

Smart Bookmarker Project Overview

Project Introduction

Smart Bookmarker is an intelligent bookmark management Chrome extension that uses artificial intelligence to automatically categorize, summarize, and tag bookmarks. The project provides advanced features including AI-powered analysis, smart categorization, Q&A system, and learning assistant functionality with optional cloud synchronization.

Core Features

AI-Powered Bookmark Analysis: Automatic categorization, intelligent summarization, tag generation, content analysis

Smart Classification System: Topic-based classification, confidence scoring, multi-level classification

Q&A System: Natural language search, intelligent recommendations, content understanding

Learning Assistant: Interactive learning, knowledge consolidation

Multi-language Support: Internationalization, localization

Cloud Sync: Optional synchronization, real-time sync, cross-platform

Technical Architecture

Chrome Extension Architecture

Core Components: - background.js: Main extension logic handling bookmark management, AI processing queue, storage synchronization, and message routing - options.html/js: Full-featured settings page with bookmark management, folder tree, AI configuration, and Q&A; system - popup.html/js: Compact popup interface for quick bookmark access and current page bookmarking - manifest.json: Chrome extension configuration file **Other Components:** - sidebar.html/js: Side panel interface - newtab.html/js: New tab page for starred bookmarks - learningAssistant.js: Learning assistant functionality

Server Architecture

Backend Technology Stack: - Express.js: Web framework - MongoDB: Database - Socket.io: Real-time synchronization - JWT: Authentication - Passport: OAuth integration (Google, GitHub) **Server Components:** - server.js: Main server entry point - routes/: API endpoints (authentication, items, user settings) - models/: Database schemas (User, Item, AIConfig) - services/: Business logic - middleware/: Authentication, error handling, logging

Data Model

Bookmark Data Structure

- clientId: Unique identifier for local bookmarks (temporary, for sync) - _id: Server-assigned ID for synchronized items (authoritative) - aiStatus: Processing state ('pending', 'processing', 'completed', 'failed') - smartCategories: AI-generated topic classifications with confidence scores - Enhanced metadata: summary, tags, key points, content type, reading level

AI Integration

Supported AI Providers: - OpenAI (GPT models) - DeepSeek - OpenRouter
AI Analysis Levels: - Basic: Summary and tags - Standard: Basic + category and key points - Detailed: Standard + content type, reading level, etc.

API Design

RESTful APIs: - For batch operations and initial sync - Authentication and user management - Bookmark CRUD operations
WebSocket: - Real-time incremental synchronization - Real-time status updates
Unified Item Model: - Both bookmarks and folders use same data structure - Smart categories with confidence metrics

Development Guide

Chrome Extension Development: 1. Load Extension: Open Chrome and go to chrome://extensions/, enable Developer mode, click "Load unpacked" and select this directory 2. Testing: Use Chrome DevTools for debugging, check extension popup and options page 3. Localization Testing: Switch between English and Chinese in options, verify all UI strings **Server Development:** 1. Install Dependencies: cd server && npm install 2. Environment Setup: Copy .env.example to .env, configure MongoDB connection and JWT secret 3. Run Server: npm run dev (development) or npm start (production)

File Organization

Extension Files: - background.js: Core extension logic and AI processing - options.html/js: Main user interface for bookmark management - popup.html/js: Quick-access popup interface - manifest.json: Extension configuration - _locales/: Internationalization files for English and Chinese - newtab.html/js: New tab page - icons/: Extension icons **Server Files:** - server/: Backend server code - server.js: Main server entry point - routes/: API endpoints - models/: Database schemas - services/: Business logic - middleware/: Authentication and error handling - config/: Database and logger configuration

Important Implementation Notes

- Storage Strategy: Uses Chrome's local storage with optional server sync - AI Queue Management: Background script manages concurrent AI processing with rate limiting - Message Passing: Extension components communicate via Chrome runtime messaging - Error Handling: Comprehensive error handling for AI API calls and network operations - Performance: Implements lazy loading and efficient data structures for large bookmark collections - Sync Protocol: Client-server ID handshake for reliable cross-platform synchronization

Version Information

- Chrome Extension Version: 1.5 - Server Version: 1.1.0 - Last Updated: 2025-11-17 - License: ISC