@push.rocks/smartai

A TypeScript library for integrating and interacting with multiple AI models, offering capabilities for chat and potentially audio responses.

- readme.md for @push.rocks/smartai
- changelog.md for @push.rocks/smartai

readme.md for @push.rocks/smartai

A unified provider registry for the Vercel AI SDK □□⚡

npm version TypeScript License: MIT

SmartAI gives you a single `getModel()` function that returns a standard `LanguageModelV3` for **any** supported provider — Anthropic, OpenAI, Google, Groq, Mistral, XAI, Perplexity, or Ollama. Use the returned model with the Vercel AI SDK's `generateText()`, `streamText()`, and tool ecosystem. Specialized capabilities like vision, audio, image generation, document analysis, and web research are available as dedicated subpath imports.

Issue Reporting and Security

For reporting bugs, issues, or security vulnerabilities, please visit community.foss.global/. This is the central community hub for all issue reporting. Developers who sign and comply with our contribution agreement and go through identification can also get a code.foss.global/ account to submit Pull Requests directly.

□□ Why SmartAI?

- □□ **One function, eight providers** — `getModel()` returns a standard `LanguageModelV3`. Switch providers by changing a string.
- □□ **Built on Vercel AI SDK** — Uses `ai` v6 under the hood. Your model works with `generateText()`, `streamText()`, tool calling, structured output, and everything else in the AI SDK ecosystem.
- □□ **Custom Ollama provider** — A full `LanguageModelV3` implementation for Ollama with support for `think` mode, `num_ctx`, auto-tuned temperature for Qwen models, and native tool calling.
- □□ **Anthropic prompt caching** — Automatic `cacheControl` middleware reduces cost and latency on repeated calls. Enabled by default, opt out with `promptCaching: false`.
- □□ **Modular subpath exports** — Vision, audio, image, document, and research capabilities ship as separate imports. Only import what you need.

- ⚡ **Zero lock-in** — Your code uses standard AI SDK types. Swap providers without touching application logic.

□□ Installation

```
pnpm install @push.rocks/smartai
```

□□ Quick Start

```
import { getModel, generateText, streamText } from '@push.rocks/smartai';

// Get a model for any provider
const model = getModel({
  provider: 'anthropic',
  model: 'claude-sonnet-4-5-20250929',
  apiKey: process.env.ANTHROPIC_TOKEN,
});

// Use it with the standard AI SDK functions
const result = await generateText({
  model,
  prompt: 'Explain quantum computing in simple terms.',
});

console.log(result.text);
```

That's it. Change `provider` to `'openai'` and `model` to `'gpt-4o'` and the rest of your code stays exactly the same.

□□ Core API

`getModel(options): LanguageModelV3`

The primary export. Returns a standard `LanguageModelV3` you can use with any AI SDK function.

```
import { getModel } from '@push.rocks/smartai';
import type { ISmartAiOptions } from '@push.rocks/smartai';

const options: ISmartAiOptions = {
  provider: 'anthropic',  // 'anthropic' | 'openai' | 'google' | 'groq' | 'mistral' | 'xai' |
'perplexity' | 'ollama'
  model: 'claude-sonnet-4-5-20250929',
  apiKey: 'sk-ant-...',
  // Anthropic-only: prompt caching (default: true)
  promptCaching: true,
  // Ollama-only: base URL (default: http://localhost:11434)
  baseUrl: 'http://localhost:11434',
  // Ollama-only: model runtime options
  ollamaOptions: { think: true, num_ctx: 4096 },
};

const model = getModel(options);
```

Re-exported AI SDK Functions

SmartAI re-exports the most commonly used functions from `ai` for convenience:

```
import {
  getModel,
  generateText,
  streamText,
  tool,
  jsonSchema,
} from '@push.rocks/smartai';

import type {
  ModelMessage,
  ToolSet,
  StreamTextResult,
  LanguageModelV3,
} from '@push.rocks/smartai';
```

□□ Supported Providers

Provider	Package	Example Models
Anthropic	`@ai-sdk/anthropic`	`claude-sonnet-4-5-20250929`, `claude-opus-4-5-20250929`
OpenAI	`@ai-sdk/openai`	`gpt-4o`, `gpt-4o-mini`, `o3-mini`
Google	`@ai-sdk/google`	`gemini-2.0-flash`, `gemini-2.5-pro`
Groq	`@ai-sdk/groq`	`llama-3.3-70b-versatile`, `mixtral-8x7b-32768`
Mistral	`@ai-sdk/mistral`	`mistral-large-latest`, `mistral-small-latest`
XAI	`@ai-sdk/xai`	`grok-3`, `grok-3-mini`
Perplexity	`@ai-sdk/perplexity`	`sonar-pro`, `sonar`
Ollama	Custom `LanguageModelV3`	`qwen3:8b`, `llama3:8b`, `deepseek-r1`

