



データ可視化画面 / チャンネル管理画面を作る

土屋 健

前章まででユーザが利用する機能として、

- データの保存と取り出し
- チャンネル管理機能

を作りました。データの保存は環境データを取得する装置が行うので、直接、ユーザが操作するものではありませんが、データの取り出しとチャンネル管理は必要時にユーザが行います。

それらの機能はREST APIで操作が可能ですが、人が使うには画面があった方がよいと思いますので、ここでは、

- 1) データ可視化画面 (データ取り出し)
- 2) チャンネル管理画面

を作成します。画面はウェブ・ブラウザで操作できるウェブ・アプリとして実現します。

ウェブ・アプリの構成

FastAPIの静的コンテンツに対応し、なおかつウェブ・ブラウザからのリクエストに対応する画面表示の実装は、第2部第2章のリスト1(mydata_server.py)を参照してください。なお、本章のリストは右上のQRコードから閲覧できます。

FastAPIの静的コンテンツ対応は91行目の、
`app.mount("/dataserver",
StaticFiles(directory="html"),
name="html")`
で行っています。ウェブ・ブラウザで
`http://<サーバアドレス>:<ポート>/
dataserver`
へアクセスした際に、htmlディレクトリ配下のファイルが結果として返されるようになります。htmlに

/dataserverがマッピングされているイメージです。そのため画面実装ファイルはhtmlディレクトリ配下に配置しています。ファイル構成と配置は第2部第2章の表2(p.59)を参照してください。

画面仕様

表1の画面を用意します。それぞれの画面イメージは図1～図3の通りです。

凝った画面ではありませんが、チャンネル管理やデータ表示の基本は押さえてあります。これをベースに拡張するとよいかと思います。

実装

画面はHTMLとJavaScriptで実装されています。画面を構成するファイルは、ソース一覧の中で分類が画面になっているものです。

ここでは、cssなどの表示形式に関する部分の説明は割愛し、画面表示や処理の本体であるHTMLファイルとJavaScript実装について説明します。

■ データ・サーバへのアクセス・ライブラリ

mylib.jsはデータ・サーバにアクセスして処理する表2の関数を提供するものです。リスト1に実装を示します。

● get_channel_list() (9～19行目)

チャンネルAPIを呼び出してチャンネル一覧を取得する処理です。

10行目でHTTPリクエストを送信する準備を行い

表1
画面仕様

画面	URL(http://サーバアドレス:ポート以降のパス)	機能
チャンネル管理	/dataserver/channel.html	チャンネル一覧表示、チャンネル・データの表示画面呼び出し、チャンネル削除、チャンネルの新規登録を行う画面
データ可視化	/dataserver/data.html?channel_id=チャンネルID	チャンネルに登録されているデータをグラフ表示する画面。measurementごとにグラフを作成する
メイン・メニュー	/dataserver/main.html	メイン画面で各種画面へのリンク。 今はチャンネル管理画面の呼び出しのみ