



制作する データ・サーバの全体像

表1 データ・サーバのコンポーネント

コンポーネント	内容
サーバ処理基本部 (FastAPI)	HTTPリクエスト処理, リクエスト・ルーティングなど
パス・リクエスト処理	URLアクセス・パスに応じて呼び出される処理でAPI実装の本体
共通ライブラリ	バックエンドへのアクセスを行うライブラリや, チャネル管理やイベント処理など独自機能実装
バックエンド・システム	データ保存などに利用するInfluxDB, MySQL, Ethereum, MinIO

センサ・データをInfluxDBに保存する処理を作成します。その前に、データ・サーバ全体について説明します。

こんなサーバ

● 実装方針…セキュリティやエラー処理は割愛

作成するデータ・サーバは、技術的な理解が主目的であるため、認証や暗号化などのセキュリティ対策やエラー処理の実装はほぼ行っていません。それらはもちろん重要ではありますが、特集で説明したいことの主題ではないので割愛しました。

また、実装についても効率というよりは、処理の分

かりやすさを最優先しているため、実処理プログラムではオーバヘッドが大きく、仕事であれば採用しないような記述となっている部分もあります。

● データ・サーバの構造

データ・サーバはFastAPIフレームワークを使って開発します。FastAPIはプログラムの骨格やHTTPリクエスト処理といった部分を担ってくれますが、それ以外にもさまざまなコンポーネントがあります。データ・サーバは、おおまかに表1、図1のコンポーネントで構成されています。

表1の中で、サーバ処理基本部はFastAPIがカバーしてくれるのと、バックエンド・システムはオープンソース・ソフトウェアを利用しているため、パス・リクエスト処理(API処理)と共通ライブラリの部分を実装してデータ・サーバを構築します。

表2に、データ・サーバの配布ファイルに含まれるファイルを示します。ここに含まれているファイルでデータ・サーバの処理が実現されています。

● データ・サーバAPI概要

データ・サーバでは、さまざまなAPIを搭載する予定です。表3にAPI一覧を載せます。

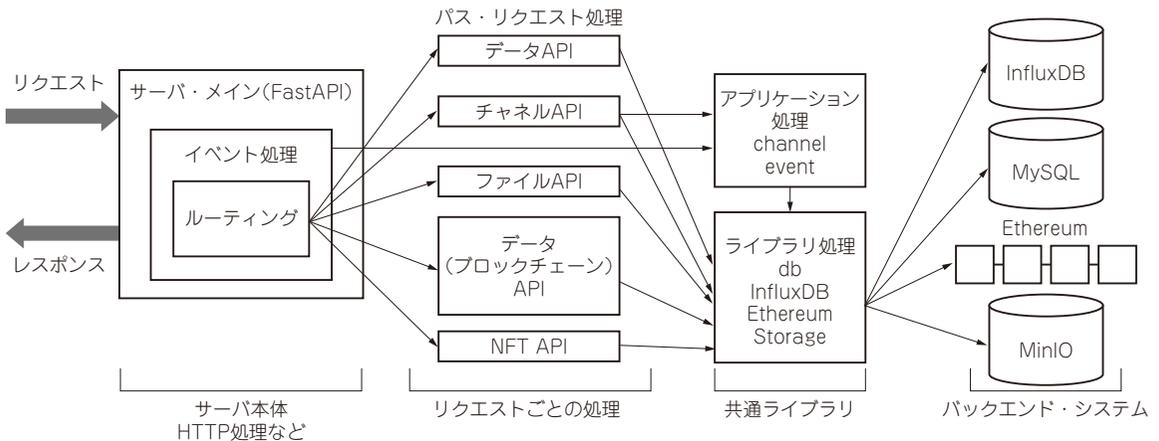


図1 データ・サーバのコンポーネント概要