Phase 2 - Workspace & Team Management API Documentation (Updated)

Base URL: `https://api.complifi.tech/api/v1`
Authentication: Bearer Token (JWT)

Overview

The Workspace & Team Management module allows companies to:

- Create multiple workspaces for different clients/projects
- Invite team members with workspace-level roles
- Control access based on role-based permissions
- Switch between multiple workspaces
- Manage team collaboration

Workspace Lifecycle

```
1. SUPER_ADMIN creates workspace
   ↓
2. SUPER_ADMIN automatically becomes workspace member with EDITOR role
   ↓
3. SUPER_ADMIN invites team members (EDITOR or VIEWER)
   ↓
4. Members collaborate within workspace
   ↓
5. SUPER_ADMIN can delete workspace (if no folders/documents exist)
```

Role Hierarchy

```
SUPER_ADMIN (Company Level)
    ↓
EDITOR (Workspace Level)
    ↓
VIEWER (Workspace Level)
```

Detailed Permissions Matrix

Action	SUPER_ADMIN	EDITOR	VIEWER
Workspace Management			
Create workspace	✅	❌	❌
View workspace	✅	✅	✅
Update workspace name	✅	❌	❌
Delete workspace	✅	❌	❌
Member Management			
View members	✅	✅	✅
Invite members	✅	❌	❌
Remove members	✅	❌	❌
Update member roles	✅	❌	❌
Content Management			
Create folders	✅	✅	❌
Upload documents	✅	✅	❌
View documents	✅	✅	✅
Edit documents	✅	✅	❌
Delete documents	✅	✅	❌

API Endpoints

1. Create Workspace

Purpose: Creates a new workspace and makes the creator an EDITOR member

Endpoint: `POST /workspaces`

Authentication: Required
Authorization: Any authenticated user (typically SUPER_ADMIN)

Request Headers:

```
Content-Type: application/json
Authorization: Bearer <token>
```

Request Body:

```json
{
  "name": "Q1 2025 Client Audit"
}
```

Validation Rules:

- `name` : Required, string, 2-100 characters

Success Response (201 Created):

```json
{
  "success": true,
  "message": "Workspace created successfully",
  "data": {
    "id": 1,
    "name": "Q1 2025 Client Audit",
    "companyId": 1,
    "createdAt": "2024-01-15T10:30:00.000Z",
    "updatedAt": "2024-01-15T10:30:00.000Z",
    "Company": {
      "id": 1,
      "name": "ABC Accounting Firm",
      "ownerId": 1
    },
    "members": [
      {
        "id": 1,
        "workspaceId": 1,
        "userId": 1,
        "role": "editor",
        "joinedAt": "2024-01-15T10:30:00.000Z",
        "User": {
          "id": 1,
          "name": "John Doe",
          "email": "john@example.com"
        }
      }
    ],
    "userRole": "editor"
  }
}
```

Important Notes:

- Creator automatically becomes an EDITOR member
- Workspace belongs to the user's company
- `userRole` indicates the current user's role in this workspace

Error Responses:

400 Bad Request - Validation error

```json
{
  "success": false,
  "message": "Validation error",
  "errors": ["Workspace name must be at least 2 characters long"]
}
```

401 Unauthorized - Invalid/missing token

```json
{
  "success": false,
  "message": "Unauthorized",
  "error": "Invalid token"
}
```

2. Get All User Workspaces

Purpose: Retrieves all workspaces where the user is a member

Endpoint: `GET /workspaces`

Authentication: Required
Authorization: Any authenticated user

Request Headers:

```
Authorization: Bearer <token>
```

Success Response (200 OK):

```
{
  "success": true,
  "message": "Workspaces fetched successfully",
  "data": [
    {
      "id": 1,
      "name": "Q1 2025 Client Audit",
      "companyId": 1,
      "createdAt": "2024-01-15T10:30:00.000Z",
      "updatedAt": "2024-01-15T10:30:00.000Z",
      "Company": {
        "id": 1,
        "name": "ABC Accounting Firm",
        "ownerId": 1
      },
      "members": [
        {
          "id": 1,
          "role": "editor",
          "User": {
            "id": 1,
            "name": "John Doe",
            "email": "john@example.com"
          }
        }
      ],
      "userRole": "editor"
    },
    {
      "id": 2,
      "name": "Tax Season Preparation",
      "companyId": 1,
      "createdAt": "2024-01-16T11:00:00.000Z",
      "updatedAt": "2024-01-16T11:00:00.000Z",
      "Company": {
        "id": 1,
        "name": "ABC Accounting Firm",
        "ownerId": 1
      },
      "members": [...],
      "userRole": "viewer"
    }
  ]
}
```

Use Cases:

- Display workspace list in sidebar
- Workspace switcher dropdown
- Dashboard overview of all accessible workspaces