□□ Text Generation

Generate Text

```
import { getModel, generateText } from '@push.rocks/smartai';

const model = getModel({
  provider: 'openai',
  model: 'gpt-4o',
  apiKey: process.env.OPENAI_TOKEN,
});

const result = await generateText({
  model,
  system: 'You are a helpful assistant.',
  prompt: 'What is 2 + 2?',
});

console.log(result.text); // "4"
```

Stream Text

```
import { getModel, streamText } from '@push.rocks/smartai';

const model = getModel({
  provider: 'anthropic',
  model: 'claude-sonnet-4-5-20250929',
  apiKey: process.env.ANTHROPIC_TOKEN,
});

const result = await streamText({
  model,
  prompt: 'Count from 1 to 10.',
});

for await (const chunk of result.textStream) {
  process.stdout.write(chunk);
}
```

Tool Calling

```
import { getModel, generateText, tool, jsonSchema } from '@push.rocks/smartai';

const model = getModel({
  provider: 'anthropic',
  model: 'claude-sonnet-4-5-20250929',
  apiKey: process.env.ANTHROPIC_TOKEN,
});

const result = await generateText({
  model,
  prompt: 'What is the weather in London?',
  tools: {
    getWeather: tool({
      description: 'Get weather for a location',
      parameters: jsonSchema({
        type: 'object',
```

```
        properties: {
          location: { type: 'string' },
        },
        required: ['location'],
      }),
      execute: async ({ location }) => {
        return { temperature: 18, condition: 'cloudy' };
      },
    }),
  },
});
```

Ollama (Local Models)

The custom Ollama provider implements `LanguageModelV3` directly, calling Ollama's native `/api/chat` endpoint. This gives you features that generic OpenAI-compatible wrappers miss:

```
import { getModel, generateText } from '@push.rocks/smartai';

const model = getModel({
  provider: 'ollama',
  model: 'qwen3:8b',
  baseUrl: 'http://localhost:11434', // default
  ollamaOptions: {
    think: true,      // Enable thinking/reasoning mode
    num_ctx: 8192,     // Context window size
    temperature: 0.7,  // Override default (Qwen models auto-default to 0.55)
  },
});

const result = await generateText({
  model,
  prompt: 'Solve this step by step: what is 15% of 340?',
});

console.log(result.text);
```

Ollama Features

- **`think` mode** — Enables reasoning for models that support it (Qwen3, QwQ, DeepSeek-R1). The `think` parameter is sent at the top level of the request body as required by the Ollama API.
- **Auto-tuned temperature** — Qwen models automatically get `temperature: 0.55` when no explicit temperature is set, matching the recommended inference setting.
- **Native tool calling** — Full tool call support via Ollama's native format (not shimmed through OpenAI-compatible endpoints).
- **Streaming with reasoning** — `doStream()` emits proper `reasoning-start`, `reasoning-delta`, `reasoning-end` parts alongside text.
- **All Ollama options** — `num_ctx`, `top_k`, `top_p`, `repeat_penalty`, `num_predict`, `stop`, `seed`.

□□ Anthropic Prompt Caching

When using the Anthropic provider, SmartAI automatically wraps the model with caching middleware that adds `cacheControl: { type: 'ephemeral' }` to the last system message and last user message. This can significantly reduce cost and latency for repeated calls with the same system prompt.

```
// Caching enabled by default
const model = getModel({
  provider: 'anthropic',
  model: 'claude-sonnet-4-5-20250929',
  apiKey: process.env.ANTHROPIC_TOKEN,
});

// Opt out of caching
const modelNoCaching = getModel({
  provider: 'anthropic',
  model: 'claude-sonnet-4-5-20250929',
  apiKey: process.env.ANTHROPIC_TOKEN,
  promptCaching: false,
});
```

You can also use the middleware directly:

```
import { createAnthropicCachingMiddleware } from '@push.rocks/smartai';
import { wrapLanguageModel } from 'ai';
```

```
const middleware = createAnthropicCachingMiddleware();
const cachedModel = wrapLanguageModel({ model: baseModel, middleware });
```

□□ Subpath Exports

SmartAI provides specialized capabilities as separate subpath imports. Each one is a focused utility that takes a model (or API key) and does one thing well.

