



## ZIKUU Repository Explorer

### Description

**ZIKUU v1.0** is a tool for exploring and managing repository data. It provides a structured view of the data, allowing users to navigate through various levels of detail, from high-level requirements to specific source code files. The interface is designed to be intuitive and easy to use, with a clear hierarchy and search capabilities.

ZIKUU is built on a robust architecture that supports a wide range of repository types and data formats. It offers a comprehensive set of features, including data visualization, reporting, and integration with external systems. The tool is designed to be flexible and scalable, making it suitable for use in a variety of environments and organizations.

**Repository Explorer** is a key component of the ZIKUU ecosystem, providing a central hub for accessing and analyzing repository data. It offers a rich set of features, including data exploration, visualization, and reporting. The tool is designed to be user-friendly and easy to integrate with existing systems, making it a valuable asset for any organization looking to optimize its repository management processes.

```
# zikuu-map

## What is this node?

This node hosts the core ZIKUU research systems including
data ingestion (Nerve), knowledge observation (Pivot),
and experimental AI interfaces.

## Node Profile

- NAME: 作る・学ぶ モノづくり塾 ZIKUU
- PLACE: 山梨県上野原市鶴島3378-18
- URL: https://zikuu.space
- MAIL: ask@zikuu.space
- SERVER: Ubuntu Server
- FACILITY: 工房 (木工、金属加工、溶接、伝統工芸)、AI研究室

## Responsibility

人間とAIがZIKUUノードのシステム構造を理解し、適切なリポジトリを調査するための入口。

## Survey Rules

調査は [ZIKUU Repository Protocol](./ZIKUU_REPOSITORY_PROTOCOL.md) に従う。

## Repositories

- Nerve

  外部のデータをZIKUUの知識として整える役目を負うシステムです。
```

Python zikuu-map  
調査は [ZIKUU Repository Protocol](./ZIKUU\_REPOSITORY\_PROTOCOL.md) に従う。  
外部のデータをZIKUUの知識として整える役目を負うシステムです。







```
[
  {
    "path": ".env.example",
    "mode": "100644",
    "type": "blob",
    "size": 40,
    "sha": "194a83bb47faae9d31372727e92daf60e45fa710",
    "url": "http://zikuu-gitea-host/api/v1/repos/admin/world-eye/git/blobs/194a83bb47faae9d31372727e92daf60e45fa710"
  },
  {
    "path": ".gitignore",
    "mode": "100644",
    "type": "blob",
    "size": 5,
    "sha": "4c49bd78f1d08f2bc09fa0bd8191ed38b7dce5e3",
    "url": "http://zikuu-gitea-host/api/v1/repos/admin/world-eye/git/blobs/4c49bd78f1d08f2bc09fa0bd8191ed38b7dce5e3"
  },
  {
    "path": "APIDOC.md",
    "mode": "100644",
    "type": "blob",
    "size": 844,
    "sha": "fac42e6bb64012c0305dd47118f871368a2c52a9",
    "url": "http://zikuu-gitea-host/api/v1/repos/admin/world-eye/git/blobs/fac42e6bb64012c0305dd47118f871368a2c52a9"
  },
  {
    "path": "ARCHITECTURE.md",
    "mode": "100644",
    "type": "blob",
    "size": 654,
    "sha": "bc80014bde6eadb0611cb5cdd5d066b1dac90038",
    "url": "http://zikuu-gitea-host/api/v1/repos/admin/world-eye/git/blobs/bc80014bde6eadb0611cb5cdd5d066b1dac90038"
  },
  {
    "path": "README.md",
    "mode": "100644",
    "type": "blob",
    "size": 579,
    "sha": "301c71bf07d44ffb9e44074cc6ae1c7ec342cfd",

```

Repository Protocolä•«æ°?æ? ä•?ã•?ã??ã?jã?ã?«ã•«ã-ã?•æ??èj?è!ç´ ä?•ã?çã?¼ã?ã??ã?-ã?•  
ã?£ã?¼ã?•APIã??è\$£èè-ã?ã??æ??æ?\_ã??ã?•ã?½ã??ã??ã?!ã?§ã?çã•®ç?®ç??ã??è²-ã??ã?•ã®?è£  
æ\_?ã?»ã®?è£ ä\_ã?»ã®?è£ ä\_ç??ã•ã•©ã•®ç?¶æ ?ã?•ã?½ã?¼ã?'ã?³ã?¼ã??ã•®ã??ã?ªã?¼ã•"é?•è!  
ã??ã?jã?ã?«ã•ã•©ã•?è"è\_°ã?ã??ã•¾ã?ã??

ä\_?ã•«ã?•æ??èj?è\$£èè-ã?ã?çã?¼ã?ã??ã?-ã?ã?ã?£ã?¼ã?•APIã•«é?çã?ã??æ??æ?\_ã•®ã¾?ã??çª°ã•  
?ã•¾ã?ã??

# TECHDOC

## Purpose

本リポジトリは、世界中のニュースイベントを時系列・国別に可視化し、リアルタイムで情報を分析できる Web アプリケーションを提供。フロントエンド（React/Vite）とバックエンド（FastAPI）をコンテナ化し、外部の Pivot API と認証サーバーを組み合わせで構築しています。

## Problem

- 大量のニュースデータをリアルタイムで集計・可視化する仕組みが不足。
- 国別・トピック別に動的にフィルタリングする UI が揃っていない。
- 認証・認可を含めた統合システムが未整備。
- スケーラブルな API とフロントエンドの分離が求められる。

## Scope

### In Scope

- ニュースイベントの取得・集計（Pivot API 経由）。
- 国別ヒートマップと主要キーワード・トピックの表示。
- 認証機能（JWT ベース）とログインフォーム。
- Docker 化、CI/CD パイプラインの構築。

### Out of Scope

- データベーススキーマの詳細設計（現時点ではシンプルにしている）。
- モバイルアプリ（React Native 等）の開発。
- データの永続化ロジック（データローダーの実装）。

## Design Principles

分離と拡張性を重視し、外部サービス（Pivot API）を明確に分離。

# ARCHITECTURE

## Purpose

本ドキュメントは、リポジトリ全体の構造と動作の概要を示します。  
フロントエンド (React/Vite) とバックエンド (FastAPI) をコンテナ化し、  
外部の Pivot サービスと認証サーバーを利用する構成です。

## System Overview

- **クライアント (ブラウザ)**: Vite でビルドされた SPA を閲覧。
- **フロントエンド**: React アプリが REST API を呼び出し、国別ヒートマップや記事リストを表示。
- **バックエンド**: FastAPI により `/api/items`, `/api/analytics`, `/health` を公開。
- **Pivot Client**: 外部 Pivot API (`pivot_base_url`) ヘクエリを投げ、結果を加工。
- **データベース**: SQLAlchemy で管理 (現時点ではスキーマは未実装)。
- **認証サーバー**: Vite 環境変数 `VITE_AUTH_BASE_URL` で参照。

## Component Structure

### Frontend

- `src/` 以下に React コンポーネント、スタイル、`i18n`。
- Vite が開発サーバー (port 5173) を起動し、`/api` をバックエンドへプロキシ。

### Backend API

- `app/main.py`: FastAPI アプリケーション。
- `app/api/items.py`, `app/api/analytics.py`: ルーター。
- `app/services/pivot_client.py`: HTTPX で Pivot API へリクエスト。
- `app/services/country_mapper.py`: 国名 ↔ ISO3 変換とフィルタリングロジック。
- `app/db.py`: SQLAlchemy の `engine` と `SessionLocal` を定義。



