

# FeedBackLoop — Add-On for Adobe Express

09.04.20XX

# Priya Pandey

MY github repo link: <a href="https://github.com/Coder2904/Feedback-loop">https://github.com/Coder2904/Feedback-loop</a>

## **Overview**

Streamline how teams review and approve designs in tools like Adobe Express — by collecting, organizing, and responding to stakeholder feedback all in one place, right inside your creative workflow.

#### Analogy:

Imagine Google Docs "suggestion" + Slack notifications... but for design projects directly within your design tool.

# CORE PROBLEM

Today: Feedback on designs is messy!

You send files/screenshots  $\rightarrow$  People reply in emails, Slack, meetings  $\rightarrow$  Feedback is scattered, hard to track, decisions delayed.

Direct pain point: Designers & reviewers waste time copying/pasting comments, chasing approvals.

#### Your Solution:

© Create a tightly integrated review/feedback system, inside the design environment, making feedback focused, fast, and actionable.

# **Goals**

# 1. Review Link Generation

- What: Instantly create a shareable review link for any current project with a single click inside the add-on panel.
- Why: Enables feedback from anyone (even those without Adobe accounts), speeding up review cycles.
- How: Panel provides options like deadline settings, anonymous feedback toggle, login requirements, and optional email notifications.

# 2. Structured Feedback Collection

- What: External reviewers access a lightweight web page showing the latest design and a structured feedback form (type, priority, comment).
- Why: All feedback is uniform, sortable, and doesn't get lost in long email or chat threads

# 3. Centralized Feedback Management Panel

- What: Designers view incoming feedback in real-time within the add-on panel.
- How: Inbox shows color-coded status, expandable cards for details, and quick actions to resolve, reply, or tag team members.
- Goal: No feedback is lost; everything's actionable at a glance.

# **Specifications**

# **©**Core Features Implemented:

#### 1. Generate Shareable Link

- Creates unique project ID and shareable URL
- Generates document thumbnail via sandbox
- Seamless UI state management

# 2. Document Integration

- Gets project title and info from Adobe Express document
- Generates visual preview/thumbnail of current design
- Exports design data for sharing

# 3. Feedback Management

- Mock feedback system (ready for backend integration)
- Real-time feedback display with status tracking
- Apply feedback workflow

# 4. Complete UI Flow

• Tab navigation: Generate → Share → Feedback

- Copy link functionality
- Status indicators and loading states

# @Implementation Structure

## 1. Project Setup

- Scaffold with Adobe Add-on CLI (npx @adobe/create-ccweb-add-on ...) using the JavaScript/React + sandbox template.
- Organize source files into:
  - /src Panel UI, feedback components.
  - /sandbox Document access logic (code. js for sandbox API interaction).
  - /server Express API, DB models, notification logic.
  - /public Static files for the feedback web app.

# 2. Panel & Web App Logic

- Panel Side (src/ui/components/):
  - Fetch and display current Adobe Express project info.
  - Generate and display review links via backend API.
  - Show real-time feedback list, color-coded by status/priority.
  - Expose actions: Mark as done, reply, tag, filter, sort.
- Feedback Webpage (public/):
  - Access via secure tokenized URL.
  - Render design preview (thumbnail/static snapshot).
  - Show guided feedback form (fields: type, priority, comment, optional name/email).
  - POST feedback securely to the backend.

# 3. Backend & API Structure

- REST Endpoints:
  - POST /create-review Generates review link, stores session.
  - GET /review/:token Serves design preview and feedback form for given session.

- POST /feedback Processes and stores feedback, triggers notifications.
- GET /feedback?reviewId=... Fetches all feedback for a session; used by panel.

#### Security:

- Enforce HTTPS, validate incoming tokens for every API call.
- Sanitize user inputs to prevent XSS/SQL injections.
- Use CORS to restrict origins, especially for review web access.

# 4. Adobe Express API Integration

#### Key APIs:

- <u>Document Sandbox API</u>: Used in /sandbox/code.js to access document structure, generate previews, and interact with artboards and metadata.
- <u>UI SDK</u>: For rendering panel, notifications, dialogs, and in-panel actions.
- <u>Event API</u>: To listen for project/document changes or review-related actions.
- Asset API: To quickly generate compressed thumbnails for web previews.