□□ Vision — `@push.rocks/smartai/vision`

Analyze images using any vision-capable model.

```
import { analyzeImage } from '@push.rocks/smartai/vision';
import { getModel } from '@push.rocks/smartai';
import * as fs from 'fs';

const model = getModel({
  provider: 'anthropic',
  model: 'claude-sonnet-4-5-20250929',
  apiKey: process.env.ANTHROPIC_TOKEN,
});

const description = await analyzeImage({
  model,
  image: fs.readFileSync('photo.jpg'),
  prompt: 'Describe this image in detail.',
  mediaType: 'image/jpeg', // optional, defaults to 'image/jpeg'
});

console.log(description);
```

`analyzeImage(options)` accepts:

- `model` — Any `LanguageModelV3` with vision support
- `image` — `Buffer` or `Uint8Array`
- `prompt` — What to ask about the image
- `mediaType` — `'image/jpeg'` | `'image/png'` | `'image/webp'` | `'image/gif'`

Audio — `@push.rocks/smartai/audio`

Text-to-speech using OpenAI's TTS models.

```
import { textToSpeech } from '@push.rocks/smartai/audio';
import * as fs from 'fs';

const stream = await textToSpeech({
  apiKey: process.env.OPENAI_TOKEN,
  text: 'Welcome to the future of AI development!',
  voice: 'nova',     // 'alloy' | 'echo' | 'fable' | 'onyx' | 'nova' | 'shimmer'
  model: 'tts-1-hd', // 'tts-1' | 'tts-1-hd'
  responseFormat: 'mp3', // 'mp3' | 'opus' | 'aac' | 'flac'
  speed: 1.0,         // 0.25 to 4.0
});

stream.pipe(fs.createWriteStream('welcome.mp3'));
```

Image — `@push.rocks/smartai/image`

Generate and edit images using OpenAI's image models.

```
import { generateImage, editImage } from '@push.rocks/smartai/image';

// Generate an image
const result = await generateImage({
  apiKey: process.env.OPENAI_TOKEN,
  prompt: 'A futuristic cityscape at sunset, digital art',
  model: 'gpt-image-1',     // 'gpt-image-1' | 'dall-e-3' | 'dall-e-2'
  quality: 'high',           // 'low' | 'medium' | 'high' | 'auto'
  size: '1024x1024',
  background: 'transparent', // gpt-image-1 only
  outputFormat: 'png',       // 'png' | 'jpeg' | 'webp'
  n: 1,
});

// result.images[0].b64_json — base64-encoded image data
const imageBuffer = Buffer.from(result.images[0].b64_json!, 'base64');
```

```
// Edit an existing image
const edited = await editImage({
  apiKey: process.env.OPENAI_TOKEN,
  image: imageBuffer,
  prompt: 'Add a rainbow in the sky',
  model: 'gpt-image-1',
});
```

□□ Document — `@push.rocks/smartai/document`

Analyze PDF documents by converting them to images and using a vision model. Uses `@push.rocks/smartpdf` for PDF-to-PNG conversion (requires Chromium/Puppeteer).

```
import { analyzeDocuments, stopSmartpdf } from '@push.rocks/smartai/document';
import { getModel } from '@push.rocks/smartai';
import * as fs from 'fs';

const model = getModel({
  provider: 'anthropic',
  model: 'claude-sonnet-4-5-20250929',
  apiKey: process.env.ANTHROPIC_TOKEN,
});

const analysis = await analyzeDocuments({
  model,
  systemMessage: 'You are a legal document analyst.',
  userMessage: 'Summarize the key terms and conditions.',
  pdfDocuments: [fs.readFileSync('contract.pdf')],
  messageHistory: [],  // optional: prior conversation context
});

console.log(analysis);

// Clean up the SmartPdf instance when done
await stopSmartpdf();
```

□□ Research — `@push.rocks/smartai/research`

Perform web-search-powered research using Anthropic's `web_search_20250305` tool.

```
import { research } from '@push.rocks/smartai/research';

const result = await research({
  apiKey: process.env.ANTHROPIC_TOKEN,
  query: 'What are the latest developments in quantum computing?',
  searchDepth: 'basic',     // 'basic' | 'advanced' | 'deep'
  maxSources: 10,           // optional: limit number of search results
  allowedDomains: ['nature.com', 'arxiv.org'],  // optional: restrict to domains
  blockedDomains: ['reddit.com'],               // optional: exclude domains
});

console.log(result.answer);
console.log('Sources:', result.sources);       // Array<{ url, title, snippet }>
console.log('Queries:', result.searchQueries); // search queries the model used
```

□□ Testing

```
# All tests
pnpm test

# Individual test files
tstest test/test.smartai.ts --verbose    # Core getModel + generateText + streamText
tstest test/test.ollama.ts --verbose     # Ollama provider (mocked, no API needed)
tstest test/test.vision.ts --verbose     # Vision analysis
tstest test/test.image.ts --verbose      # Image generation
tstest test/test.research.ts --verbose   # Web research
tstest test/test.audio.ts --verbose      # Text-to-speech
tstest test/test.document.ts --verbose   # Document analysis (needs Chromium)
```

Most tests skip gracefully when API keys are not set. The Ollama tests are fully mocked and require no external services.

