EMMA API — Swagger Testing Guide (Swagger-only Examples)

Generated: 2025-08-28 00:20:28

Use this guide to test APIs directly in Swagger first, then validate the same flows in the UI. For each POST/PUT endpoint you get Minimal vs Full request bodies. Expected status codes and ProblemDetails samples are included for fast troubleshooting.

# JWT Cheat-Sheet (Dev Token → orgId/userId mapping)

* Open Swagger → Authorize → you’ll paste a Bearer token from the dev-token endpoint.
* Call POST /api/auth/dev-token (supply an email).
* Copy the 'token' value into Swagger’s Authorize dialog as: Bearer <token>.
* Use 'orgId' from the dev-token response in bodies requiring organizationId.
* Use 'userId' for endpoints assigning ownership (e.g., PUT /api/Contact/{id}/assign).
* Org scope guard: your JWT orgId must match entity.OrganizationId.

**Sample dev-token response**

{  
 "token": "eyJhbGciOi...REPLACE...",  
 "orgId": "00000000-0000-0000-0000-000000000000",  
 "userId": "11111111-1111-1111-1111-111111111111",  
 "email": "devtester@example.com"  
}

# Suggested Swagger Test Order

1. POST /api/auth/dev-token (Dev only)
2. POST /api/Organization/invitations/{token}/register (if onboarding)
3. POST /api/auth/verify-email
4. POST /api/Contact
5. GET /api/Contact/{id}
6. GET /api/Contact?orgId={orgId}
7. PUT /api/Contact/{id}/assign
8. POST /api/contacts/{contactId}/interactions
9. GET /api/contacts/{contactId}/interactions
10. POST /api/agent/suggest-followup (Dev stub)
11. GET /api/enums/{type}
12. GET /api/health, /api/health/postgres, /api/health/cosmos

# RFC7807 ProblemDetails — Common Samples

**401 Unauthorized**

{  
 "type": "https://tools.ietf.org/html/rfc7235",  
 "title": "Unauthorized",  
 "status": 401,  
 "detail": "Missing or invalid bearer token."  
}

**403 Forbidden**

{  
 "type": "https://emma.ai/errors/forbidden",  
 "title": "Forbidden",  
 "status": 403,  
 "detail": "You do not have permission to access this resource."  
}

**404 Not Found**

{  
 "type": "https://emma.ai/errors/not-found",  
 "title": "Not Found",  
 "status": 404,  
 "detail": "The requested resource was not found."  
}

**409 Conflict**

{  
 "type": "https://emma.ai/errors/conflict",  
 "title": "Conflict",  
 "status": 409,  
 "detail": "A resource with those unique fields already exists."  
}

**422 Validation Failed**

{  
 "type": "https://emma.ai/errors/validation",  
 "title": "One or more validation errors occurred.",  
 "status": 422,  
 "errors": {  
 "organizationId": ["The organizationId field is required."],  
 "firstName": ["Minimum length is 1."]  
 }  
}

**500 Server Error**

{  
 "type": "about:blank",  
 "title": "An unexpected error occurred",  
 "status": 500,  
 "detail": "TraceId: 00-REPLACE-TRACE"  
}

# Authentication & Account

## POST /api/auth/dev-token — Issue development JWT

**Purpose:** Mint a development token and (optional) org/user IDs for local testing.

**Input:** Body (optional): { orgId?: Guid, userId?: Guid, email?: string }

**Output:** JSON: { token, orgId, userId, email }

**Expected Status:** 200 OK; 400 Bad Request; 500 Server Error

### Minimal (Swagger Body)

{ "email": "devtester@example.com" }

### Full (Swagger Body)

{  
 "orgId": "00000000-0000-0000-0000-000000000000",  
 "userId": "11111111-1111-1111-1111-111111111111",  
 "email": "devtester@example.com"  
}

## POST /api/auth/verify-email — Verify Email

**Purpose:** Validate a verification token; idempotent if already verified.

**Input:** Body: { "token": "..." }

**Output:** 204 No Content on success

**Expected Status:** 204 No Content; 400 Bad Request; 404 Not Found; 409 Conflict

### Minimal (Swagger Body)

{ "token": "REPLACE\_WITH\_VERIFICATION\_TOKEN" }

### Full (Swagger Body)

{ "token": "REPLACE\_WITH\_VERIFICATION\_TOKEN" }

# Organization / Onboarding

## POST /api/Organization/invitations/{token}/register — Register From Invitation

**Purpose:** Register a user from an invitation token; triggers verification flow (dev stub).

**Input:** Path: token; Body: { firstName?, lastName?/fullName?, password? }

**Output:** 204 No Content

**Expected Status:** 204 No Content; 400 Bad Request; 404 Not Found; 409 Conflict

### Minimal (Swagger Body)

{  
 "firstName": "Emma",  
 "lastName": "Tester",  
 "password": "StrongP@ssw0rd!"  
}

### Full (Swagger Body)

{  
 "firstName": "Emma",  
 "lastName": "Tester",  
 "password": "StrongP@ssw0rd!"  
}

# Contacts

## POST /api/Contact — Create Contact

**Purpose:** Create a new contact in your organization.

**Input:** Body: { organizationId: Guid, firstName, lastName, ownerId?, relationshipState?, tags?, customFields? }

**Output:** 201 Created with Contact resource

**Expected Status:** 201 Created; 400 Bad Request; 401 Unauthorized; 403 Forbidden; 409 Conflict; 422 Validation Failed; 500 Server Error

