

GitHubService

September 5, 2026

GitHubService

Kind: Service**Source:** [atloria-monorepo/apps/api/src/github/github.service.ts](https://github.com/atloria-monorepo/apps/api/src/github/github.service.ts)

`GitHubService` manages GitHub OAuth connectivity for the API, including generating authorization URLs, processing OAuth callbacks, and disconnecting linked accounts. It also exposes repository and branch listing operations for authenticated GitHub users, acting as the backend integration layer between the application and GitHub.

Methods

Method	Signature	Returns	Description
<code>getAuthUrl</code>	<code>getAuthUrl(userId: string)</code>	<code>string</code>	Generate the GitHub OAuth authorization URL
<code>handleCallback</code>	<code>handleCallback(code: string, userId: string)</code>	<code>Promise<void></code>	Exchange authorization code for access token and store it
<code>getConnectionStatus</code>	<code>getConnectionStatus(userId: string)</code>	<code>`Promise<{ connected: boolean; username: string }>`</code>	<code>null }</code>
<code>listRepos</code>	<code>listRepos(userId: string)</code>	<code>Promise<any[]></code>	List repositories for the connected GitHub account

Method	Signature	Returns	Description
<code>listBranches</code>	<code>listBranches(userId: string, owner: string, repo: string)</code>	<code>Promise<any[]></code>	List branches for a specific repository
<code>disconnect</code>	<code>disconnect(userId: string)</code>	<code>Promise<void></code>	Disconnect GitHub from user account

Dependencies

- `PrismaService`
- `ConfigService`

Where it refuses work

- `GitHubService` stops the work with `Error` when `!response.ok`.
- `GitHubService` stops the work with `UnauthorizedException` when `!user?.githubAccessToken` — “GitHub not connected. Please authorize first.”.

Diagram

```
sequenceDiagram
    participant Client
    participant API as GitHub Controller
    participant Service as GitHubService
    participant GitHub as GitHub OAuth/API

    Client->>API: Request GitHub authorization URL
    API->>Service: getAuthUrl()
    Service->>GitHub: Build OAuth authorization request
    Service-->>API: Authorization URL
    API-->>Client: Redirect URL

    Client->>GitHub: Authorize application
    GitHub->>API: OAuth callback
    API->>Service: handleCallback()
    Service->>GitHub: Exchange code for access token
    Service->>Service: Persist GitHub connection

    Client->>API: List repositories / branches
    API->>Service: listRepos() or listBranches()
    Service->>GitHub: Fetch authenticated resources
    GitHub-->>Service: Repository or branch data
```

Service-->API: Resource list
API-->Client: JSON response

Usage

```
import { Controller, Get, Redirect } from '@nestjs/common';
import { GitHubService } from './github.service';

@Controller('github')
export class GitHubController {
  constructor(private readonly githubService: GitHubService) {}

  @Get('connect')
  @Redirect()
  connect() {
    return {
      url: this.githubService.getAuthUrl(),
    };
  }

  @Get('status')
  getStatus() {
    return this.githubService.getConnectionStatus();
  }

  @Get('repos')
  async getRepos() {
    return this.githubService.listRepos();
  }

  @Get('disconnect')
  async disconnect() {
    await this.githubService.disconnect();

    return { connected: false };
  }
}
```

AI Coding Instructions

- Keep GitHub OAuth token exchange, persistence, and API access encapsulated in `GitHubService` ; controllers should only map HTTP requests to service methods.
- Call `getConnectionStatus()` before repository or branch operations when the caller may not have an active GitHub connection.
- Ensure `handleCallback()` is invoked only from the OAuth callback route after the provider authorization response is available.

- Handle GitHub API failures and expired or revoked credentials consistently, returning an appropriate application-level error or disconnected status.
- Preserve the existing connection context when adding methods such as repository details, pull requests, or organization queries.

Relationships

- `DEPENDS_ON` → `PrismaService`
- `DEPENDS_ON` → `configservice`

Referenced By

- `GitHubController` (`DEPENDS_ON`)
- `GitHubModule` (`MODULE_PROVIDES`)
- `GitHubModule` (`MODULE_EXPORTS`)