□□ Architecture

```
@push.rocks/smartai
├── ts/                            # Core package
│   ├── index.ts                   # Re-exports getModel, AI SDK functions, types
│   ├── smartai.classes.smartai.ts  # getModel() — provider switch
│   ├── smartai.interfaces.ts      # ISmartAiOptions, TProvider, IOllamaModelOptions
│   ├── smartai.provider.ollama.ts  # Custom LanguageModelV3 for Ollama
│   ├── smartai.middleware.anthropic.ts  # Prompt caching middleware
│   └── plugins.ts                 # AI SDK provider factories
├── ts_vision/                     # @push.rocks/smartai/vision
├── ts_audio/                      # @push.rocks/smartai/audio
├── ts_image/                      # @push.rocks/smartai/image
├── ts_document/                   # @push.rocks/smartai/document
└── ts_research/                   # @push.rocks/smartai/research
```

The core package is a thin registry. `getModel()` creates the appropriate `@ai-sdk/*` provider, calls it with the model ID, and returns the resulting `LanguageModelV3`. For Anthropic, it optionally wraps the model with prompt caching middleware. For Ollama, it returns a custom `LanguageModelV3` implementation that talks directly to Ollama's `/api/chat` endpoint.

Subpath modules are independent — they import `ai` and provider SDKs directly, not through the core package. This keeps the dependency graph clean and allows tree-shaking.

License and Legal Information

Trademarks

license granted herein.

Use of these trademarks must comply with Task Venture Capital GmbH's Trademark Guidelines or the guidelines of the respective third-party owners, and any usage must be approved in writing. Third-party trademarks used herein are the property of their respective owners and used only in a descriptive manner, e.g. for an implementation of an API or similar.

Company Information

Task Venture Capital GmbH Registered at District Court Bremen HRB 35230 HB, Germany

For any legal inquiries or further information, please contact us via email at hello@task.vc.

By using this repository, you acknowledge that you have read this section, agree to comply with its terms, and understand that the licensing of the code does not imply endorsement by Task Venture Capital GmbH of any derivative works.

changelog.md for @push.rocks/smartai

2026-03-05 - 2.0.0 - BREAKING CHANGE(vercel-ai-sdk)

migrate to Vercel AI SDK v6 and introduce provider registry (getModel) returning LanguageModelV3

- Major API rewrite and module reorganization; bump package version to 1.0.0
- Replace many legacy provider implementations with @ai-sdk/* providers and a new Ollama adapter (LanguageModelV3-based)
- Add subpath exports for capability packages: ./vision, ./audio, ./image, ./document, ./research
- Introduce Anthropic prompt-caching middleware and provider-level promptCaching option
- Split functionality into focused ts_* packages (ts_audio, ts_image, ts_document, ts_vision, ts_research) and adapt tests accordingly
- Update dependencies and devDependencies to use ai SDK providers and newer package versions

2026-01-20 - 0.13.3 - fix()

no changes detected

- No files changed in the provided diff.
- No version bump required.

2026-01-20 - 0.13.2 - fix(repo)

no changes detected in diff; nothing to commit

- Git diff reported no changes — no files modified

- No code or dependency updates detected, so no version bump required

2026-01-20 - 0.13.1 - fix()

no changes detected; no release required

- No changes found in the provided git diff
- Current package version is 0.13.0

2026-01-20 - 0.13.0 - feat(provider.ollama)

add chain-of-thought reasoning support to chat messages and Ollama provider

- Added optional reasoning?: string to chat message and chat response interfaces to surface chain-of-thought data.
- Propagates reasoning from message history into formatted requests sent to Ollama.
- Maps Ollama response fields (thinking or reasoning) into ChatResponse.reasoning so downstream code can access model reasoning output.

