

TranslateQueueService

September 4, 2026

TranslateQueueService

Kind: Service

Source: `atloria-monorepo/apps/api/src/l10n/translate-queue.service.ts`

The 'l10n-translate' queue registration (B6). Present in every api pod for ENQUEUE; processing lives in TranslateWorker (kill-switch: L10N_TRANSLATE_WORKER !== 'false').

Jobs are DEDUPED per (document, language) via a deterministic jobId

`translate-<documentId>-<language>` — a burst of bulk-translate clicks collapses to one pending job per variant. jobIds NEVER contain ':' (BullMQ rejects colon custom ids); language codes are validated upstream to `[a-zA-Z-0-9-]` tokens, so the id stays legal.

`removeOnComplete` AND `removeOnFail` are both `true` — same reasoning as the docs-repo queue:

a RETAINED job with a deterministic id makes BullMQ silently ignore the next `add()` with

that id, wedging retranslation of that variant forever.

`TranslateQueueService` registers the `l10n-translate` BullMQ queue in every API pod and provides the API-side entry point for enqueueing translation work. Jobs are deduplicated per document and language using a deterministic `translate-<documentId>-<language>` ID, while actual processing is handled by `TranslateWorker` when `L10N_TRANSLATE_WORKER !== 'false'`.

Methods

Method	Signature	Returns	Description
<code>onModuleInit</code>	<code>onModuleInit()</code>	<code>void</code>	
<code>enqueueTranslate</code>	<code>enqueueTranslate(job: Omit<TranslateJob, 'type'>)</code>	<code>Promise<boolean></code>	Enqueue one variant translation.

Method	Signature	Returns	Description
onModuleDestroy	onModuleDestroy()	Promise<void>	

Dependencies

- ConfigService

Where it refuses work

- TranslateQueueService stops the work with an early return when `!this.queue`.

When something fails

- TranslateQueueService handles failure in 2 places: it logs it and continues in 1, and turns it into a return value in 1.

Diagram

```
sequenceDiagram
    participant API as API Controller/Service
    participant TQS as TranslateQueueService
    participant Queue as l10n-translate Queue
    participant Worker as TranslateWorker

    API->>TQS: enqueueTranslate(documentId, language)
    TQS->>TQS: Build translate-<documentId>-<language> jobId
    TQS->>Queue: add(job, payload, deterministic jobId)
    Queue-->>TQS: Job accepted or deduplicated
    TQS-->>API: Promise<boolean>

    Note over Worker: Runs only when L10N_TRANSLATE_WORKER !== "false"
    Worker->>Queue: Consume translation job
    Worker->>Worker: Translate document variant
    Worker->>Queue: Complete/fail and remove job
```

Usage

```
import { Injectable } from '@nestjs/common';
import { TranslateQueueService } from '../translate-queue.service';

@Injectable()
export class DocumentTranslationService {
  constructor(
```

```

    private readonly translateQueueService: TranslateQueueService,
  ) {}

  async requestTranslation(documentId: string, language: string) {
    const queued = await this.translateQueueService.enqueueTranslate(
      documentId,
      language,
    );

    return {
      documentId,
      language,
      queued,
    };
  }
}

```

AI Coding Instructions

- Use `TranslateQueueService` only to enqueue work; keep translation execution logic in `TranslateWorker` .
- Preserve deterministic job IDs in the format `translate-<documentId>-<language>` so repeated requests for the same variant collapse into one pending job.
- Do not add `:` characters to custom BullMQ job IDs; BullMQ rejects colon-containing IDs.
- Validate language codes upstream as `[a-zA-Z0-9-]` tokens before enqueueing to keep job IDs valid.
- Keep `removeOnComplete` and `removeOnFail` enabled; retained deterministic IDs can prevent future retractions from being enqueued.

Relationships

- `DEPENDS_ON` → `configservice`

Referenced By

- `L10nAutomationService` (`DEPENDS_ON`)
- `L10nController` (`DEPENDS_ON`)
- `L10nModule` (`MODULE_PROVIDES`)
- `L10nModule` (`MODULE_EXPORTS`)