#### • Typical sequence:

- User chooses "Generate Review Link."
- Active document snapshot is created via sandbox API.
- Backend is called to log review session and produce secure shareable URL.
- Incoming feedback is fetched by UI SDK routines—no page reload needed.

# 5. Technology Stack

#### Adobe Express Add-on

- **Framework**: React 18 with TypeScript
- **UI Library**: Adobe Spectrum Web Components
- State Management: React Context + useReducer
- **Communication**: Adobe Runtime API with apiProxy pattern
- **Build System**: Webpack 5 with Adobe Express CLI
- **Styling**: Pure CSS (Tailwind not supported in add-on environment)

#### 6. Backend API

- **Runtime**: Node.js 18+ with Express.js
- **Language**: javascript
- **Database**: PostgreSQL 14+ (primary) + Redis 6+ (caching/sessions)
- **Storage**: AWS S3 for thumbnails and file exports
- **Authentication**: JWT tokens + OAuth2 (Adobe ID integration)
- **Real-time**: Socket.io for live notifications
- **Queue Processing**: Bull (Redis-based) for background tasks
- Al Integration: OpenAl GPT-4 Vision for design analysis

#### 7. Web Review Platform

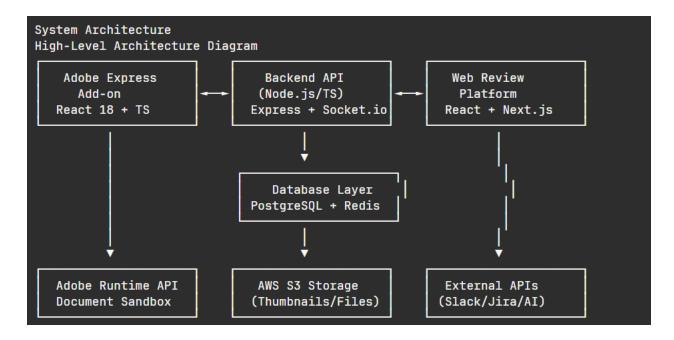
- **Framework**: React 18 with Next.js 13 (App Router)
- **Styling**: Tailwind CSS + Custom components
- **State Management**: SWR for data fetching + React Context
- **Real-time**: Socket.io client integration
- **Deployment**: Vercel with serverless functions

# **<u>Feedback Loop - Complete Workflow Architecture</u>**

System Overview

Adobe Express Add-on + Backend API + Web Review Platform + Real-time Notifications

# Overview On APIs and Endpoints(inshort)



## Authentication

- All API requests from the add-on require Adobe OAuth2 tokens.
- Public Review endpoints allow anonymous or email+PIN.

## **Core Endpoints**

#### **Projects**

Endpoint	Description
POST /api/projects/create	Create project
GET /api/projects/{project_id}/info	Project info
PUT /api/projects/{project_id}	Update

DELETE /api/projects/{project_id}	Delete

## Feedback

Endpoint	Description
<pre>GET /api/projects/{project_id}/feedback</pre>	Get feedback (filterable)
POST /api/feedback/{feedback_id}/resolve	Mark feedback as resolved
POST /api/feedback/bulk-resolve	Bulk resolve feedback
POST /review/{shareable_token}/feedback	Public feedback submit

# Analytics

Endpoint	Description
GET /api/analytics/feedback	Feedback analytics

# **©**Core Workflows

# Workflow 1: Generate Shareable Link Technical:

```
sequenceDiagram
   participant D as Designer (Add-on)
   participant S as Document Sandbox
   participant A as Backend API
   participant DB as PostgreSQL
   participant S3 as AWS S3
   participant WS as WebSocket
   D->>S: generateThumbnail()
   S-->>D: {dataURL, dimensions}
   D->>S: getDocumentInfo()
   S-->>D: {title, id, metadata}
   D->>A: POST /api/projects/create
   Note over A: Validate Adobe session
   A->>S3: Upload thumbnail
   $3-->>A: {url, key}
   A->>DB: INSERT project record
   DB-->>A: {project_id, token}
   A->>WS: Emit project_created event
   A-->>D: {project_id, shareable_link}
   D->>D: Update UI with link
```