2026-01-20 - 0.12.1 - fix(docs)

update documentation: clarify provider capabilities, add provider capabilities summary, polish examples and formatting, and remove Serena project config

- Removed .serena/project.yml and cleaned up .serena/.gitignore
- Added Provider Capabilities Summary and expanded/clarified provider tables in readme.md and readme.hints.md
- Clarified Anthropic extended thinking details and Mistral native PDF OCR notes
- Polished example code snippets and fixed minor typos/formatting (GPT-5 mention, ElevenLabs model note, consistent punctuation)
- Updated test command references and other README usage instructions

2026-01-20 - 0.12.0 - feat(ollama)

add support for base64-encoded images in chat messages and forward them to the Ollama provider

- Add optional images?: string[] to ChatMessage and ChatOptions interfaces (multimodal/vision support)
- Propagate images from messageHistory and ChatOptions to the Ollama API payload in chat, chatStreaming, and streaming handlers
- Changes are non-breaking: images are optional and existing behavior is preserved when absent

2026-01-20 - 0.11.0 - feat(ollama)

support defaultOptions and defaultTimeout for ollama provider

- Added ollama.defaultOptions object with fields: num_ctx, temperature, top_k, top_p, repeat_penalty, num_predict, stop, seed
- Added ollama.defaultTimeout option
- Pass defaultOptions and defaultTimeout into OllamaProvider constructor when initializing the provider
- Non-breaking change: existing behavior preserved if new fields are undefined

2026-01-20 - 0.10.1 - fix()

no changes detected — no release necessary

- No files changed in the provided diff; there are no code, documentation, or configuration modifications to release.

2026-01-18 - 0.10.0 - feat(mistral)

add Mistral provider with native PDF OCR and chat integration

- Adds dependency @mistralai/mistralai
- Implements ts/provider.mistral.ts providing chat() and document() (OCR) functionality
- Registers and exposes MistralProvider in SmartAi (options, lifecycle, conversation routing)
- Adds unit/integration tests: test.chat.mistral.ts and test.document.mistral.ts
- Updates readme.hints.md with Mistral usage, configuration and notes

2026-01-18 - 0.9.0 - feat(providers)

Add Anthropic extended thinking and adapt providers to new streaming/file APIs; bump dependencies and update docs, tests and configuration

- Add IAnthropicProviderOptions.extendedThinking with thinking modes (quick/normal/deep/off) and getThinkingConfig mapping budgets; apply thinking to Anthropic requests and omit temperature when thinking is enabled.
- Update Anthropic research flow to include thinking configuration and conditionally set temperature.
- OpenAI image editing: use openai.toFile to convert image/mask Buffers to uploadable files (image/png) before sending.
- ElevenLabs streaming: switch from response.streamNode() to response.stream() and convert web stream to Node stream using Readable.fromWeb().
- Upgrade dependencies and dev tools: @anthropic-ai/sdk ^0.71.2, @push.rocks/smartrequest ^5.0.1, @git.zone/tsbuild and related @git.zone packages, and other bumps in package.json.
- Tests and test imports updated to use @git.zone/tstest/tapbundle; many test files adjusted accordingly.
- Docs and hints updated: README and readme.hints.md include extended thinking docs, examples, formatting fixes, security/issue reporting guidance, and trademark/license clarifications.
- Project config tweaks: package build script changed, tsconfig baseUrl/paths added, npmextra.json reorganized (release registries added), .gitignore updated to ignore .claude/.serena local tooling files.

2025-10-30 - 0.8.0 - feat(provider.anthropic)

Add extended thinking modes to AnthropicProvider and apply thinking budgets to API calls

- Introduce IAnthropicProviderOptions.extendedThinking to configure thinking modes: 'quick' | 'normal' | 'deep' | 'off'.
- Add getThinkingConfig() helper mapping modes to token budgets (quick=2048, normal=8000, deep=16000, off=0).
- Apply thinking configuration to Anthropic API calls (chat, chatStream, vision, document, research) and increase max_tokens where appropriate (up to 20000).

- Add comprehensive tests (test/test.thinking.anthropic.ts) and update readme.hints.md with usage examples and recommendations.
- Add .claude/settings.local.json for local assistant permissions used in development/testing.

2025-10-10 - 0.7.7 - fix(MultiModalModel)

Lazy-load SmartPdf and guard document processing across providers; ensure SmartPdf is initialized only when needed

- Make SmartPdf lazy-loaded: smartpdfInstance is now nullable and no longer started automatically in start()
- Add ensureSmartpdfReady() to initialize and start SmartPdf on demand before document processing
- Providers updated (OpenAI, Anthropic, Ollama, xAI) to call ensureSmartpdfReady() and use the smartpdfInstance for PDF -> image conversion
- stop() now cleans up and nullifies smartpdfInstance to release resources
- Avoids starting a browser/process unless document() is actually used (reduces unnecessary resource usage)
- Add local Claude permissions file (.claude/settings.local.json) for tooling/configuration

2025-10-09 - 0.7.6 - fix(provider.elevenlabs)

Provide default ElevenLabs TTS voice fallback and add local tool/project configs

- ElevenLabsProvider: fallback to Samara voice id ('19STyYD15bswVz51nqLf') when no voiceId or defaultVoiceId is provided — avoids throwing an error on TTS calls.
- ElevenLabsProvider: continue to use 'eleven_v3' as the default model for TTS.
- Add .claude/settings.local.json with expanded allowed permissions for local tooling and web search.
- Add .serena/project.yml and .serena/.gitignore to include Serena project configuration and ignore cache.

