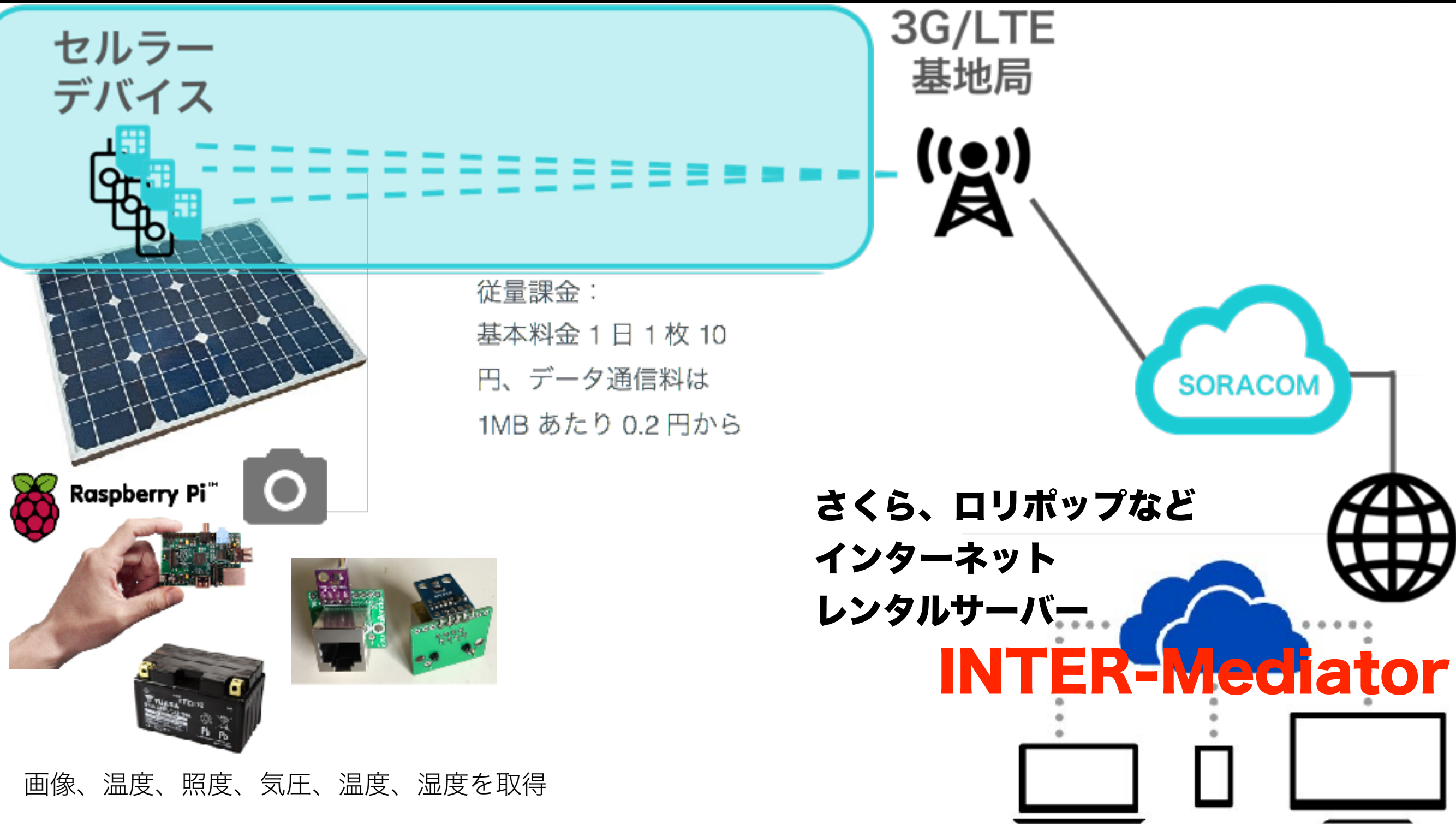


遠隔地からのIoT収集データを INTER-Mediatorで閲覧する

レンタルサーバーでインターネット公開
(具体例：さくら、ロリポップなど)

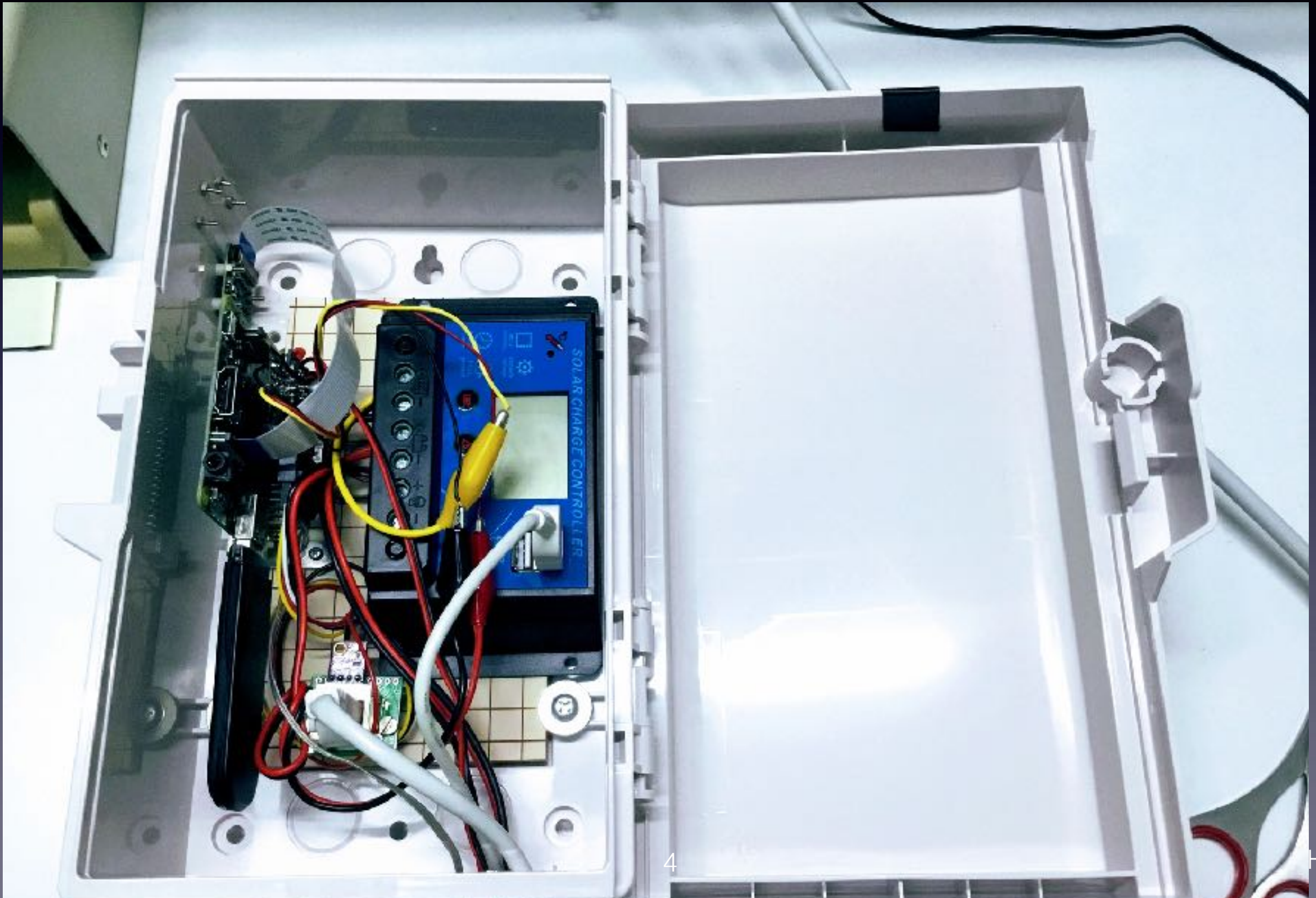


電源もネットワークもない場所から24時間データを送信



INTER-Mediatorを使った IoTデータ収集システムの全体像

無人の麦畑に放置するシステム



無人の麦畑に放置する

- 電力無し、ネット無し
- 曇りや雨の日が20日続くことも
- 通常は電源断・測定時のみ起動
- 動作時に新規アップデータを探索、取り込む

開発したもの

- プログラマブル電源スイッチ これは「ラズパイ用電源制御基板」 (<https://jiwashin.blogspot.jp/2016/07/power-controller-for-raspberry-pi.html>)を使っています。倉橋浩一さんに感謝
- 温度・湿度・気圧・照度センサーコントロール、カメラ画像取得
- サーバーへの送信プログラム
- サーバー側データ受信プログラム
- サーバーデータ閲覧プログラム

INTER-Mediatorが活躍

サーバーは皆違う

- PHP、Apache、Python、SSH、MySQLなど使えるが「使わせ方が違う」（涙）
- データベースの命名規則が違う
- ロリポップではテーブル名に英数半角のみ