#### Workflow 2: Review & Feedback

```
sequenceDiagram
    participant R as Reviewer
    participant W as Web Platform
    participant A as Backend API
    participant DB as PostgreSQL
    participant WS as WebSocket
    participant D as Designer Add-on
    participant E as External APIs
    R->>W: Visit shareable link
    W->>A: GET /review/{token}
    A->>DB: Query project details
    DB-->>A: {project_info, settings}
    A-->>W: Project data + form
    W-->>R: Render review interface
    Note over R: Reviewer submits feedback
    R->>W: Submit feedback form
    W->>A: POST /review/{token}/feedback
    A->>DB: INSERT feedback record
    A->>WS: Emit feedback_received
    WS->>D: Notify designer
    A->>E: Send external notifications
    A-->>W: Success response
   W-->>R: Thank you page
```

## **Workflow 3: Apply Feedback**

```
sequenceDiagram
    participant F as Feedback Submit
    participant A as Backend API
    participant R as Redis Queue
    participant WS as WebSocket Server
    participant D as Designer Add-on
    participant E as External Services
    participant N as Notification Services
    F->>A: POST feedback
    A->>R: Queue notification job
    A->>WS: Emit immediate event
    WS->>D: Real-time notification
    R->>N: Process notifications
    N->>E: Send Slack/Email alerts
    E-->>N: Delivery confirmation
    N->>A: Update notification status
```

# Security & Edge Cases

#### <u>Authentication & Authorization</u>

- Adobe Add-on: OAuth2 with Adobe ID
- **Review Platform**: Anonymous + optional email verification
- API Security: Rate limiting, CORS, input validation

• Data Privacy: GDPR compliance, data retention policies

#### Edge Case Handling

#### **Network Issues**

- Offline Mode: Cache feedback locally in add-on
- **Retry Logic**: Exponential backoff for API calls
- **Sync Resolution**: Conflict resolution for offline changes

#### Scale Handling

- CDN Distribution: Global thumbnail/asset delivery
- Database Sharding: Project-based partitioning
- Cache Strategy: Redis for hot feedback data
- **Queue Processing**: Background thumbnail generation

#### **Error Scenarios**

```
// Robust error handling patterns
try {
   const result = await sandboxProxy.generateThumbnail();
} catch (error) {
   if (error.code === 'SANDBOX_TIMEOUT') {
        // Fallback to default thumbnail
        return generateDefaultThumbnail();
}
   if (error.code === 'PERMISSION_DENIED') {
        // Show permission request dialog
        return requestDocumentAccess();
}
   // Log and show user-friendly message
   logger.error('Thumbnail generation failed', error);
   showNotification('Unable to generate preview. Please try ag
}
```

# **®**Performance Optimizations

# Frontend (Add-on)

- Lazy Loading: Load feedback on-demand
- **Debounced Requests**: Prevent excessive API calls
- **Local Caching**: Store project metadata locally
- **Progressive Enhancement**: Core features work offline

#### **Backend**

- **Database Indexing**: Optimized queries for feedback retrieval
- Connection Pooling: Efficient database connections
- Response Caching: Cache project metadata
- **Compression**: Gzip API responses

#### **Real-time Features**

- Room-based Sockets: Isolate project-specific updates
- **Heartbeat Monitoring**: Detect disconnected clients
- Message Queuing: Buffer updates during high load

# Deployment Architecture

## **Development Environment**

- **Docker Compose**: Local full-stack development
- Hot Reload: Instant add-on updates
- Mock Services: Simulated external dependencies

# **Production Deployment**

- Adobe Express: Add-on published to marketplace
- Backend: Node.js on AWS ECS/Lambda
- **Database**: AWS RDS (PostgreSQL) + ElastiCache (Redis)
- **Storage**: S3 with CloudFront CDN
- **Monitoring**: CloudWatch + application metrics

## **CI/CD Pipeline**

# Automated deployment workflow
stages:
- test: Unit + integration tests
- build: Create add-on package + backend images
- deploy-staging: Automated staging deployment
- e2e-tests: Full workflow validation
- deploy-production: Manual approval + deployment
- monitor: Post-deployment health checks

# **Monitoring & Analytics**

# **Key Metrics**

- Usage: Projects created, links shared, feedback submitted
- **Performance**: API response times, thumbnail generation speed
- **Errors**: Failed operations, user drop-off points
- **Engagement**: Time spent reviewing, feedback quality

