

KGSyncService

September 4, 2026

KGSyncService

Kind: Service**Source:** [atloria-monorepo/apps/parser-orchestrator/src/knowledge-graph/sync/kg-sync.service.ts](#)

`KGSyncService` coordinates synchronization between parsed project data and the knowledge graph. It supports full syncs, batch syncs, project-specific sync and resync operations, and exposes aggregate synchronization status for monitoring or API consumers.

Methods

Method	Signature	Returns	Description
<code>sync</code>	<code>sync(parseResult: ParseResult)</code>	<code>Promise<SyncResult></code>	Sync a single parse result to the Knowledge Graph
<code>syncBatch</code>	<code>syncBatch(parseResults: ParseResult[])</code>	<code>Promise<SyncResult></code>	Sync multiple parse results in batch (optimized)
<code>syncProject</code>	<code>syncProject(projectResult: ProjectParseResult)</code>	<code>Promise<SyncResult></code>	Sync a project parse result
<code>resyncProject</code>	<code>resyncProject(projectId: string, parseResults: ParseResult[])</code>	<code>Promise<SyncResult></code>	Clear and resync all parse results for a project
<code>getSyncStatus</code>	<code>getSyncStatus(projectId: string)</code>	<code>Promise<{</code>	

```
totalEntities: number;
totalRelationships: number;
lastSyncedAt?: Date;
```

}> | Get sync status for a project |

Dependencies

- KnowledgeGraphService

When something fails

- KGSyncService handles failure in 4 places: it logs it and continues in all 4.

Diagram

```
sequenceDiagram
    participant Caller
    participant KGSyncService
    participant Parser as Parsed Project Data
    participant KG as Knowledge Graph

    Caller->>KGSyncService: sync() / syncProject()
    KGSyncService->>Parser: Load entities and relationships
    Parser-->>KGSyncService: Parsed graph data
    KGSyncService->>KG: Upsert entities
    KGSyncService->>KG: Upsert relationships
    KG-->>KGSyncService: Sync result
    KGSyncService-->>Caller: Promise<SyncResult>

    Caller->>KGSyncService: getSyncStatus()
    KGSyncService->>KG: Query graph counts and last sync time
    KG-->>KGSyncService: Status data
    KGSyncService-->>Caller: totalEntities, totalRelationships, lastSyncedAt
```

Usage

```
import { Injectable } from '@nestjs/common';
import { KGSyncService } from '../knowledge-graph/sync/kg-sync.service';

@Injectable()
export class KnowledgeGraphAdminService {
  constructor(private readonly kgSyncService: KGSyncService) {}

  async syncCurrentProject() {
```

```

const result = await this.kgSyncService.syncProject();

return {
  success: true,
  result,
  status: await this.kgSyncService.getSyncStatus(),
};
}

async rebuildProjectGraph() {
  // Use resync when existing project graph data should be refreshed.
  return this.kgSyncService.resyncProject();
}
}

```

AI Coding Instructions

- Use `syncProject()` for normal project-level synchronization and `resyncProject()` when existing graph data must be rebuilt or refreshed.
- Prefer `syncBatch()` when processing multiple synchronization units to avoid repeatedly invoking individual sync operations.
- Treat `SyncResult` as the source of truth for operation outcomes; do not assume all entities or relationships were synchronized successfully.
- Call `getSyncStatus()` for observability and administrative endpoints rather than calculating graph totals in callers.
- Preserve NestJS dependency injection patterns: inject `KGSyncService` through constructors instead of instantiating it directly.

Relationships

- `DEPENDS_ON` → `knowledgegraphservice`

Referenced By

- `KnowledgeGraphModule` (`MODULE_PROVIDES`)
- `KnowledgeGraphModule` (`MODULE_EXPORTS`)
- `OrchestratorService` (`DEPENDS_ON`)