2025-10-08 - 0.7.5 - fix(provider.elevenlabs)

Update ElevenLabs default TTS model to eleven_v3 and add local Claude permissions file

- Changed default ElevenLabs modelId from 'eleven_multilingual_v2' to 'eleven_v3' in ts/provider.elevenlabs.ts to use the newer/default TTS model.
- Added .claude/settings.local.json with a permissions allow-list for local Claude tooling and CI tasks.

2025-10-03 - 0.7.4 - fix(provider.anthropic)

Use image/png for embedded PDF images in Anthropic provider and add local Claude settings for development permissions

- AnthropicProvider: change media_type from 'image/jpeg' to 'image/png' when embedding images extracted from PDFs to ensure correct format in Anthropic requests.
- Add .claude/settings.local.json with development/testing permissions for local Claude usage (shell commands, webfetch, websearch, test/run tasks).

2025-10-03 - 0.7.3 - fix(tests)

Add extensive provider/feature tests and local Claude CI permissions

- Add many focused test files covering providers and features: OpenAI, Anthropic, Perplexity, Groq, Ollama, Exo, XAI (chat, audio, vision, document, research, image generation, stubs, interfaces, basic)
- Introduce .claude/settings.local.json to declare allowed permissions for local Claude/CI actions
- Replace older aggregated test files with modular per-feature tests (removed legacy combined tests and split into smaller suites)
- No changes to library runtime code — this change adds tests and CI/local agent configuration only

2025-10-03 - 0.7.2 - fix(anthropic)

Update Anthropic provider branding to Claude Sonnet 4.5 and add local Claude permissions

- Docs: Replace 'Claude 3 Opus' with 'Claude Sonnet 4.5' in README provider capabilities matrix.
- Config: Add .claude/settings.local.json to define local Claude permissions for tests and development commands.

2025-10-03 - 0.7.1 - fix(docs)

Add README image generation docs and .claude local settings

- Add .claude/settings.local.json with permission allow-list for local assistant tooling and web search
- Update README provider capabilities table to include an Images column and reference gpt-image-1
- Add Image Generation & Editing section with examples, options, and gpt-image-1 advantages
- Mark image generation support as implemented in the roadmap and remove duplicate entry

2025-10-03 - 0.7.0 - feat(providers)

Add research API and image generation/editing support; extend providers and tests

- Introduce ResearchOptions and ResearchResponse to the MultiModalModel interface and implement research() where supported
- OpenAiProvider: implement research(), add imageGenerate() and imageEdit() methods (gpt-image-1 / DALL·E support), and expose imageModel option
- AnthropicProvider: implement research() and vision handling; explicitly throw for unsupported image generation/editing
- PerplexityProvider: implement research() (sonar / sonar-pro support) and expose citation parsing
- Add image/document-related interfaces (ImageGenerateOptions, ImageEditOptions, ImageResponse) to abstract API

- Add image generation/editing/no-op stubs for other providers (Exo, Groq, Ollama, XAI) that throw informative errors to preserve API compatibility
- Add comprehensive OpenAI image generation tests and helper to save test outputs (test/test.image.openai.ts)
- Update README with Research & Web Search documentation, capability matrix, and roadmap entry for Research & Web Search API
- Add local Claude agent permissions file (.claude/settings.local.json) and various provider type/import updates

2025-09-28 - 0.6.1 - fix(provider.anthropic)

Fix Anthropic research tool identifier and add tests + local Claude permissions

- Replace Anthropic research tool type from 'computer_20241022' to 'web_search_20250305' to match the expected web-search tool schema.
- Add comprehensive test suites and fixtures for providers and research features (new/updated tests under test/ including anthropic, openai, research.* and stubs).
- Fix test usage of XAI provider class name (use XAIProvider) and adjust basic provider test expectations (provider instantiation moved to start()).
- Add .claude/settings.local.json with local Claude permissions to allow common CI/dev commands and web search during testing.