### Minimal (Swagger Body)

{  
 "organizationId": "00000000-0000-0000-0000-000000000000",  
 "firstName": "Ada",  
 "lastName": "Lovelace"  
}

### Full (Swagger Body)

{  
 "organizationId": "00000000-0000-0000-0000-000000000000",  
 "firstName": "Ada",  
 "lastName": "Lovelace",  
 "relationshipState": "Lead",  
 "ownerId": "11111111-1111-1111-1111-111111111111",  
 "tags": ["VIP", "Newsletter"],  
 "customFields": { "PreferredLanguage": "en-CA" }  
}

## GET /api/Contact/{id} — Get Contact by Id

**Purpose:** Retrieve contact by id within org scope.

**Input:** Path: id (Guid)

**Output:** 200 OK with Contact resource

**Expected Status:** 200 OK; 401 Unauthorized; 403 Forbidden; 404 Not Found

## GET /api/Contact?orgId={orgId} — List Contacts

**Purpose:** List contacts for an org with pagination/filters.

**Input:** Query: orgId (Guid), page?, limit?, filters?

**Output:** 200 OK with List<Contact> and paging metadata

**Expected Status:** 200 OK; 401 Unauthorized; 403 Forbidden

## PUT /api/Contact/{id}/assign — Assign Contact

**Purpose:** Assign the contact to a specific user (owner/agent).

**Input:** Path: id; Body: { "userId": Guid }

**Output:** 204 No Content (or 200 OK with updated resource)

**Expected Status:** 204 No Content; 400 Bad Request; 401 Unauthorized; 403 Forbidden; 404 Not Found

### Minimal (Swagger Body)

{ "userId": "22222222-2222-2222-2222-222222222222" }

### Full (Swagger Body)

{  
 "userId": "22222222-2222-2222-2222-222222222222"  
}

# Interactions

## POST /api/contacts/{contactId}/interactions — Create Interaction

**Purpose:** Log a communication/event for a contact (note, call, sms, email, meeting, task).

**Input:** Path: contactId; Body: { type, direction, channel, content?, occurredAt, status?, tags?, externalIds?, customFields? }

**Output:** 201 Created with Interaction resource

**Expected Status:** 201 Created; 400 Bad Request; 401 Unauthorized; 403 Forbidden; 404 Not Found; 422 Validation Failed; 500 Server Error

### Minimal (Swagger Body)

{  
 "type": "note",  
 "direction": "system",  
 "channel": "gog",  
 "content": "Initial intake completed.",  
 "occurredAt": "2025-08-27T13:00:00Z"  
}

### Full (Swagger Body)

{  
 "type": "call",  
 "direction": "outbound",  
 "channel": "twilio",  
 "status": "completed",  
 "content": "Chatted about pre-approval; follow-up tomorrow.",  
 "tags": ["CRM"],  
 "externalIds": { "twilioSid": "CA123..." },  
 "occurredAt": "2025-08-27T13:00:00Z",  
 "customFields": { "durationSec": "420" }  
}

## GET /api/contacts/{contactId}/interactions — List Interactions

**Purpose:** Return interactions for a contact (scoped by org).

**Input:** Path: contactId

**Output:** 200 OK with List<Interaction>

**Expected Status:** 200 OK; 401 Unauthorized; 403 Forbidden; 404 Not Found

# Agent / Orchestration (Dev Stub)

## POST /api/agent/suggest-followup — Suggest Follow-up (Dev Only)

**Purpose:** Return a mock suggestion payload for UI testing (no DB writes).

**Input:** Body (optional): { contactId?: Guid, context?: string }

**Output:** 200 OK with suggestion payload

**Expected Status:** 200 OK; 401 Unauthorized; 403 Forbidden; 404 Not Found (if contactId resolved)

### Minimal (Swagger Body)

{}

### Full (Swagger Body)

{  
 "contactId": "REPLACE\_CONTACT\_ID",  
 "context": "Recent call indicated interest in viewing downtown condos."  
}

# Enums

## GET /api/enums/{type} — Get Enum Values

**Purpose:** Return dynamic enum values by type (e.g., RelationshipState).

**Input:** Path: type

**Output:** 200 OK with array of enum values

**Expected Status:** 200 OK; 404 Not Found

# Health & Diagnostics

## GET /api/health — Health

**Purpose:** Basic liveness probe.

**Expected Status:** 200 OK

## GET /api/health/postgres — Postgres Health

**Purpose:** Connectivity test to PostgreSQL.

**Expected Status:** 200 OK; 500 Server Error

## GET /api/health/cosmos — Cosmos Health

**Purpose:** Connectivity test to Cosmos DB.

**Expected Status:** 200 OK; 500 Server Error

## GET /api/health/cosmos/item — Cosmos Sample Item

**Purpose:** Return a known development sample item.

**Expected Status:** 200 OK; 404 Not Found

# Common Mistakes Checklist

* For POST/PUT, 'Content-Type: application/json' missing in Swagger.
* JWT not added via Authorize modal (401).
* Org scope mismatch — JWT orgId differs from body.OrganizationId (403).
* Enum typos (e.g., RelationshipState) causing 422.
* Non-UTC timestamps in bodies causing validation or business rule failures.
* Using an ID from another tenant/organization (404/403).