3. Get Workspace by ID

Purpose: Retrieves detailed information about a specific workspace

Endpoint: `GET /workspaces/:id`

Authentication: Required
Authorization: User must be a member of the workspace

Path Parameters:

- `id` (integer): Workspace ID

Request Headers:

```
Authorization: Bearer <token>
```

Success Response (200 OK):

```json
{
  "success": true,
  "message": "Workspace fetched successfully",
  "data": {
    "id": 1,
    "name": "Q1 2025 Client Audit",
    "companyId": 1,
    "createdAt": "2024-01-15T10:30:00.000Z",
    "updatedAt": "2024-01-15T10:30:00.000Z",
    "Company": {
      "id": 1,
      "name": "ABC Accounting Firm",
      "ownerId": 1,
      "Owner": {
        "id": 1,
        "name": "John Doe",
        "email": "john@example.com"
      }
    },
    "members": [
      {
        "id": 1,
        "workspaceId": 1,
        "userId": 1,
        "role": "editor",
        "joinedAt": "2024-01-15T10:30:00.000Z",
        "User": {
          "id": 1,
          "name": "John Doe",
          "email": "john@example.com"
        }
      },
      {
        "id": 2,
        "workspaceId": 1,
        "userId": 3,
        "role": "viewer",
        "joinedAt": "2024-01-16T09:00:00.000Z",
        "User": {
          "id": 3,
          "name": "Sarah Wilson",
          "email": "sarah@example.com"
        }
      }
    ],
    "userRole": "editor"
  }
}
```

Error Responses:

403 Forbidden - Not a workspace member

```
{
  "success": false,
  "message": "Access denied to this workspace"
}
```

404 Not Found - Workspace doesn't exist

```
{
  "success": false,
  "message": "Workspace not found"
}
```

4. Update Workspace

Purpose: Updates workspace information (currently only name)

Endpoint: `PUT /workspaces/:id`

Authentication: Required
Authorization: User must be SUPER_ADMIN (company owner)

Path Parameters:

- `id` (integer): Workspace ID

Request Headers:

```
Content-Type: application/json
Authorization: Bearer <token>
```

Request Body:

```
{
  "name": "Q1 2025 Audit - Updated"
}
```

Validation Rules:

- `name` : Optional, string, 2-100 characters

Success Response (200 OK):

```
{
  "success": true,
  "message": "Workspace updated successfully",
  "data": {
    "id": 1,
    "name": "Q1 2025 Audit - Updated",
    "companyId": 1,
    "createdAt": "2024-01-15T10:30:00.000Z",
    "updatedAt": "2024-01-20T14:45:00.000Z",
    "Company": {...},
    "members": [...]
  }
}
```

Error Responses:

403 Forbidden - Not SUPER_ADMIN

```
{
  "success": false,
  "message": "Only company owner can update workspace"
}
```

5. Delete Workspace

Purpose: Deletes a workspace and all its members (cascade delete)

Endpoint: `DELETE /workspaces/:id`

Authentication: Required
Authorization: User must be SUPER_ADMIN (company owner)

Path Parameters:

- `id` (integer): Workspace ID

Request Headers:

```
Authorization: Bearer <token>
```

Success Response (200 OK):

```json
{
  "success": true,
  "message": "Workspace deleted successfully",
  "data": null
}
```

Important Notes:

- Only SUPER_ADMIN can delete workspaces
- All workspace members are removed
- All folders and documents in the workspace must be deleted first
- This is a destructive operation - confirm with user before calling

Error Responses:

403 Forbidden - Not SUPER_ADMIN

```json
{
  "success": false,
  "message": "Only company owner can delete workspace"
}
```

400 Bad Request - Workspace has existing data

```
{
  "success": false,
  "message": "Cannot delete workspace with existing folders or documents"
}
```

6. Get Workspace Members

Purpose: Retrieves all members of a workspace with their roles

Endpoint: `GET /workspaces/:id/members`

Authentication: Required
Authorization: User must be a member of the workspace

Path Parameters:

- `id` (integer): Workspace ID

Request Headers:

```
Authorization: Bearer <token>
```

Success Response (200 OK):

```json
{
  "success": true,
  "message": "Members fetched successfully",
  "data": [
    {
      "id": 1,
      "workspaceId": 1,
      "userId": 1,
      "role": "editor",
      "joinedAt": "2024-01-15T10:30:00.000Z",
      "createdAt": "2024-01-15T10:30:00.000Z",
      "updatedAt": "2024-01-15T10:30:00.000Z",
      "User": {
        "id": 1,
        "name": "John Doe",
        "email": "john@example.com"
      }
    },
    {
      "id": 2,
      "workspaceId": 1,
      "userId": 3,
      "role": "viewer",
      "joinedAt": "2024-01-16T09:00:00.000Z",
      "createdAt": "2024-01-16T09:00:00.000Z",
      "updatedAt": "2024-01-16T09:00:00.000Z",
      "User": {
        "id": 3,
        "name": "Sarah Wilson",
        "email": "sarah@example.com"
      }
    }
  ]
}
```

