}/approve        → approve (Admin/SalesManager)
11. POST /api/contracts                     → create contract
12. PUT  /api/contracts/{id}/activate       → activate contract
13. GET  /api/dashboard/overview            → review metrics
```

Contract renewal

```
1. GET  /api/contracts/expiring?daysUntilExpiry=90   → identify expiring contracts
2. POST /api/contracts/{id}/renewals                  → create renewal record
3. PUT  /api/contracts/renewals/{renewalId}/complete  → complete renewal
```

Response Codes

Code	Meaning
`200`	Success
`201`	Resource created
`204`	Deleted (no content returned)
`400`	Validation error or invalid request (e.g. invalid role, tenant not found)
`401`	Missing or expired token
`403`	Insufficient role for this action
`404`	Resource not found — also returned when accessing another tenant's resource

---

Sales Automation API — v2.0---