# **Alerting**

- System Health: API downtime, database connectivity
- **User Experience**: High error rates, slow responses

• Business Metrics: Drop in project creation, feedback rates



# **Database Schema**

```
-- Users table

CREATE TABLE users (
   id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
   adobe_user_id VARCHAR UNIQUE NOT NULL,
   email VARCHAR NOT NULL,
   name VARCHAR NOT NULL,
   avatar_url VARCHAR,
   role user_role DEFAULT 'designer',
   settings JSONB DEFAULT '{}',
   created_at TIMESTAMP DEFAULT NOW(),
   updated_at TIMESTAMP DEFAULT NOW(),
   last_login TIMESTAMP
);
```

```
CREATE TABLE projects (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  adobe_document_id VARCHAR NOT NULL,
  title VARCHAR NOT NULL,
  description TEXT,
  creator_user_id UUID NOT NULL REFERENCES users(id),
  thumbnail_url VARCHAR,
  shareable_token VARCHAR UNIQUE NOT NULL,
  shareable_link VARCHAR UNIQUE NOT NULL,
  status project_status DEFAULT 'active',
  settings JSONB DEFAULT '{}',
  metadata JSONB DEFAULT '{}',
  expires_at TIMESTAMP DEFAULT (NOW() + INTERVAL '30 days'),
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW(),
 last_feedback_at TIMESTAMP
);
```

```
CREATE TABLE feedback (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  project_id UUID NOT NULL REFERENCES projects(id) ON DELETE CASCADE,
  reviewer_name VARCHAR NOT NULL,
  reviewer_email VARCHAR,
  reviewer_ip INET,
  comment TEXT NOT NULL,
  category feedback_category DEFAULT 'general',
  priority priority_level DEFAULT 'medium',
  rating INTEGER CHECK (rating >= 1 AND rating <= 5),
  position_data JSONB,
  browser_info JSONB,
  status feedback_status DEFAULT 'pending',
  resolution_notes TEXT,
  resolved_by UUID REFERENCES users(id),
  resolved_at TIMESTAMP,
  created_at TIMESTAMP DEFAULT NOW(),
  updated_at TIMESTAMP DEFAULT NOW()
```

```
CREATE TABLE review_sessions (
   id UUID PRIMARY KEY DEFAULT gen_random_uvid(),
   project_id UUID NOT NULL REFERENCES projects(id) ON DELETE CASCADE,
   reviewer_info JSONB NOT NULL,
   session_token VARCHAR UNIQUE NOT NULL,
  ip_address INET,
  user_agent TEXT,
  started_at TIMESTAMP DEFAULT NOW(),
  last_activity TIMESTAMP DEFAULT NOW(),
   expires_at TIMESTAMP DEFAULT (NOW() + INTERVAL '24 hours')
 );
CREATE TABLE notifications (
 id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
 user_id UUID NOT NULL REFERENCES users(id) ON DELETE CASCADE,
 project_id UUID REFERENCES projects(id) ON DELETE CASCADE,
 feedback_id UUID REFERENCES feedback(id) ON DELETE CASCADE,
 type notification_type NOT NULL,
 title VARCHAR NOT NULL,
 message TEXT NOT NULL,
 data JSONB DEFAULT '{}',
 sent_at TIMESTAMP,
 read_at TIMESTAMP,
 created_at TIMESTAMP DEFAULT NOW()
```

# **@**Future Enhancements(VERSION-2 OF THE PRODUCT)

#### **Advanced Features**

- Al-Powered Suggestions: Automated design recommendations
- Version Control: Design iteration tracking
- Collaborative Editing: Real-time multi-user design
- Advanced Analytics: Feedback sentiment analysis

## **Integration Expansions**

- **Slack/Teams**: Direct feedback notifications
- **Figma/Sketch**: Cross-platform compatibility
- **CRM Systems**: Customer feedback integration
- **Project Management**: Jira/Asana task creation

#### **Enterprise Features**

- **SSO Integration**: Enterprise authentication
- **Custom Branding**: White-label review platform
- Advanced Permissions: Role-based access control
- API Access: Third-party integrations