7. Invite Member

Purpose: Invites a user to join the workspace with a specified role

Endpoint: `POST /workspaces/:id/invite`

Authentication: Required
Authorization: User must be SUPER_ADMIN (company owner)

Path Parameters:

- `id` (integer): Workspace ID

Request Headers:

```
Content-Type: application/json
Authorization: Bearer <token>
```

Request Body:

```
{
  "email": "sarah@example.com",
  "role": "editor"
}
```

Validation Rules:

- `email` : Required, valid email format
- `role` : Required, must be one of: `"editor"` , `"viewer"`

Success Response (201 Created):

```json
{
  "success": true,
  "message": "Member invited successfully",
  "data": {
    "id": 2,
    "workspaceId": 1,
    "userId": 3,
    "role": "editor",
    "joinedAt": "2024-01-16T09:00:00.000Z",
    "createdAt": "2024-01-16T09:00:00.000Z",
    "updatedAt": "2024-01-16T09:00:00.000Z",
    "User": {
      "id": 3,
      "name": "Sarah Wilson",
      "email": "sarah@example.com"
    }
  }
}
```

Important Notes:

- Only SUPER_ADMIN can invite members
- User's email is automatically verified upon invitation (no OTP required)
- Cannot invite the same user twice to the same workspace

Error Responses:

403 Forbidden - Not SUPER_ADMIN

```json
{
  "success": false,
  "message": "Only company owner can invite members"
}
```

404 Not Found - Email not registered

```json
{
  "success": false,
  "message": "User with this email not found"
}
```

409 Conflict - Already a member

```
{
  "success": false,
  "message": "User is already a member of this workspace"
}
```

8. Accept Invitation

Purpose: Accepts a workspace invitation and creates/activates a user account

Endpoint: `POST /workspaces/accept-invitation`

Authentication: Not required (Public endpoint)
Authorization: Valid invitation token required

Request Headers:

```
Content-Type: application/json
```

Request Body:

```
{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "userData": {
    "name": "Sarah Wilson",
    "password": "SecurePass123!"
  }
}
```

Validation Rules:

- `token` : Required, string (JWT invitation token)
- `userData.name` : Required, string, 2-100 characters
- `userData.password` : Required, string, minimum 8 characters

Success Response (200 OK):

```json
{
  "success": true,
  "message": "Invitation accepted successfully",
  "data": {
    "user": {
      "id": 3,
      "name": "Sarah Wilson",
      "email": "sarah@example.com",
      "isEmailVerified": true,
      "hasCompletedOnboarding": true,
      "isSuperAdmin": false,
      "createdAt": "2024-01-16T09:00:00.000Z",
      "updatedAt": "2024-01-16T09:00:00.000Z"
    },
    "workspace": {
      "id": 1,
      "name": "Q1 2025 Client Audit",
      "companyId": 1
    },
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
  }
}
```

Important Notes:

- This is a **public endpoint** - no authentication required
- Email is extracted from the invitation token (not provided in request)
- If user doesn't exist, a new account is **automatically created**
- Email is **automatically verified** (no OTP required)
- User is immediately added as a workspace member with the invited role
- Returns an authentication token for immediate login
- Invitation token can only be used once

Workflow:

1. SUPER_ADMIN invites user via email using `/workspaces/:id/invite`
2. User receives invitation email with unique token
3. User clicks invitation link and provides name + password
4. System creates account (if new) or links existing account
5. User is added to workspace with assigned role
6. User receives auth token and can immediately access workspace

Error Responses:

400 Bad Request - Invalid/expired token

```
{
  "success": false,
  "message": "Invalid or expired invitation"
}
```

400 Bad Request - Validation error

```
{
  "success": false,
  "message": "Validation error",
  "errors": [
    "Name must be at least 2 characters long",
    "Password must be at least 8 characters long"
  ]
}
```

400 Bad Request - Invalid invitation data

```
{
  "success": false,
  "message": "Invalid invitation: email not found"
}
```

9. Remove Member

Purpose: Removes a member from the workspace

Endpoint: `DELETE /workspaces/:id/members/:memberId`

Authentication: Required
Authorization: User must be SUPER_ADMIN (company owner)

Path Parameters:

- `id` (integer): Workspace ID

- `memberId` (integer): User ID of the member to remove

Request Headers:

```
Authorization: Bearer <token>
```

Success Response (200 OK):

```json
{
  "success": true,
  "message": "Member removed successfully",
  "data": null
}
```

Important Notes:

- Only SUPER_ADMIN can remove members
- Cannot remove the SUPER_ADMIN themselves

Error Responses:

403 Forbidden - Not SUPER_ADMIN

```json
{
  "success": false,
  "message": "Only company owner can remove members"
}
```

400 Bad Request - Attempting to remove SUPER_ADMIN

```json
{
  "success": false,
  "message": "Cannot remove company owner from workspace"
}
```

404 Not Found - Member not found

```
{
  "success": false,
  "message": "Member not found in workspace"
}
```

10. Update Member Role

Purpose: Updates a member's role in the workspace

Endpoint: `PATCH /workspaces/:id/members/:memberId`

Authentication: Required
Authorization: User must be SUPER_ADMIN (company owner)

Path Parameters:

- `id` (integer): Workspace ID
- `memberId` (integer): User ID of the member

Request Headers:

```
Content-Type: application/json
Authorization: Bearer <token>
```

Request Body:

```
{
  "role": "viewer"
}
```

Validation Rules:

- `role` : Required, must be one of: `"editor"` , `"viewer"`

Success Response (200 OK):

```json
{
  "success": true,
  "message": "Member role updated successfully",
  "data": {
    "id": 2,
    "workspaceId": 1,
    "userId": 3,
    "role": "viewer",
    "joinedAt": "2024-01-16T09:00:00.000Z",
    "createdAt": "2024-01-16T09:00:00.000Z",
    "updatedAt": "2024-01-20T15:30:00.000Z",
    "User": {
      "id": 3,
      "name": "Sarah Wilson",
      "email": "sarah@example.com"
    }
  }
}
```

Error Responses:

403 Forbidden - Not SUPER_ADMIN

```json
{
  "success": false,
  "message": "Only company owner can update member roles"
}
```

400 Bad Request - Attempting to change SUPER_ADMIN's role

```json
{
  "success": false,
  "message": "Cannot change company owner's role"
}
```

11. Switch Workspace

Purpose: Retrieves workspace details for switching context in the UI

Endpoint: `GET /workspaces/:id/switch`

Authentication: Required
Authorization: User must be a member of the workspace

Path Parameters:

- `id` (integer): Workspace ID

Request Headers:

```
Authorization: Bearer <token>
```

Success Response (200 OK):

```json
{
  "success": true,
  "message": "Workspace switched successfully",
  "data": {
    "id": 2,
    "name": "Tax Season Preparation",
    "companyId": 1,
    "createdAt": "2024-01-16T11:00:00.000Z",
    "updatedAt": "2024-01-16T11:00:00.000Z",
    "Company": {...},
    "members": [...],
    "userRole": "viewer"
  }
}
```

Use Cases:

- Workspace switcher in navigation
- Load workspace-specific data
- Check user's role in the new workspace

Error Handling

Common Error Codes

Status Code	Meaning	Common Causes
400 Bad Request	Invalid input, validation errors	Invalid data
401 Unauthorized	Missing/invalid token	Not authenticated
403 Forbidden	Insufficient permissions	Not SUPER_ADMIN
404 Not Found	Resource doesn't exist	Workspace/member not found
409 Conflict	Duplicate member, constraint violation	Already invited
500 Server Error	Internal server error	System error

Error Response Format

```
{
  "success": false,
  "message": "Human-readable error message",
  "error": "Detailed error (development only)"
}
```

Quick Reference

Base URL

```
https://api.complifi.tech/api/v1
```

Authentication Header

```
Authorization: Bearer <your-jwt-token>
```

Endpoints Summary

Method	Endpoint	Auth	Authorization	Description
POST	`/workspaces`	✅	Any	Create workspace
GET	`/workspaces`	✅	Any	Get user workspaces
GET	`/workspaces/:id`	✅	Member	Get workspace details
PUT	`/workspaces/:id`	✅	SUPER_ADMIN	Update workspace
DELETE	`/workspaces/:id`	✅	SUPER_ADMIN	Delete workspace
GET	`/workspaces/:id/members`	✅	Member	Get members
POST	`/workspaces/:id/invite`	✅	SUPER_ADMIN	Invite member
POST	`/workspaces/accept-invitation`	❌	Public	Accept invitation
DELETE	`/workspaces/:id/members/:memberId`	✅	SUPER_ADMIN	Remove member
PATCH	`/workspaces/:id/members/:memberId`	✅	SUPER_ADMIN	Update role
GET	`/workspaces/:id/switch`	✅	Member	Switch workspace

Role Capabilities

Capability	SUPER_ADMIN	EDITOR	VIEWER
View workspace	✅	✅	✅
Update workspace	✅	❌	❌
Delete workspace	✅	❌	❌
Manage members	✅	❌	❌

Capability	SUPER_ADMIN	EDITOR	VIEWER
Create/edit content	✅	✅	❌
View content	✅	✅	✅