	エコノミー	ライト 現在のプランです	スタンダード おすすめプラン	エンタープライズ
月額利用料	¥100	¥250～	¥500～	¥2000～
ディスク容量	10 GB	50 GB	120 GB	400 GB
メール容量	1048576 KB	2097152 KB	2097152 KB	2097152 KB
データベース (※2)	-	1 個	30 個	100 個
転送量	40 GB/日	60 GB/日	100 GB/日	無制限
独自ドメイン登録数	20 件	50 件	100 件	無制限
独自ドメインのメール アドレス登録数	20 件	無制限	無制限	無制限
ロリポップログ	1 個	3 個	5 個	5 個
7世代バックアップ	○ (※1)	○ (※1)	○ (※1)	○ (無料)
ホームページ作成 (ロ リポップ!スタジオ)	○ (※1)	○ (※1)	○ (※1)	○ (※1)
PHP (CGI版)	○	○	○	○
PHP (モジュール版)	-	-	○	○
共有SSL	-	○	○	○
独自SSL	-	○ (※1)	○ (※1)	○ (※1)
SSH	-	-	○	○
同時アクセス数拡張	-	-	○	○
電話サポート	-	-	○	○

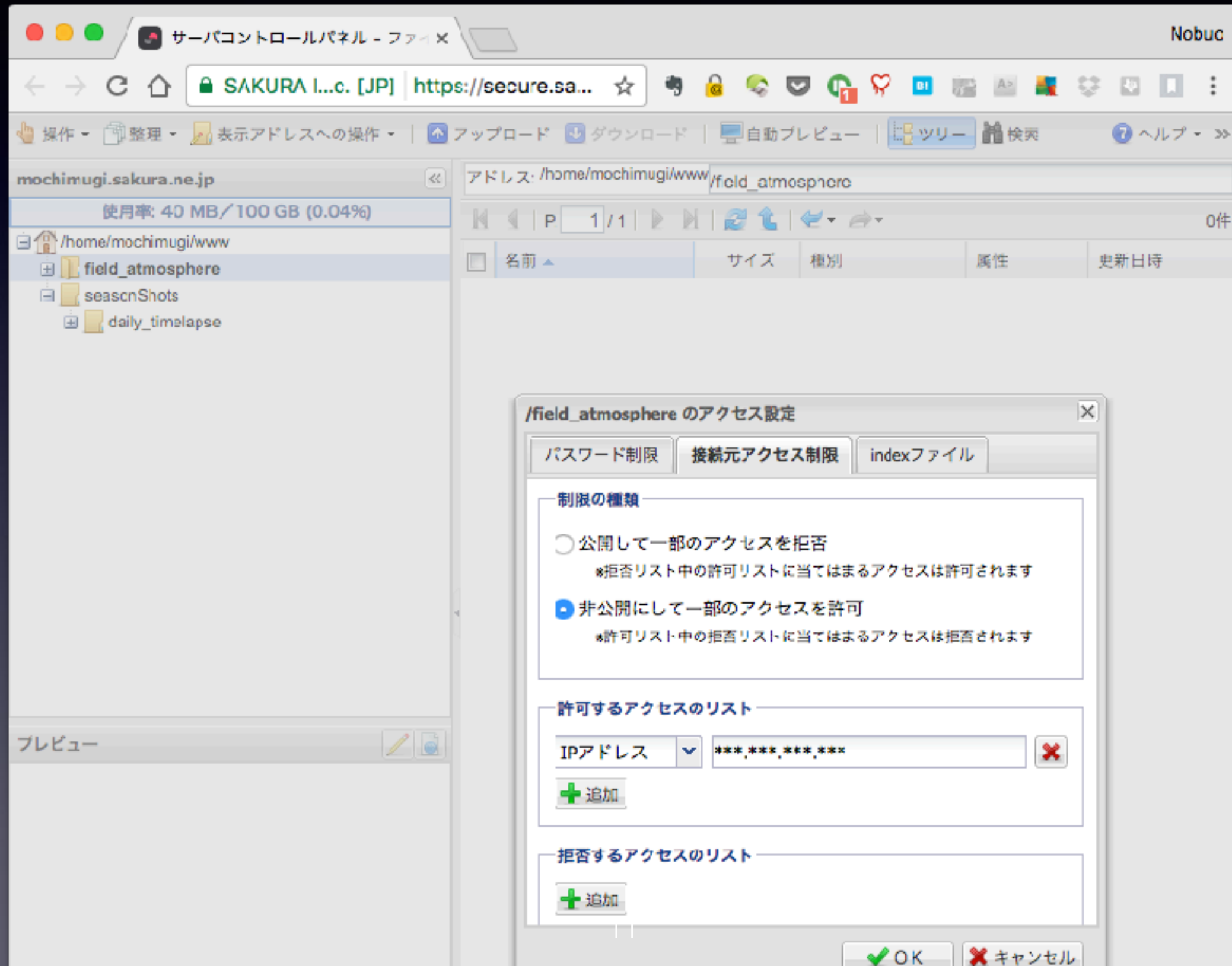
このあたりは必要
特にSSHは必須

開発と公開エリア

開発用エリアを用意

- INTER-Mediatorは設定ファイル、HTMLエディタを内包している
- IPアドレスによるアクセス制限をかける

IPアドレスによるアクセス制限



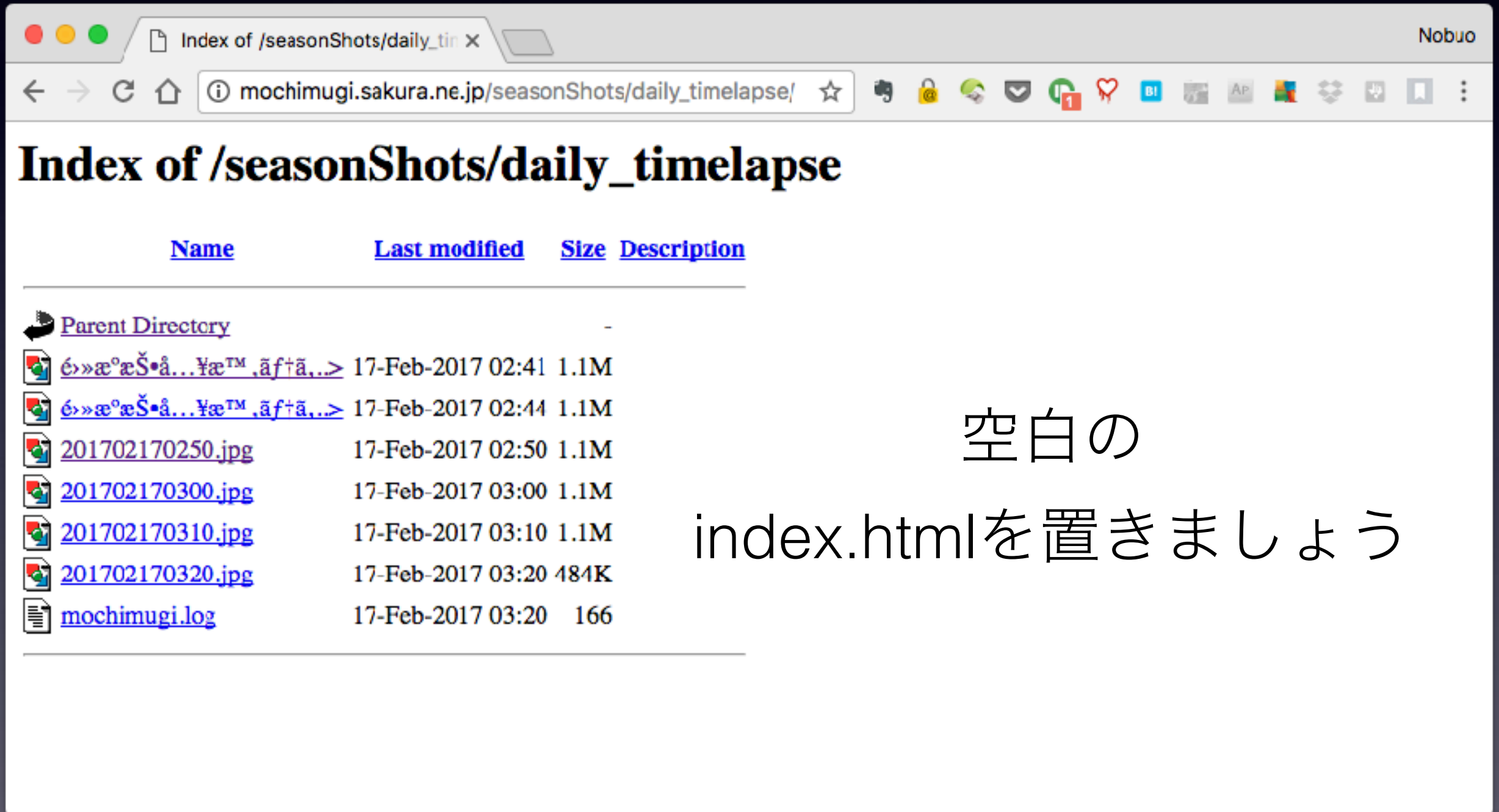
IPアドレスによるアクセス制限

```
DirectoryIndex index.html index.htm index.shtml index.php index.cgi .ht  
AuthUserFile /home/mochimugi/www/.htpasswd  
AuthType Basic  
AuthName "Web access"  
Satisfy all  
Order allow,deny  
Allow from 221.***.***.*
```