2025-09-28 - 0.6.0 - feat(research)

Introduce research API with provider implementations, docs and tests

- Add ResearchOptions and ResearchResponse interfaces and a new abstract research() method to MultiModalModel
- Implement research() for OpenAiProvider (deep research model selection, optional web search/tools, background flag, source extraction)
- Implement research() for AnthropicProvider (web search tool support, domain filters, citation extraction)
- Implement research() for PerplexityProvider (sonar / sonar-pro model usage and citation parsing)
- Add research() stubs to Exo, Groq, Ollama and XAI providers that throw a clear 'not yet supported' error to preserve interface compatibility
- Add tests for research interfaces and provider research methods (test files updated/added)

- Add documentation: readme.research.md describing the research API, usage and configuration
- Export additional providers from ts/index.ts and update provider typings/imports across files
- Add a 'typecheck' script to package.json
- Add .claude/settings.local.json (local agent permissions for CI/dev tasks)

2025-08-12 - 0.5.11 - fix(openaiProvider)

Update default chat model to gpt-5-mini and bump dependency versions

- Changed default chat model in OpenAiProvider from 'o3-mini' and 'o4-mini' to 'gpt-5-mini'
- Upgraded @anthropic-ai/sdk from ^0.57.0 to ^0.59.0
- Upgraded openai from ^5.11.0 to ^5.12.2
- Added new local Claude settings configuration (.claude/settings.local.json)

2025-08-03 - 0.5.10 - fix(dependencies)

Update SmartPdf to v4.1.1 for enhanced PDF processing capabilities

- Updated @push.rocks/smartpdf from ^3.3.0 to ^4.1.1
- Enhanced PDF conversion with improved scale options and quality controls
- Dependency updates for better performance and compatibility

2025-08-01 - 0.5.9 - fix(documentation)

Remove contribution section from readme

- Removed the contribution section from readme.md as requested
- Kept the roadmap section for future development plans

2025-08-01 - 0.5.8 - fix(core)

Fix SmartPdf lifecycle management and update dependencies

- Moved SmartPdf instance management to the MultiModalModel base class for better resource sharing
- Fixed memory leaks by properly implementing cleanup in the base class stop() method
- Updated SmartAi class to properly stop all providers on shutdown
- Updated @push.rocks/smartrequest from v2.1.0 to v4.2.1 with migration to new API
- Enhanced readme with professional documentation and feature matrix

2025-07-26 - 0.5.7 - fix(provider.openai)

Fix stream type mismatch in audio method

- Fixed type error where OpenAI SDK returns a web ReadableStream but the audio method needs to return a Node.js ReadableStream
- Added conversion using Node.js's built-in Readable.fromWeb() method

2025-07-25 - 0.5.5 - feat(documentation)

Comprehensive documentation enhancement and test improvements

- Completely rewrote readme.md with detailed provider comparisons, advanced usage examples, and performance tips
- Added comprehensive examples for all supported providers (OpenAI, Anthropic, Perplexity, Groq, XAI, Ollama, Exo)
- Included detailed sections on chat interactions, streaming, TTS, vision processing, and document analysis
- Added verbose flag to test script for better debugging

2025-05-13 - 0.5.4 - fix(provider.openai)

Update dependency versions, clean test imports, and adjust default OpenAI model configurations

- Bump dependency versions in package.json (@git.zone/tsbuild, @push.rocks/tapbundle, openai, etc.)
- Change default chatModel from 'gpt-4o' to 'o4-mini' and visionModel from 'gpt-4o' to '04-mini' in provider.openai.ts
- Remove unused 'expectAsync' import from test file

2025-04-03 - 0.5.3 - fix(package.json)

Add explicit packageManager field to package.json

- Include the packageManager property to specify the pnpm version and checksum.
- Align package metadata with current standards.

2025-04-03 - 0.5.2 - fix(readme)

Remove redundant conclusion section from README to streamline documentation.

- Eliminated the conclusion block describing SmartAi's capabilities and documentation pointers.

2025-02-25 - 0.5.1 - fix(OpenAiProvider)

Corrected audio model ID in OpenAiProvider

- Fixed audio model identifier from 'o3-mini' to 'tts-1-hd' in the OpenAiProvider's audio method.
- Addressed minor code formatting issues in test suite for better readability.
- Corrected spelling errors in test documentation and comments.

2025-02-25 - 0.5.0 - feat(documentation and configuration)

Enhanced package and README documentation

- Expanded the package description to better reflect the library's capabilities.
- Improved README with detailed usage examples for initialization, chat interactions, streaming chat, audio generation, document analysis, and vision processing.
- Provided error handling strategies and advanced streaming customization examples.

2025-02-25 - 0.4.2 - fix(core)

Fix OpenAI chat streaming and PDF document processing logic.

- Updated OpenAI chat streaming to handle new async iterable format.
- Improved PDF document processing by filtering out empty image buffers.
- Removed unsupported temperature options from OpenAI requests.

2025-02-25 - 0.4.1 - fix(provider)

Fix provider modules for consistency

- Updated TypeScript interfaces and options in provider modules for better type safety.
- Modified transform stream handlers in Exo, Groq, and Ollama providers for consistency.
- Added optional model options to OpenAI provider for custom model usage.