# **External Integrations**

#### Slack & Teams

- Connect via OAuth2 (UI flow, DB config)
- Project/review notifications posted as Slack messages (interactive deep link buttons)

#### <u>Jira</u>

- Ticket auto-generation from high-priority feedback
- Deep linking to Adobe Express files

#### Al (OpenAl GPT-4 Vision)

Secure single-use API keys

- Design/UX prompt engineering
- Results stored, privacy-respecting

#### **Development Setup**

- Local:
  - docker-compose up (DB, Redis, backend, frontend, add-on sandbox)
  - Hot reload for all codebases
  - Environment configs in .env.\*
  - Seed scripts: /scripts/seed.ts
- Staging:
  - Deployable to AWS ECS, Vercel

#### Deployment Architecture

- Backend: AWS ECS/Lambda
- DB: AWS RDS Postgres + ElastiCache
- Storage: S3 + CloudFront
- CI/CD: GitHub Actions, workflows for test/build/deploy/monitor
- IaC: Terraform for all infra (examples in /infra/)

#### Monitoring & Analytics

- Health checks (liveness/readiness) on all services
- Sentry/DataDog hooks for all major errors
- Analytics: Usage, response times, engagement, feedback quality, reviewer activity
- Automated alerts: high error rates, slow queries, service downtime

#### Testing Strategy

- End-to-end: Cypress scripts for link generation, feedback submission, notification,
   CRUD
- Unit: Business logic (feedback validation, analytics rollup, webhooks, etc.)
- API Contract: Type-checked DTOs and API integration tests

# **Technical Implementation Priority**

#### Phase 1 (MVP - 2 weeks)

- 1. Backend API core endpoints
- 2. Database setup & basic schema
- 3. Review platform web interface
- 4. Add-on ↔ Backend integration

#### Phase 2 (Enhanced - 4 weeks)

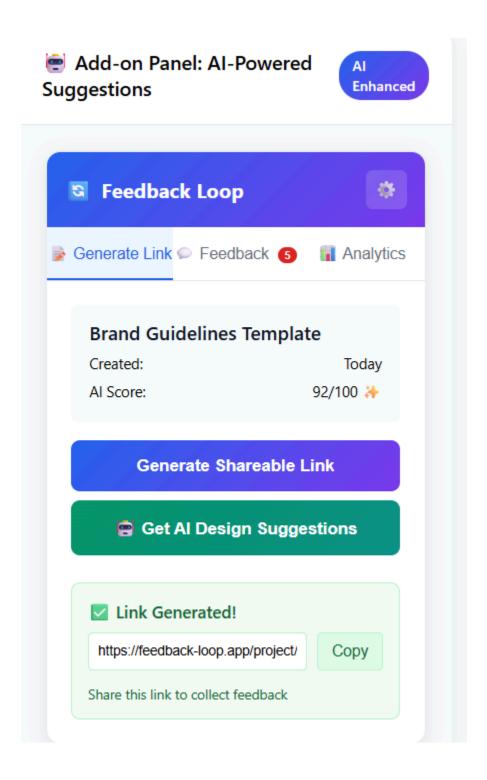
- 1. Real-time notifications
- 2. Advanced feedback positioning
- 3. Export/import functionality
- 4. Performance optimizations

## Phase 3 (Scale - 8 weeks)

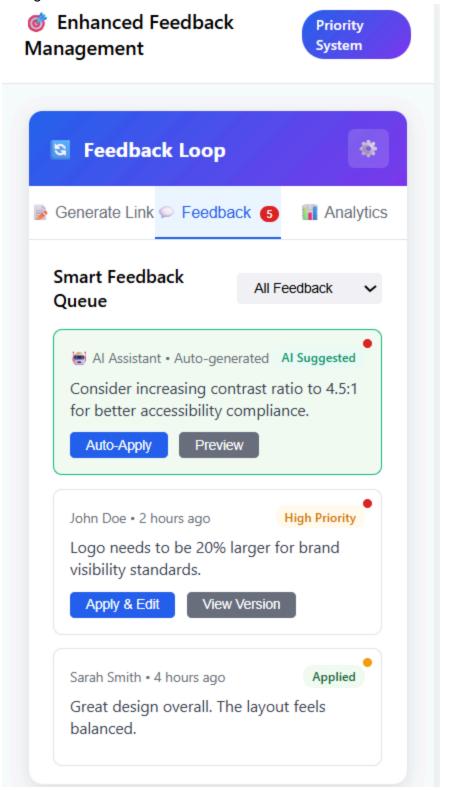
- 1. Enterprise features
- 2. Advanced analytics
- 3. Third-party integrations
- 4. Global deployment

This architecture ensures scalability, reliability, and maintainability while providing a seamless user experience across all platforms.