.htaccess が生成される

ロリポップの場合は、これを手動で作成

ファイルリスト表示を抑制



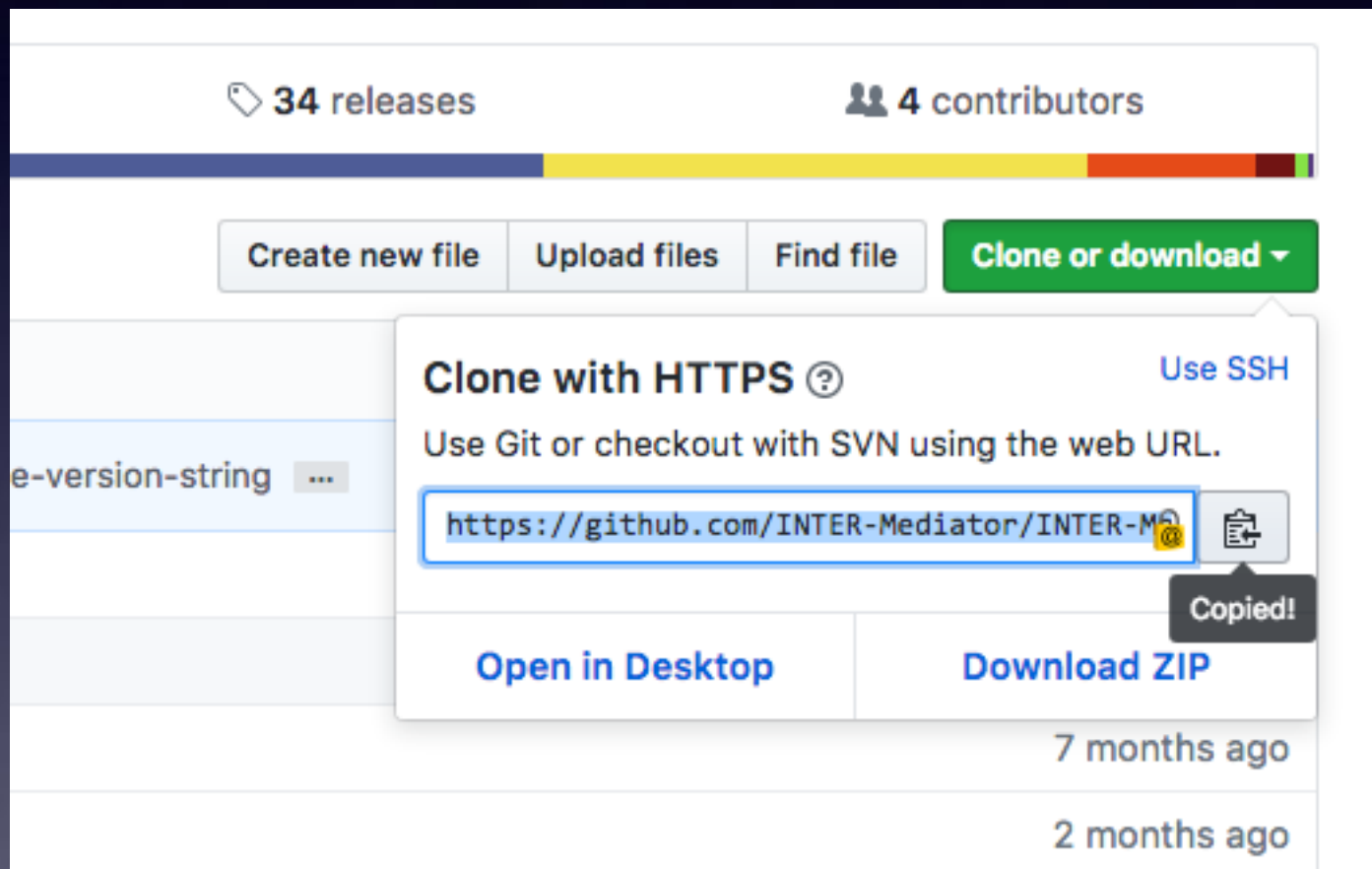
Index of /seasonShots/daily_timelapse

Name	Last modified	Size	Description
Parent Directory	-	-	-
é»æ°æŠ•â...¥æ™,ãf†ã,...>	17-Feb-2017 02:41	1.1M	
é»æ°æŠ•â...¥æ™,ãf†ã,...>	17-Feb-2017 02:44	1.1M	
201702170250.jpg	17-Feb-2017 02:50	1.1M	
201702170300.jpg	17-Feb-2017 03:00	1.1M	
201702170310.jpg	17-Feb-2017 03:10	1.1M	
201702170320.jpg	17-Feb-2017 03:20	484K	
mochimugi.log	17-Feb-2017 03:20	166	

空白の index.htmlを置きましょう

秘匿エリアにIMをダウンロード

1. `$ cd ~/web/IM/dev`
2. WebブラウザでINTER-Mediatorのリポジトリを表示させる
<https://github.com/INTER-Mediator/INTER-Mediator/tree/stable>
3. リポジトリのURLをクリップボード取得



4. SSH画面でリポジトリをクローン
`$ git clone https://github.com/INTER-Mediator/INTER-Mediator.git`

```
remote: Total 17367 (delta 0), reused 0 (delta 0), pack-reused 17367
Receiving objects: 100% (17367/17367), 16.32 MiB | 13.38 MiB/s, done.
Resolving deltas: 100% (11568/11568), done.
```

数秒で完了する。

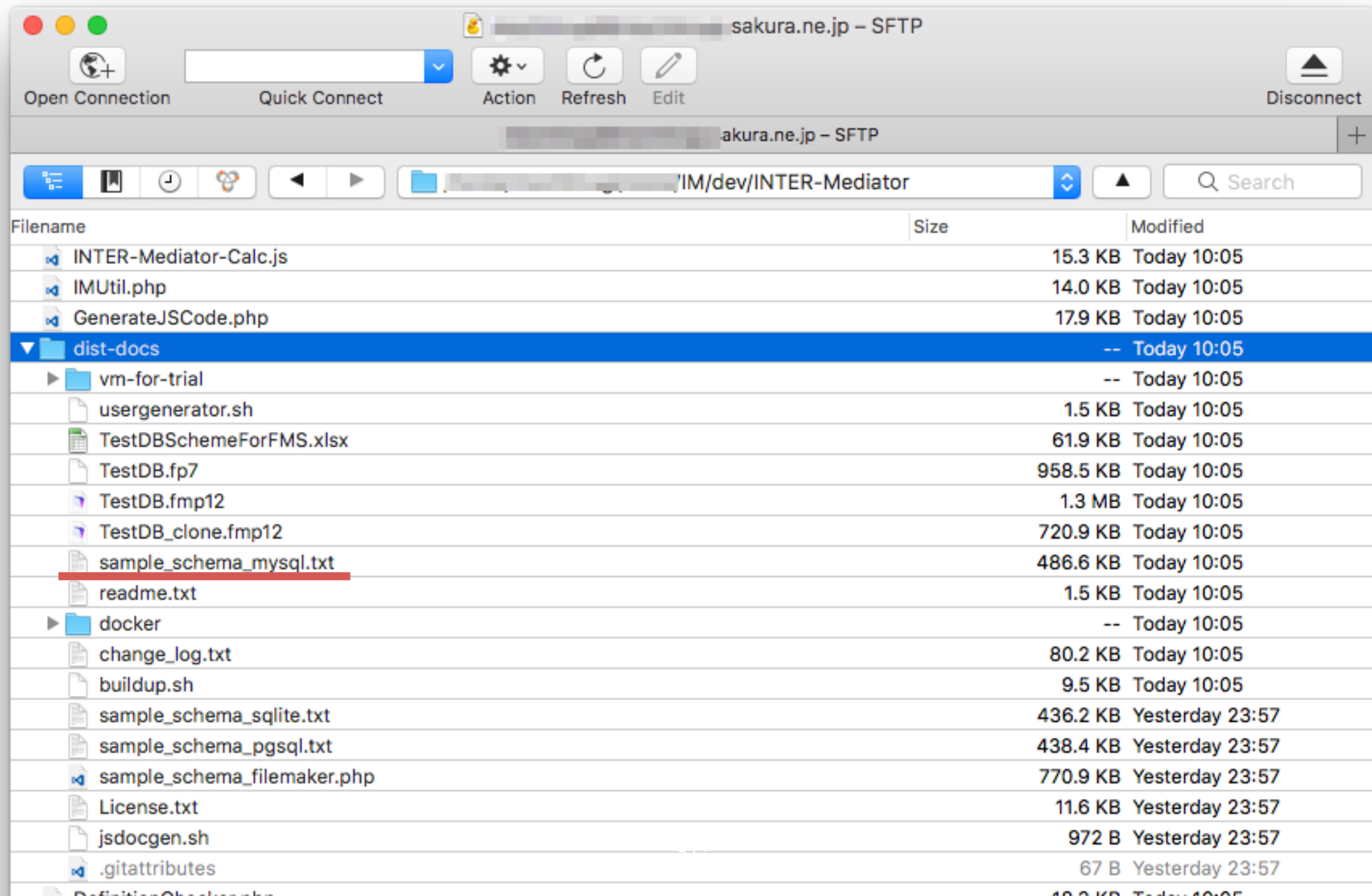
5. `stable (IM5.6)` にセット
`git checkout stable`

IMサンプルを動かす データベースの作成

* データベースの新規作成 *		
データベース バージョン	MySQL 5.5	
データベース ユーザ名 (初期アカウント名)		
データベース名	test_db	1文字以上 16文字以内
接続用パスワード	***** ユーザ名とパスワードは設定済みです。	8文字以上 32文字以内
データベース 文字コード	UTF-8	
データベースを作成する		

