180625\_SIP\_web\_application\_structure

Htmlに関して、

1. Dashboardはfarm-sum.html
2. Planningはplannning.html
3. Profileはprofile.html
4. Aboutはabout.html
5. 左側の選択バーはnavi.html

これらは、sip-gui/public/template/index/配下に保存されている。

Weather scenario の更新アルゴリズム

1. Updateボタンを押すと、[sip-gui](https://github.com/listenfield/sip-gui/tree/951434aa47eb2af1828daa788930935c0778aa4d)/[public](https://github.com/listenfield/sip-gui/tree/951434aa47eb2af1828daa788930935c0778aa4d/public)/[js](https://github.com/listenfield/sip-gui/tree/951434aa47eb2af1828daa788930935c0778aa4d/public/js)/配下のagro\_ajax.js内の関数にアクセスする
2. updateWeatherScenarios関数が動作する
3. この関数内のsaveLastScenarioUpdateDate関数が作動する
4. saveLastScenarioUpdateDate関数はangularを利用している（参考ページurl⇨ <https://html5experts.jp/canidoweb/16732/>）
5. app.jsのapp.post(237行目)のputメソッドによってweather scenario を更新する。
6. この時、参考となるのがAPI-wgenディレクトリに存在するapplication.py (249行目)である。
7. さらに、application.pyの673行目に、どのように更新するかを記した関数が用意されている。
   1. Put メソッドで気象データを指定の日付まで更新
   2. update\_wth\_scenarios関数を呼び出す（更新の実行）
   3. get\_hist\_wth関数で指定した日までのnaro1kmデータを取得(request\_naro関数と協調)
   4. update\_wtde\_files関数で前に取得したnaro1kmデータを辞書型に変換（日付をインデックスとして、各気象データを抽出）し、該当箇所を更新する新しいファイルを作成
8. XMLHttpRequest()というサーバクライアント間通信を担保するオブジェクトを起動する。(https://qiita.com/sirone/items/412b2a171dccb11e1bb6)
   1. Onreadystatechange関数によって、通信が成功した時の機能を指示する。
   2. 通信が成功したら、putメソッドでデータを更新