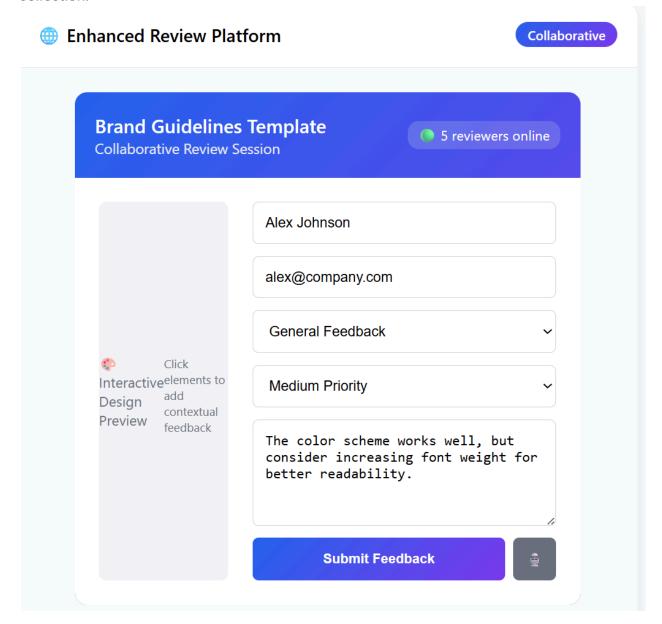
1.Al-enhanced add-on with design scoring, automated suggestions, and three-tab navigation including analytics. Smart recommendations based on design principles and user feedback patterns.



2.Advanced feedback management with AI suggestions, priority indicators, version control, and smart filtering. Auto-apply for AI recommendations and preview modes for changes.



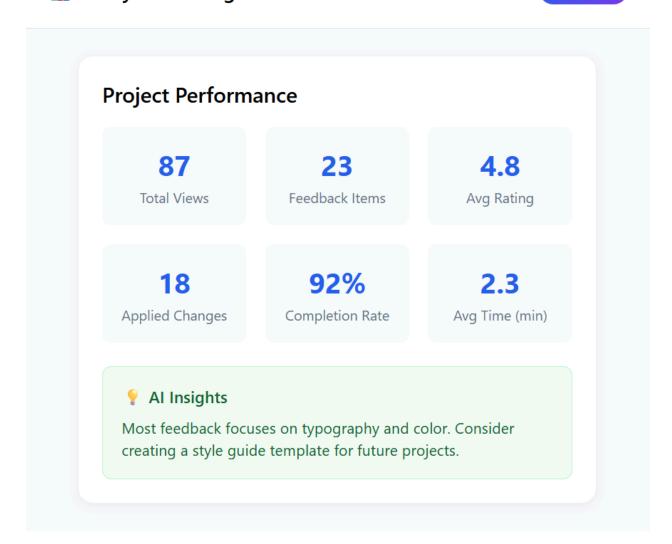
3. Enhanced review platform with real-time collaboration, contextual feedback on design elements, priority selection, categorization, and voice note support for richer feedback collection.



**4.**Comprehensive analytics showing project performance, engagement metrics, completion rates, and Al-powered insights to help improve design processes and feedback quality.



Analytics



**5.**Enterprise-grade integration hub connecting with popular tools like Slack, Jira, Figma, and project management platforms. Includes SSO, custom branding, and API access for large

organizations.



# 

**Enterprise** 

# **Connected Tools**

Streamline your workflow with integrated tools



Connected



Jira

Connected



**Figma** 

Available



**Teams** 

Connected



SSO

Enterprise



# **Analytics**

Connected

# Enterprise Features

- Single Sign-On (SSO) integration
- · Custom branding and white-label options
- · Advanced role-based permissions
- API access for custom integrations