IMサンプルを動かす

IMで用意されたSQLスキーマでデータを生成





データベース

mochimugi_test_db (30)

mochimugi_test_db (30)

asset
authcor
authgroup
authuser
category
chat
contact
contact_kind
contact_way
cor_way_kind
cor_way_kindname
customer
fileupload
history
information
invoice
issuedhash
item
item_display
item_master
person
postalcode
product
registeredcontext
registeredpks
rent
saleslog
staff
survey
testtable

asset

[表示](#) [構造](#) [SQL](#) [検索](#) [挿入](#) [エクスポート](#) [インポート](#) [操作](#) [空にする](#) [削除](#)表示中の列 0 - 18 (~19¹ 合計, クエリの実行時間 0.0095 秒)

```
SELECT *  
FROM `asset`  
LIMIT 0 , 30
```

☐ プロファイリング [編集] [EXPLAIN で確認]

表示: 30 開始行 0

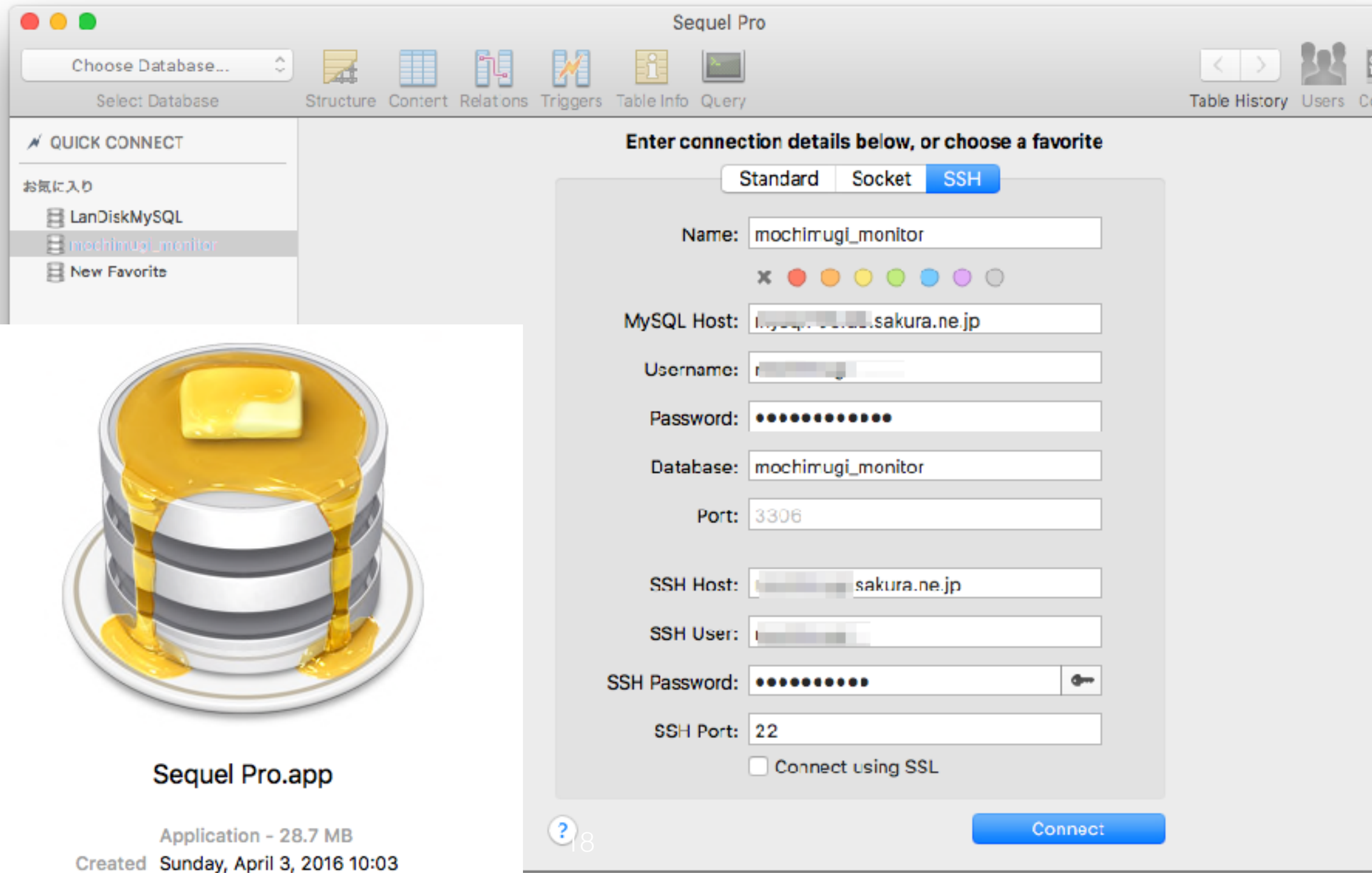
モード: 水平 (100 セルごとにヘッダを表示)

キーでソート: なし

+ オプション

			asset_id	name	category	manufacture	productinfo	purchase
☐			11	MacBook Air[1]	個人用	Apple	2012/250GB/4GB	2012-08-01
☐			12	MacBook Air[2]	個人用	Apple	2012/250GB/4GB	2012-08-01
☐			13	MacBook Air[3]	個人用	Apple	2012/250GB/4GB	2012-08-01
☐			14	VAIO type A[1]	個人用	ソニー	VGN-AR85S	2008-06-12
☐			15	VAIO type A[2]	個人用	ソニー	VGN-AR85S	2008-06-12
☐			16	プロジェクタ	共用	エプソン	EB-460T	2010-11-23
☐			17	ホワイトボード[1]	共用	不明	不明	NULL
☐			18	ホワイトボード[2]	共用	不明	不明	NULL
☐			19	加湿器	共用	シャープ	プラズマクラスター加湿器	2011-12-02
☐			20	事務室エアコン	共用			NULL
☐			21	会議室エアコン	共用			NULL
☐			22	携帯電話ドコモ	共用	京セラ	P904i	2010-04-04
☐			23	携帯電話au	個人用	シャープ	SH001	2012-03-03
☐			24	携帯電話Softbank[1]	個人用	Apple	iPhone 5	2012-10-01
☐			25	携帯電話Softbank[2]	個人用	Apple	iPhone 5	2012-10-01

さくらサーバーではこれが使える



The screenshot shows the Sequel Pro application window. On the left, the 'QUICK CONNECT' sidebar lists 'LanDiskMySQL', 'mochimugi_monitor' (selected), and 'New Favorite'. The main panel is titled 'Enter connection details below, or choose a favorite' and has tabs for 'Standard', 'Socket', and 'SSH'. The 'SSH' tab is active, showing fields for 'Name' (mochimugi_monitor), 'MySQL Host' (mysql.sakura.ne.jp), 'Username' (redacted), 'Password' (masked), 'Database' (mochimugi_monitor), 'Port' (3306), 'SSH Host' (sakura.ne.jp), 'SSH User' (redacted), 'SSH Password' (masked), and 'SSH Port' (22). There is a 'Connect' button at the bottom right. Below the screenshot, the application name 'Sequel Pro.app' is displayed, along with its size 'Application - 28.7 MB' and creation date 'Created Sunday, April 3, 2016 10:03'.

Sequel Pro

Choose Database... Select Database Structure Content Relations Triggers Table Info Query Table History Users

QUICK CONNECT

お気に入り

- LanDiskMySQL
- mochimugi_monitor
- New Favorite

Enter connection details below, or choose a favorite

Standard Socket SSH

Name: mochimugi_monitor

MySQL Host: mysql.sakura.ne.jp

Username: [redacted]

Password: [masked]

Database: mochimugi_monitor

Port: 3306

SSH Host: sakura.ne.jp

SSH User: [redacted]

SSH Password: [masked]

