

GitSyncController

September 4, 2026

GitSyncController

Kind: Controller

Source: `atlوريا-monorepo/apps/api/src/git-sync/git-sync.controller.ts`

B1 git-sync settings, project-scoped. ProjectOrgGuard (not OrganizationGuard) because every route is keyed on `:projectId` and OrganizationGuard no-ops on projectId-only routes.

`GitSyncController` exposes project-scoped API endpoints for managing B1 Git Sync settings. It uses `ProjectOrgGuard` to authorize access through the `:projectId` route parameter, then delegates Git Sync configuration and operations to the underlying service layer.

Diagram

```
graph LR
  Client[API Client] --> Route["Route[\"/projects/:projectId/git-sync\"]"]
  Route --> Guard[Guard[ProjectOrgGuard]]
  Guard --> Controller[Controller[GitSyncController]]
  Controller --> Service[Service[Git Sync Service]]
  Service --> Settings[Settings[Project Git Sync Settings]]
  Service --> GitProvider[GitProvider[Git Provider Integration]]
```

Usage

```
type GitSyncSettings = {
  enabled: boolean;
  repositoryUrl?: string;
  branch?: string;
};

async function updateGitSyncSettings(
  projectId: string,
  settings: GitSyncSettings,
```

```

    accessToken: string,
  ) {
    const response = await fetch(
      `/api/projects/${projectId}/git-sync/settings`,
      {
        method: "PATCH",
        headers: {
          Authorization: `Bearer ${accessToken}`,
          "Content-Type": "application/json",
        },
        body: JSON.stringify(settings),
      },
    );

    if (!response.ok) {
      throw new Error(`Unable to update Git Sync settings: ${response.status}`);
    }

    return response.json();
  }

  await updateGitSyncSettings(
    "project_123",
    {
      enabled: true,
      repositoryUrl: "https://github.com/acme/project-config.git",
      branch: "main",
    },
    userAccessToken,
  );

```

AI Coding Instructions

- Keep all controller routes project-scoped and preserve the `:projectId` parameter expected by `ProjectOrgGuard`.
- Use `ProjectOrgGuard`, not `OrganizationGuard`; organization-only guard resolution does not apply to project-ID-only routes.
- Keep controller methods thin: validate request DTOs and delegate Git Sync behavior to the service layer.
- Ensure Git provider credentials, repository URLs, and sync configuration are handled securely and never returned unnecessarily in API responses.
- When adding endpoints, follow the existing project route and authorization patterns so settings cannot be accessed across projects.

Relationships

- MODULE_DECLARES → `get`
- MODULE_DECLARES → `update`
- MODULE_DECLARES → `mirrorNow`
- DEPENDS_ON → `GitSyncService`

Referenced By

- `GitSyncModule` (MODULE_DECLARES)