AIUsageService

September 4, 2026

AIUsageService

Kind: Service

Source: `atloria-monorepo/apps/api/src/ai/ai-usage.service.ts`

`AIUsageService` centralizes AI consumption accounting for the API. It calculates request costs, records usage logs, deducts organization credits, and exposes usage and balance queries for billing-aware AI features.

Methods

Method	Signature	Returns	Description
`calculateCost`	`calculateCost(model: string, usage: TokenUsage)`	`UsageCost`	Calculate cost based on token usage and model pricing
`trackUsage`	`trackUsage(params: TrackUsageParams)`	`Promise<UsageCost>`	Track AI usage and deduct from organization credits (if applicable)
`deductCredits`	`deductCredits(organizationId: string, amount: number, transaction: {		

```
  referenceType: string;
  referenceId: string;
  description: string;
  userId: string;
})` | `Promise<{ balance: number }>` | Deduct credits from organization balance. |
```

| `getOrganizationUsage` | `getOrganizationUsage(organizationId: string, startDate: Date, endDate: Date)` | `unknown` | Get usage statistics for an organization |

`queryLogs`	`queryLogs(opts: { organizationId: string; projectId?: string; userId?: string; provider?: string; model?: string; operationType?: string; startDate?: string; endDate?: string; limit?: number; offset?: number; })`	`unknown`	List raw AI usage rows for an organization (paginated request history).
`getOrganizationBalance`	`getOrganizationBalance(organizationId: string)`	`unknown`	Get organization credit balance
`addCredits`	`addCredits(organizationId: string, amount: number, userId: string, description: string, reference: { type: string; id: string })`	`unknown`	Add credits to organization (when they purchase)

Dependencies

- `PrismaService`

Where it refuses work

- `AIUsageService` stops the work with `BadRequestException` when `!Number.isFinite(amount) || amount <= 0` — “Credit deduction amount must be a positive number”.
- `AIUsageService` stops the work with `PaymentRequiredException` when `result.count === 0`.
- `AIUsageService` stops the work with `BadRequestException` when `!Number.isFinite(amount) || amount <= 0` — “Credit amount must be a positive number”.
- `AIUsageService` stops the work with an early return when `orgCredit`.

When something fails

- `AIUsageService` handles failure in 1 place: it logs it and continues in all 1.

Diagram

```
sequenceDiagram
  participant Client
  participant AIService as AI Feature Service
  participant Usage as AIUsageService
  participant DB as Database

  Client->>AIService: Submit AI request
  AIService->>Usage: trackUsage(usage details)
  Usage->>Usage: calculateCost()
```

```
  Usage->>DB: Create usage log
  Usage->>Usage: deductCredits(cost)
  Usage->>DB: Update organization balance
  DB-->>Usage: Updated balance
  Usage-->>AIService: UsageCost
  AIService-->>Client: AI response and remaining credits
```

Usage

```ts
import { Injectable } from '@nestjs/common';
import { AIUsageService } from './ai-usage.service';

@Injectable()
export class GenerationService {
  constructor(private readonly aiUsageService: AIUsageService) {}

  async generateForOrganization(organizationId: string) {
    const usage = await this.aiUsageService.trackUsage({
      organizationId,
      model: 'gpt-4o-mini',
      inputTokens: 1_200,
      outputTokens: 450,
    });

    const { balance } = await this.aiUsageService.deductCredits(
      organizationId,
      usage.cost,
    );

    return {
      usage,
      remainingCredits: balance,
    };
  }
}
```

AI Coding Instructions

- Calculate and persist usage through `trackUsage()` rather than creating usage logs directly from feature services.
- Use `calculateCost()` with the correct model and token counts; avoid hard-coding pricing logic in AI endpoints.
- Deduct credits only after usage has been successfully recorded to keep billing and audit logs consistent.

- Check `getOrganizationBalance()` before expensive AI operations when insufficient-credit handling is required.
- Use `getOrganizationUsage()` and `queryLogs()` for reporting and administrative views instead of querying usage storage directly.

Relationships

- DEPENDS_ON → `PrismaService`

Referenced By

- `AIController` (DEPENDS_ON)
- `AIModule` (MODULE_PROVIDES)
- `AIModule` (MODULE_EXPORTS)
- `BillingService` (DEPENDS_ON)
- `ChangelogDraftsService` (DEPENDS_ON)
- `DocAutomationService` (DEPENDS_ON)
- `TechnicalDocsChatController` (DEPENDS_ON)
- `TechnicalDocsEnrichmentService` (DEPENDS_ON)
- `TechnicalDocsGenerationService` (DEPENDS_ON)
- `TechnicalDocsMcpController` (DEPENDS_ON)
- `TechnicalDocsQueue` (DEPENDS_ON)
- `TechnicalDocsController` (DEPENDS_ON)