SSH Port: 22

☐ Connect using SSL

Connect

Sequel Pro.app

Application - 28.7 MB

Created Sunday, April 3, 2016 10:03

Params.phpを編集

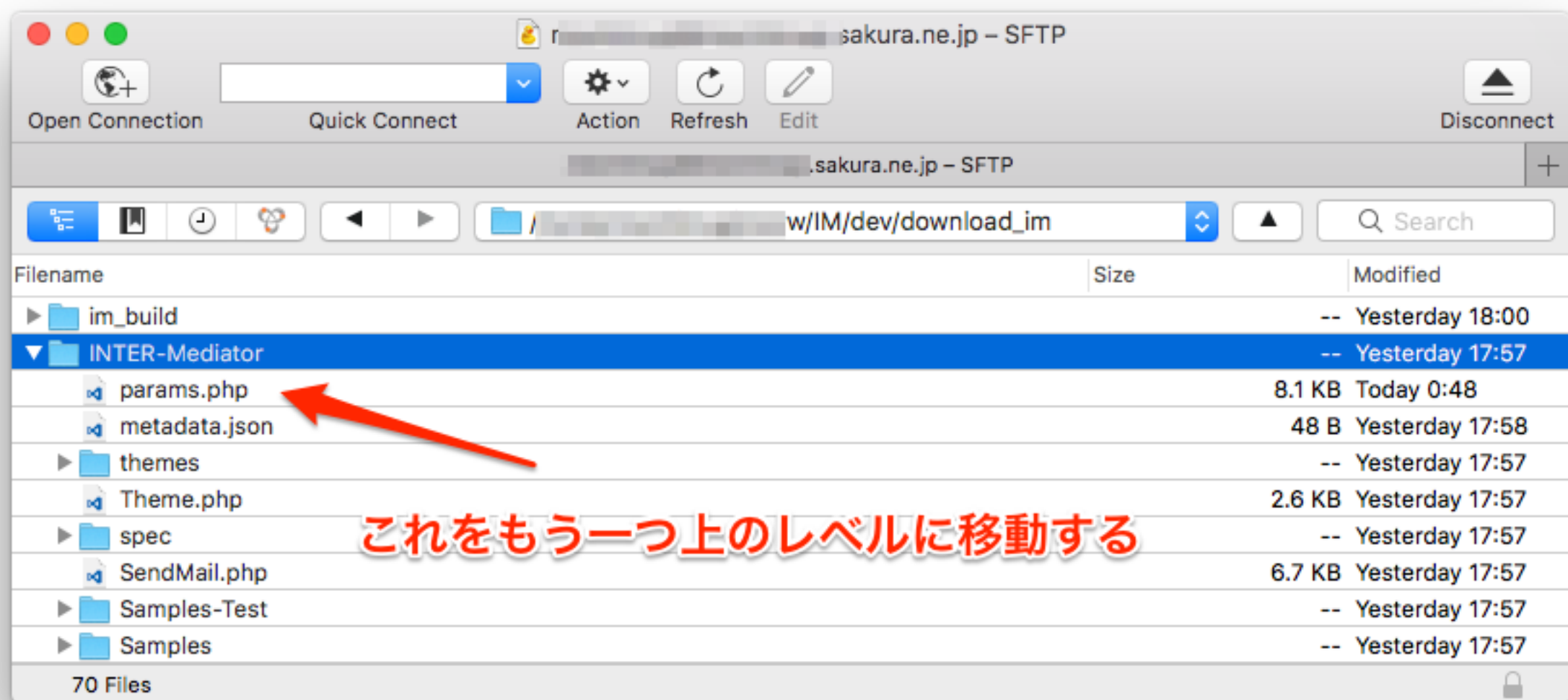
```
/*
 * database connection (PDO or FileMaker_FX)
 */
$dbClass = 'PDO';

/*
 * common settings for DB_FileMaker_FX and DB_PDO:
 */
$dbUser = 'UserName';
$dbPassword = 'mySQL_password';

/* DB_FileMaker_FX aware below:
 */
$dbServer = '127.0.0.1';
$dbPort = '80';
$dbDataType = 'FMPro12';
$dbDatabase = 'TestDB';
$dbProtocol = 'HTTP';

/* DB_PDO awares below:
 */
$dbDSN =
'mysql:host=yourDB.db.sakura.ne.jp;dbname=dbName_test_db;charset=utf8';
```


Params.phpを移動



サンプルが動き出す

The screenshot shows the INTER-Mediator web application interface. The left sidebar contains a menu with items like 'Practices' and 'Any Kinds of Sample'. The main content area displays a search results page titled '検索機能のサンプル (JavaScriptプログラムで実現)'. The search criteria are set to '郵便番号' (Postal Code) and the results are displayed in a table with columns: '郵便番号' (Postal Code), '都道府県' (Prefecture), '市区町村' (City/Town/Village), and '町域名' (Municipality Name). The table lists 16 results for postal codes in Chiyoda-ku, Tokyo, including entries for '以下に掲載がない場合' (Not listed below), '飯田橋' (Iidabashi), '一番町' (Ichibancho), '岩本町' (Iwamotocho), '内神田' (Uchikanda), '内幸町' (Uchisaiyamacho), '大手町 (次のビルを除く)' (Nishi-Shinjuku (excluding the following buildings)), '大手町 J Aビル (地階・階層不明)' (Nishi-Shinjuku J A Building (ground floor/unknown floor)), '大手町 J Aビル (1階)' (Nishi-Shinjuku J A Building (1st floor)), '大手町 J Aビル (2階)' (Nishi-Shinjuku J A Building (2nd floor)), '大手町 J Aビル (3階)' (Nishi-Shinjuku J A Building (3rd floor)), '大手町 J Aビル (4階)' (Nishi-Shinjuku J A Building (4th floor)), '大手町 J Aビル (5階)' (Nishi-Shinjuku J A Building (5th floor)), and '大手町 J Aビル (6階)' (Nishi-Shinjuku J A Building (6th floor)).

INTER-Mediator
For Database-driven Web Application

検索機能のサンプル (JavaScriptプログラムで実現)

検索条件: 表示件数: search

更新 レコード番号1-20 / 3654 << < > >> 1 ページ目へ

郵便番号 ▲ ▼	都道府県	市区町村	町域名 ▲ ▼
1000000	東京都	千代田区	以下に掲載がない場合
1020072	東京都	千代田区	飯田橋
1020082	東京都	千代田区	一番町
1010032	東京都	千代田区	岩本町
1010047	東京都	千代田区	内神田
1000011	東京都	千代田区	内幸町
1000004	東京都	千代田区	大手町 (次のビルを除く)
1006890	東京都	千代田区	大手町 J Aビル (地階・階層不明)
1006801	東京都	千代田区	大手町 J Aビル (1階)
1006802	東京都	千代田区	大手町 J Aビル (2階)
1006803	東京都	千代田区	大手町 J Aビル (3階)
1006804	東京都	千代田区	大手町 J Aビル (4階)
1006805	東京都	千代田区	大手町 J Aビル (5階)
1006806	東京都	千代田区	大手町 J Aビル (6階)

定義ファイルエディター

セキュリティを保つためにインストール直後は動かない設定になっている

<http://ホスト名/INTER-Mediator/INTER-Mediator-Support/defedit.html?target=../Page2.php>

The definition file editor doesn't work initially for security reasons.

If you want to use it, remove the comment before the IM_Entry function call of the defedit.php file.

You shouldn't these files to be able to access from any client.

The authentication is suitable way. One typical setting is to set the .htaccess file in the INTER-Mediator-Support folder.

定義ファイルエディタは、セキュリティ上の理由から既定値では機能しないようになっています。

機能させるためには、defedit.phpファイルのIM_Entry関数の前にあるコメントを消してください。

その場合は、何らかの方法で、認証をかけるようにすべきです。たとえば、INTER-Mediator-Supportフォルダ内に .htaccessファイルを配置して、Web認証を機能させるなどの手段が利用できます。

defedit.php, pageedit.phpの末尾にあるIM_Entryのコメントアウトを削除する。

定義ファイルエディター

注意！ アクセス制限がかけられない場合

インターネットからアクセスできる環境では
定義ファイルエディタは動作しないようにしておこう

じゃあどうするの？！

そのためにVMが用意されています。

VMを使って存分に開発を！！

センサーデータを讀込む

まずデータベースを用意

(MySQL 5.5.54-log) i [redacted] d/atmos_test SSH Connected

Select Database Structure Content Relations Triggers Table Info Query Table History Users Console

TABLES

- atmos
- atmos_test
- fieldcommentmeta
- fieldcomments
- fieldlinks
- fieldoptions
- fieldpostmeta
- fieldposts
- fieldterm_relationships
- fieldterm_taxonomy
- fieldtermmeta
- fieldterms
- fieldusermeta
- fieldusers

Field	Type	Length	Unsigned	Zerofill	Binary	Allow Null	Key	Default	Extra	Encoding	Collation	Comment
id	INT	11	☐	☐	☐	☐	PK		auto_increment			
date	DATETIME		☐	☐	☐	☐			None			
temp	FLOAT	8,4	☐	☐	☐	☐			None			
pressure	FLOAT	8,4	☐	☐	☐	☐			None			
humid	FLOAT	8,4	☐	☐	☐	☐			None			
lux	INT	11	☐	☐	☐	☐			None			
cpu	FLOAT	8,4	☐	☐	☐	☐			None			
sensor_temp	FLOAT	8,4	☐	☐	☐	☐			None			
v0	FLOAT	5,3	☐	☐	☐	☐			None			
v1	FLOAT	5,3	☐	☐	☐	☐			None			
memo	TEXT		☐	☐	☐	☒			None	UTF-8 Unicode	utf8mb4_general	
remark	TEXT		☐	☐	☐	☒			None	UTF-8 Unicode	utf8mb4_general	
loc	TEXT		☐	☐	☐	☒			None	UTF-8 Unicode	utf8mb4_general	

INDEXES

Non_unique	Key name	Seq_in_index	Column name	Collation	Cardinality	Sub part	Packed	Comment
0	PRIMARY	1	id	A	605	NULL	NULL	

TABLE INFORMATION

- created: 2017/08/20
- engine: InnoDB
- rows: 581
- size: 50.0 KB
- encoding: utf8mb4
- auto_increment: 918

