

BitbucketService

September 4, 2026

BitbucketService

Kind: Service**Source:** [atloria-monorepo/apps/api/src/bitbucket/bitbucket.service.ts](#)

`BitbucketService` manages the Bitbucket integration lifecycle for the API, including OAuth authorization, callback handling, connection status, repository and branch discovery, and disconnection. It is used by backend controllers or workflows that need to connect a user or workspace to Bitbucket and access their repository metadata.

Methods

Method	Signature	Returns
<code>getAuthUrl</code>	<code>getAuthUrl(userId: string)</code>	<code>string</code>
<code>handleCallback</code>	<code>handleCallback(code: string, userId: string)</code>	<code>Promise<void></code>
<code>getConnectionStatus</code>	<code>getConnectionStatus(userId: string)</code>	<code>`Promise<{ connected: boolean; username: string</code>
<code>listRepos</code>	<code>listRepos(userId: string)</code>	<code>Promise<any[]></code>
<code>listBranches</code>	<code>listBranches(userId: string, owner: string, repo: string)</code>	<code>Promise<any[]></code>
<code>disconnect</code>	<code>disconnect(userId: string)</code>	<code>Promise<void></code>

Dependencies

- `PrismaService`
- `ConfigService`

Where it refuses work

- `BitbucketService` stops the work with `Error` when `!retryResponse.ok` .
- `BitbucketService` stops the work with `Error` when `!response.ok` .
- `BitbucketService` stops the work with `UnauthorizedException` when `!user?.bitbucketAccessToken` — “Bitbucket not connected. Please authorize first.”.

Diagram

```
sequenceDiagram
    participant Client
    participant API as Bitbucket Controller
    participant Service as BitbucketService
    participant Bitbucket as Bitbucket OAuth/API

    Client->>API: Request Bitbucket connection
    API->>Service: getAuthUrl()
    Service-->>API: OAuth authorization URL
    API-->>Client: Redirect to Bitbucket

    Client->>API: OAuth callback
    API->>Service: handleCallback()
    Service->>Bitbucket: Exchange authorization code for tokens
    Bitbucket-->>Service: Access/refresh tokens
    Service-->>API: Connection saved

    Client->>API: List repositories or branches
    API->>Service: listRepos() / listBranches()
    Service->>Bitbucket: Fetch repository metadata
    Bitbucket-->>Service: Repository or branch list
    Service-->>API: Results
    API-->>Client: Results
```

Usage

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

@Controller('integrations/bitbucket')
export class BitbucketController {
  constructor(private readonly bitbucketService: BitbucketService) {}

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

```

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

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

@Get('branches')
async getBranches() {
    return this.bitbucketService.listBranches();
}

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

    return { connected: false };
}
}

```

AI Coding Instructions

- Keep OAuth concerns inside `BitbucketService` ; controllers should delegate authorization URL generation and callback processing rather than implementing token exchange logic.
- Call `handleCallback()` only from the configured OAuth callback route after validating the callback request and preserving any required OAuth state.
- Check `getConnectionStatus()` before calling `listRepos()` or `listBranches()` so disconnected users receive a clear integration error rather than an upstream API failure.
- Treat repository and branch responses as external API data; add explicit DTOs or response mapping before exposing `any[]` results to public API consumers.
- Ensure `disconnect()` removes or invalidates stored Bitbucket credentials so subsequent repository requests cannot use stale tokens.

Relationships

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

Referenced By

- `BitbucketController` (DEPENDS_ON)
- `BitbucketModule` (MODULE_PROVIDES)
- `BitbucketModule` (MODULE_EXPORTS)