2025-02-08 - 0.4.0 - feat(core)

Added support for Exo AI provider

- Introduced ExoProvider with chat functionalities.
- Updated SmartAi class to initialize ExoProvider.
- Extended Conversation class to support ExoProvider.

2025-02-05 - 0.3.3 - fix(documentation)

Update readme with detailed license and legal information.

- Added explicit section on License and Legal Information in the README.
- Clarified the use of trademarks and company information.

2025-02-05 - 0.3.2 - fix(documentation)

Remove redundant badges from readme

- Removed Build Status badge from the readme file.
- Removed License badge from the readme file.

2025-02-05 - 0.3.1 - fix(documentation)

Updated README structure and added detailed usage examples

- Introduced a Table of Contents
- Included comprehensive sections for chat, streaming chat, audio generation, document processing, and vision processing
- Added example code and detailed configuration steps for supported AI providers
- Clarified the development setup with instructions for running tests and building the project

2025-02-05 - 0.3.0 - feat(integration-xai)

Add support for X.AI provider with chat and document processing capabilities.

- Introduced XAIProvider class for integrating X.AI features.
- Implemented chat streaming and synchronous chat for X.AI.
- Enabled document processing capabilities with PDF conversion in X.AI.

2025-02-03 - 0.2.0 - feat(provider.anthropic)

Add support for vision and document processing in Anthropic provider

- Implemented vision tasks for Anthropic provider using Claude-3-opus-20240229 model.
- Implemented document processing for Anthropic provider, supporting conversion of PDF documents to images and analysis with Claude-3-opus-20240229 model.
- Updated documentation to reflect the new capabilities of the Anthropic provider.

2025-02-03 - 0.1.0 - feat(providers)

Add vision and document processing capabilities to providers

- OpenAI and Ollama providers now support vision tasks using GPT-4 Vision and Llava models respectively.
- Document processing has been implemented for OpenAI and Ollama providers, converting PDFs to images for analysis.
- Introduced abstract methods for vision and document processing in the MultiModalModel class.
- Updated the readme file with examples for vision and document processing.

2025-02-03 - 0.0.19 - fix(core)

Enhanced chat streaming and error handling across providers

- Refactored chatStream method to properly handle input streams and processes in Perplexity, OpenAI, Ollama, and Anthropic providers.
- Improved error handling and message parsing in chatStream implementations.
- Defined distinct interfaces for chat options, messages, and responses.
- Adjusted the test logic in test/test.ts for the new classification response requirement.

2024-09-19 - 0.0.18 - fix(dependencies)

Update dependencies to the latest versions.

- Updated @git.zone/tsbuild from ^2.1.76 to ^2.1.84
- Updated @git.zone/tsrun from ^1.2.46 to ^1.2.49
- Updated @push.rocks/tapbundle from ^5.0.23 to ^5.3.0
- Updated @types/node from ^20.12.12 to ^22.5.5
- Updated @anthropic-ai/sdk from ^0.21.0 to ^0.27.3
- Updated @push.rocks/smartfile from ^11.0.14 to ^11.0.21
- Updated @push.rocks/smartpromise from ^4.0.3 to ^4.0.4
- Updated @push.rocks/webstream from ^1.0.8 to ^1.0.10
- Updated openai from ^4.47.1 to ^4.62.1

2024-05-29 - 0.0.17 - Documentation

Updated project description.

- Improved project description for clarity and details.

2024-05-17 - 0.0.16 to 0.0.15 - Core

Fixes and updates.

- Various core updates and fixes for stability improvements.

2024-04-29 - 0.0.14 to 0.0.13 - Core

Fixes and updates.

- Multiple core updates and fixes for enhanced functionality.

2024-04-29 - 0.0.12 - Core

Fixes and updates.

- Core update and bug fixes.

2024-04-29 - 0.0.11 - Provider

Fix integration for anthropic provider.

- Correction in the integration process with anthropic provider for better compatibility.

2024-04-27 - 0.0.10 to 0.0.9 - Core

Fixes and updates.

- Updates and fixes to core components.

- Updated tsconfig for improved TypeScript configuration.

2024-04-01 - 0.0.8 to 0.0.7 - Core and npmextra

Core updates and npmextra configuration.

- Core fixes and updates.
- Updates to npmextra.json for githost configuration.

2024-03-31 - 0.0.6 to 0.0.2 - Core

Initial core updates and fixes.

- Multiple updates and fixes to core following initial versions.

This summarizes the relevant updates and changes based on the provided commit messages. The changelog excludes commits that are version tags without meaningful content or repeated entries.