センサーデータを讀込む

**Server side
受け取り用アプリ
を用意**

```
<?php
require_once(dirname(__FILE__) . '/../INTER-Mediator/INTER-Mediator.php');
spl_autoload_register('loadClass');

if (!isset($_GET["c"])) {
    echo json_encode(array("ERROR" => "No authentication code."));
    exit();
}
$authCode = "12345678asdfghj";//this is the key auth
if (isset($_GET["c"]) && $_GET["c"] != $authCode) {
    echo json_encode(array("ERROR" => "Invalid authentication code."));
    exit();
}

$aValue = mb_ereg_replace("/[^0-9]/", "", $_GET["date"]);
// $aValue = $_GET["date"];
$bValue = $_GET["temperature"];
$cValue = $_GET["pressure"];
$dValue = $_GET["humid"];
$eValue = $_GET["lux"];
$fValue = $_GET["temp"]; //data "temp" is cpu temperature
if ($aValue < 1) {
    echo json_encode(array("ERROR" => "Invalid Number."));
    exit();
}
$dbInstance = new DB_Proxy();
$dbInstance->ignoringPost();
$dbInstance->initialize(
    array(array('name' => 'atmos', 'key' => 'id',),),
    array(), array("db-class" => "PDO"), 2, "atmos");
$dbInstance->dbSettings->addValueWithField("date", $aValue);
$dbInstance->dbSettings->addValueWithField("temp", $bValue);
$dbInstance->dbSettings->addValueWithField("pressure", $cValue);
$dbInstance->dbSettings->addValueWithField("humid", $dValue);
$dbInstance->dbSettings->addValueWithField("lux", $eValue);
$dbInstance->dbSettings->addValueWithField("cpu", $fValue);
$dbInstance->processingRequest("create");
$pInfo = $dbInstance->getDatabaseResult();
$logInfo = $dbInstance->logger->getMessagesForJS();
echo json_encode(array("data"=>$pInfo,"log"=>$logInfo));
?>
```

センサーデータを讀込む

テストコマンドの例

```
$ curl 'https://YourServerID.sakura.ne.jp/IM/dev/webAPI/getDataAPI.php?c="12345678asdfghj"&date=2017-03-17%2002:36:54&temp=13.55&pressure=1003.5555&lux=324' | jq .
```

```
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %         0      0     0     0      0     0      0      0      0      0
100  717    0   717    0     0    2350      0 --:--:-- --:--:-- --:--:--   2358
{
  "data": [
    {
      "id": "1775",
      "date": "2017-03-17 02:36:54",
      "temp": "13.5500",
      "pressure": "1003.5555",
      "humid": "53.5555",
      "lux": "324",
      "cpu": "0.0000",
      "v0": "0.000",
      "v1": "0.000"
    }
  ]
}
```

センサーデータを送込む

```
#!/usr/bin/python
#coding: utf-8
```

```
import httplib, urllib
import time
import datetime # datetimeモジュールのインポート
import locale   # import文はどこに書いてもOK(可読性などの為、慣例でコードの始めの方)
```

```
def captureSensorData():
    #センサーからデータ収集するプログラムを実装
    #I2C、SPIなどを使ってデータキャプチャ
    return temperature, pressure, humid, lightLevel
```

```
# today()メソッドで現在日付・時刻のdatetime型データの変数を取得
d = datetime.datetime.today()
```

```
print 'データ取得時刻 == %s : %s\n' % (d, type(d)) # Microsecond( $10^{-6}$ sec)まで取得
```

```
#Calculate CPU temperature of Raspberry Pi in Degrees C
temp = int(open('/sys/class/thermal/thermal_zone0/temp').read()) / 1e3 # Get Raspberry Pi CPU temp
```

```
temperature, pressure, humid, lightLevel = captureSensorData()
```

```
#send date to さくらレンタルサーバー
```

```
params_IM = urllib.urlencode({'c': "1234567890ABCSDFGHIJK", 'date': str(d), 'temp': temp, 'temperature': temperature, 'pressure': pressure, 'humid': humid, 'lux': lightLevel })
```

```
if __name__ == '__main__':
    try:
```

```
        conn = httplib.HTTPSConnection("Your Server HostName")
        conn.request("GET", "/IM/webAPI/**/**/putDataAPI_withAuth.php?" + params_IM)
        print "connection requested"
        response = conn.getresponse()
        print response.status, response.reason
        data = response.read()
        print data
        conn.close()
```

```
    except:
```

```
        print "connection failed"
```

Raspberry pi上で動作する
Python プログラム

DBに入ったデータを閲覧

```

<body>
<input type="text" data-im="_@condition:atmos_list:date:*match*" />
  <button data-im="_@update:atmos_list@">検索</button>
  レコード数: <select data-im="_@limitnumber:atmos_list">
    <option value="5" selected="selected">5</option>
    <option value="10">10</option>
    <option value="20">20</option>
  </select>
<a href="https://ambidata.io/ch/channel.html?id=999">現地の気象データ／グラフ表示はこちら</a>
<a href="http://mochimugi.sakura.ne.jp/seasonShots/index_sandBox.html">ホームページ（写真表示など）はこちら</a>
  <div id="IM_NAVIGATOR"></div>
  <table>
    <thead>
      <tr><th></th><th>日付<span style="cursor: pointer" data-im="_@addorder:atmos_list@:date:asc"> [昇順] </span>
        <span style="cursor: pointer" data-im="_@addorder:atmos_list@:date:desc"> [降順] </span></th><th>温度 (°C) </th><th>CPU温度
        (°C) </th><th>気圧 (hPa) </th><th>湿度 (%) </th><th>明るさ lux<br> (5000以下は曇り) </th><th>電池 v</th><th>Solar</th><th>ケース温度</th><th>写真</th></tr>
    </thead>
    <tbody>
      <tr style="text-align:right">
        <td></td>
        <td data-im="atmos_list@date"></td>
        <td data-im="atmos_list@temp atmos_list@color_temp@style.color"></td>
        <td data-im="atmos_list@cpu"></td>
        <td data-im="atmos_list@pressure"></td>
        <td data-im="atmos_list@humid"></td>
        <td data-im="atmos_list@lux atmos_list@color_lux@style.color"></td>
        <td data-im="atmos_list@v0"></td>
        <td data-im="atmos_list@v1"></td>
        <td data-im="atmos_list@sensor_temp"></td>

```

DBに入ったデータを閲覧

モchimugi.sakura.ne.jp/IM/fin X Nobuo

モchimugi.sakura.ne.jp/IM/fin_build/sandBox.html ☆

検索 レコード数: 5 現地の気象データ/グラフ表示はこちら ホームページ (写真表示など) はこちら

レコード番号 1-5 / 581 最初へ < > 最後へ 1 ページ目へ

	日付 [昇順] [降順]	温度 (°C)	CPU温度 (°C)	気圧 (hPa)	湿度 (%)	明るさ lux (5000以下は曇り)	電池 v	Solar	ケース温度	写真
詳細	2017-07-08 17:00	36.98	52.46	1,004.02	41.70	1,131.00	9.16	9.39	46.3349	201707081700.jpg
詳細	2017-07-08 16:00	39.25	55.15	1,004.02	38.79	2,979.00	7.05	7.12	49.4746	201707081600.jpg
詳細	2017-07-08 15:00	41.33	56.22	1,003.98	35.86	5,073.00	4.96	5.11	53.7900	201707081500.jpg
詳細	2017-07-08 14:00	40.29	55.15	1,004.20	40.25	5,981.00	5.01	4.93	54.4719	201707081400.jpg
詳細	2017-07-08 13:00	35.65	50.31	1,004.33	50.78	6,921.00	4.67	4.65	48.2635	201707081300.jpg

ログ内容の詳細

```

Log cleared at: 201707081600
[___main___] 2017-07-08 16:00:57,165 INFO: sending data to さくらレンタルサーバー via
INTER-Mediator
[___main___] 2017-07-08 16:00:58,406 INFO: Connection status:200
[___main___] 2017-07-08 16:00:58,429 INFO: Status reason: OK
[___main___] 2017-07-08 16:00:58,446 INFO: sending powerControlCommand:sudo
/usr/sbin/i2cset -y 1 0x40 40 11
[___main___] 2017-07-08 16:01:03,741 INFO: I2C command success: 1 time retry. System
will powerdown
***** above-mentioned is previous log *****
2017-07-08T16:01:53 rc.local (rc.local:33:main) Started logging to :
/home/pi/Documents/mochimugi/daily_timelapseSandBox/mochimugi.log
***** rc.local ver. 1.2 更新: 2017/06/13 01:14 *****
2017-07-08T16:02:03 rc.local (rc.local:63:main) PROGRAM SWITCH is off. Now system start
normally
2017-07-08T16:02:04 rc.local (rc.local:130:main) crontab is off
2017-07-08T16:02:04 rc.local (rc.local:136:main) pop is up and running
remote      refid      st t when poll reach  delay  offset jitter
=====
=====

```

表示を工夫

- データ値により色を変える
- 桁数調整
- 異常値感知し、別プログラムをキック

DBに入ったデータを閲覧

データ値によって色を変える

item:Color

expression:if (lux > 5000, 'red', if(lux >500, 'blue', (if (lux >250, 'black', 'silver'))))

Calculations				
	field	color	expression	if (lux > 5000, 'red', if(lux >500, 'blue', (if (lux >250, 'black', 'silver'))))
	field		expression	if (temp >30, 'red', if (temp >10, 'green', 'blue'))

ここまでできたら
いよいよ公開

公開用IMのエンジンを作る

編集機能などを削ったIMを作る

デプロイの方法と注意事項

<http://inter-mediator.info/ja/for-novices/deploy.html#section006>

IM/INTER-Mediator/dist-docs/buildup.sh

を使ってデプロイ用のファイルをインストールする。インストールレベルは次の4種類。

Choose the build result from these:

- (1) Complete (everything contains)
- (2) Core only (the least set to work wep applications)
- (3) Core + Support (add Auth_Support and INTER-Mediator-Support)

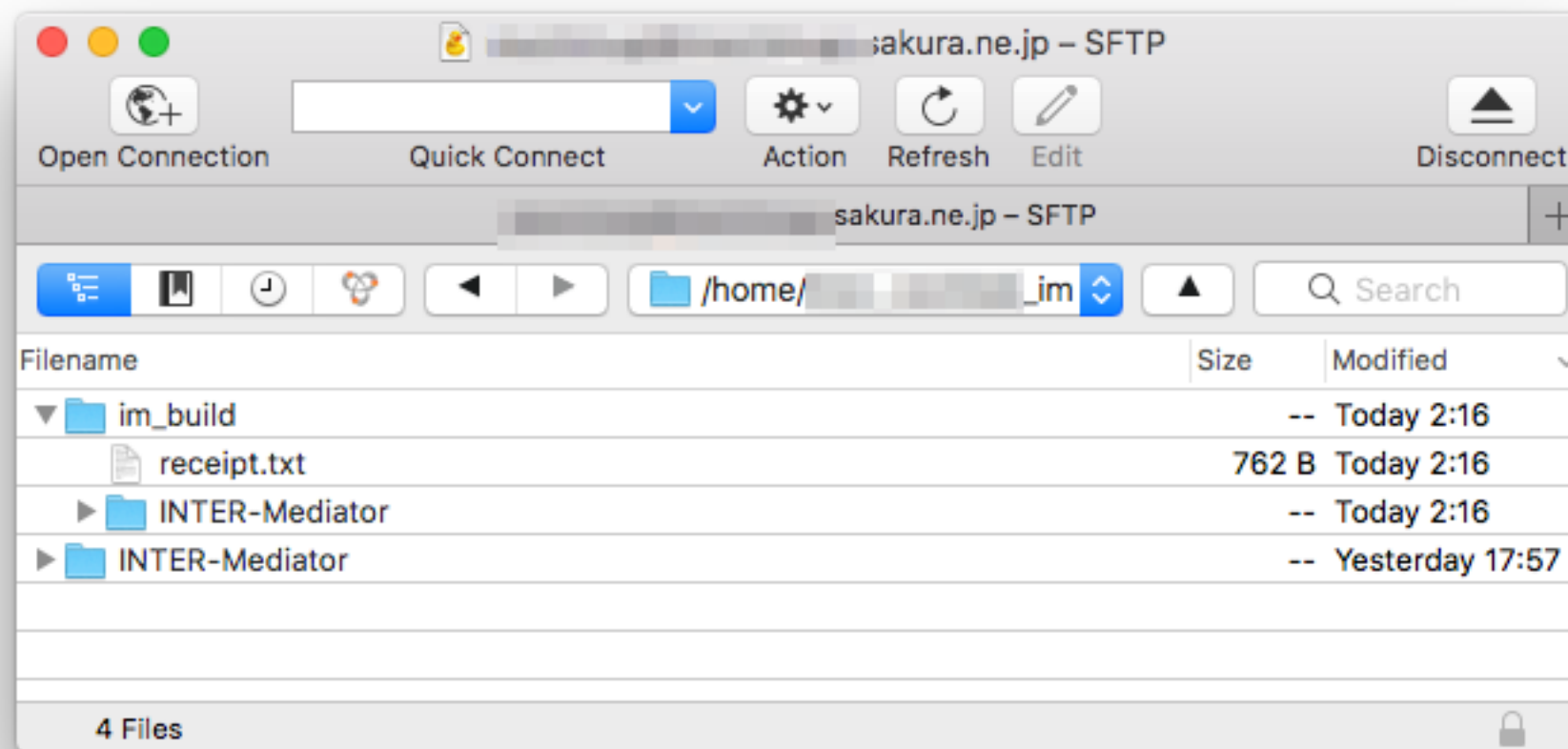
Type 1, 2 or 3, and then type return---->

公開サーバーでは2)を使うのがいい。

(3) は定義ファイルエディタを含むので、セキュリティには注意が必要。

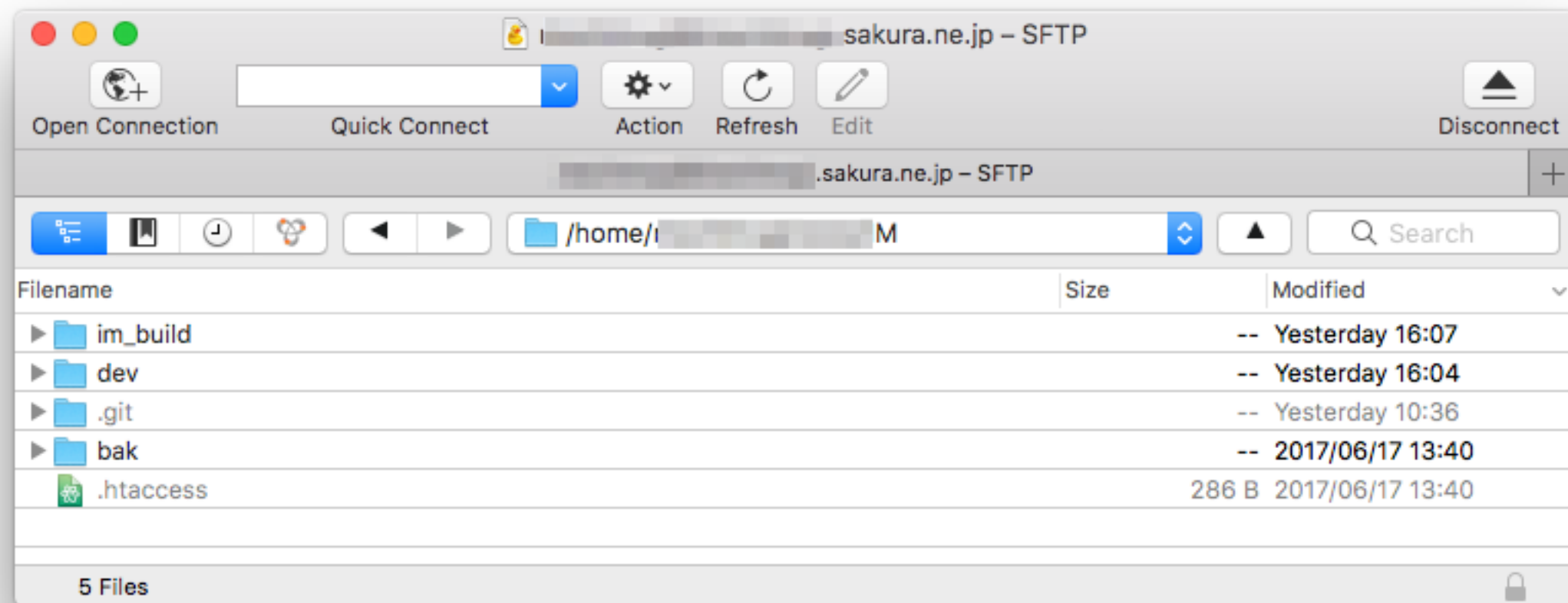
公開用IMのエンジンを作る

INTER-Mediatorと同一レベルにim_buildができる。
これを公開用エリアに配置する



公開用IMのエンジンを作る

devと同一レベルにim_buildを作る。
この中にアプリを配置



公開用IMのエンジンのセキュリティを確保

セキュリティを保つため

RSAの秘密鍵・公開鍵をparams.phpにセットする

これを設定しないと

Please change the value of \$generatedPrivateKey in params.php.

というIMのメッセージが出る

```
openssl genrsa -out gen.key 2048
```

を実行し、

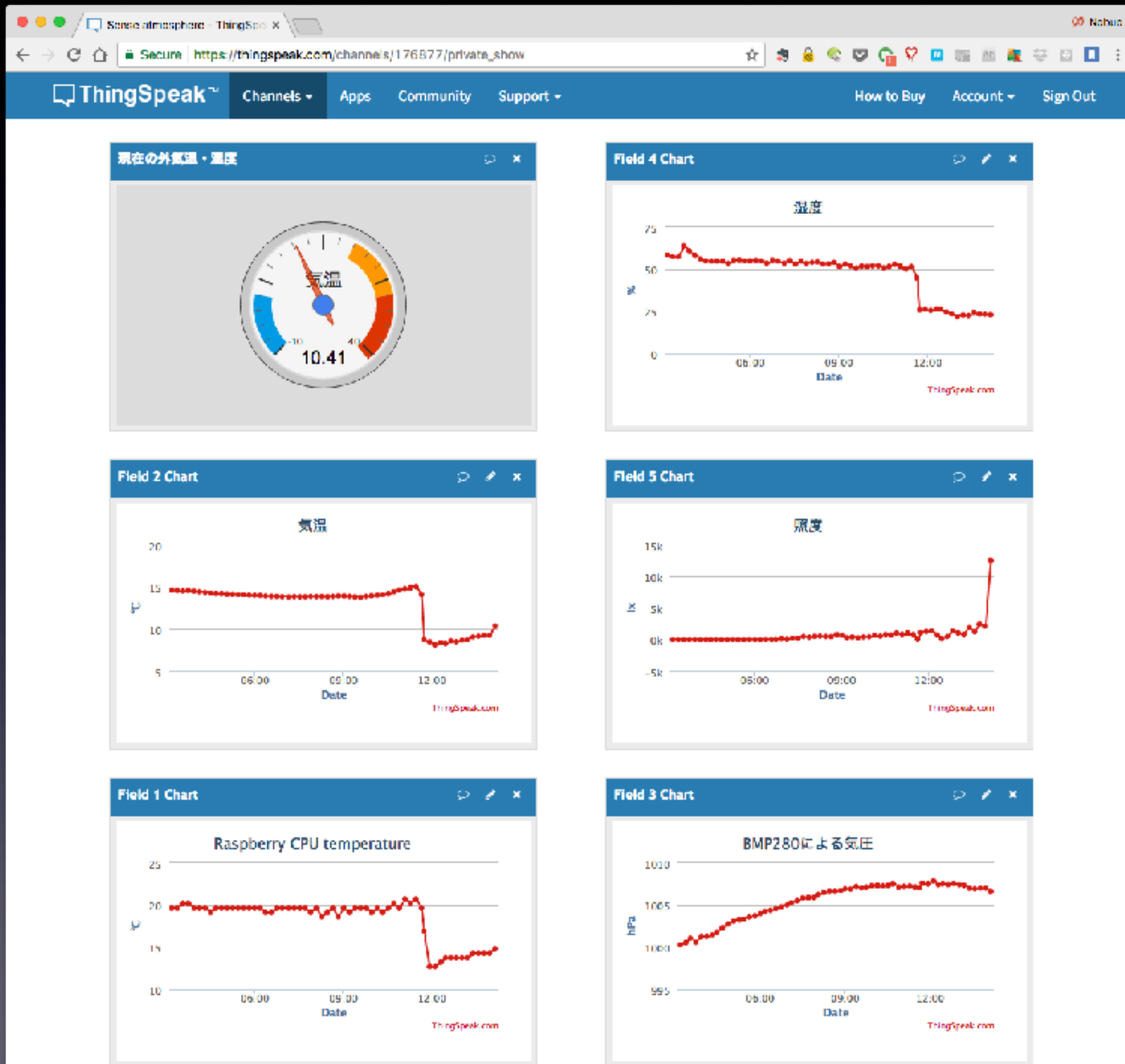
カレントディレクトリのgen.keyの内容を、params.phpの
該当部分に貼り付けるだけ

データのグラフ化

thingspeak.com

```
params = urllib.urlencode({'field1': temp, 'field2': temperature, 'field3': pressure,  
'field4': humid, 'field5': lightLevel, 'field6': voltage_ch1, 'field7': voltage_ch2,  
'key':key})  
headers = {"Content-type": "application/x-www-form-urlencoded", "Accept": "text/plain"}  
conn = httplib.HTTPConnection("api.thingspeak.com:80")
```

INTER-Mediator 《大》 勉強会 2017

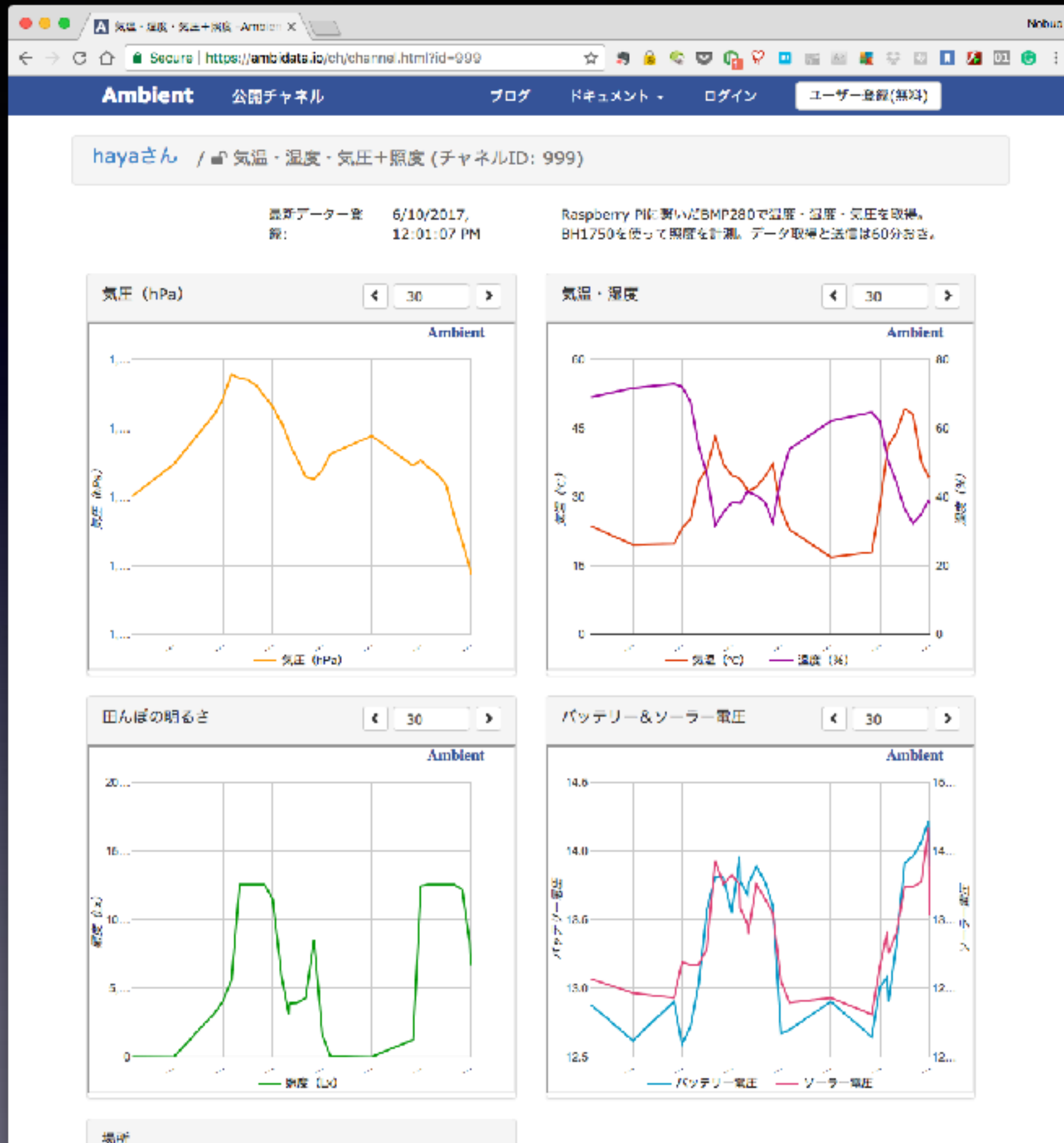


データのグラフ化

<https://ambidata.io/ch/channel.html?id=999>

```
params = urllib.urlencode({'field1': temp, 'field2': temperature, 'field3': pressure,  
'field4': humid, 'field5': lightLevel, 'field6': voltage_ch1, 'field7': voltage_ch2,  
'key':key})  
headers = {"Content-type": "application/x-www-form-urlencoded", "Accept": "text/plain"}  
conn = httplib.HTTPConnection("api.thingspeak.com:80")
```

INTER-Mediator 《大》勉強会 2017



オートアップデート

<https://www.evernote.com/l/AAUgukzoz5VJhLiXNJ1Pj1dzQSysGQTtHIU>

```
git pull git@github.com:yourRepository/  
sendAllDataPicture.git
```


完成

http://mochimugi.sakura.ne.jp/IM/im_build/sandBox.html

The screenshot shows a web browser window displaying a data collection interface. The address bar shows the URL: mochimugi.sakura.ne.jp/IM/im_build/sandBox.html. The page has a search bar and navigation links. Below the navigation bar is a table with 12 columns: 日付 [昇順] [降順], 温度 (°C), CPU温度 (°C), 気圧 (hPa), 湿度 (%), 明るさ lux (5000以下は曇り), 電池 v, Solar, ケース温度, and 写真. The table contains 5 rows of data for the date 2017-07-08. Below the table is a section titled 'ログ内容の詳細' (Log content details) which displays a log file. The log content includes messages about data sending, connection status, and system startup.

	日付 [昇順] [降順]	温度 (°C)	CPU温度 (°C)	気圧 (hPa)	湿度 (%)	明るさ lux (5000以下は曇り)	電池 v	Solar	ケース温度	写真
詳細	2017-07-08 17:00	36.98	52.46	1,004.02	41.70	1,131.00	9.16	9.39	46.3349	201707081700.jpg
詳細	2017-07-08 16:00	39.25	55.15	1,004.02	38.79	2,979.00	7.05	7.12	49.4746	201707081600.jpg
詳細	2017-07-08 15:00	41.33	56.22	1,003.98	35.86	5,073.00	4.96	5.11	53.7900	201707081500.jpg
詳細	2017-07-08 14:00	40.29	55.15	1,004.20	40.25	5,981.00	5.01	4.93	54.4719	201707081400.jpg
詳細	2017-07-08 13:00	35.65	50.31	1,004.33	50.78	6,921.00	4.67	4.65	48.2635	201707081300.jpg

ログ内容の詳細

```

Log cleared at: 201707081600
[___main___] 2017-07-08 16:00:57,165 INFO : sending data to さくらレンタルサーバー via
INTER-Mediator
[___main___] 2017-07-08 16:00:58,406 INFO : Connection status:200
[___main___] 2017-07-08 16:00:58,429 INFO : Status reason: OK
[___main___] 2017-07-08 16:00:58,446 INFO : sending powerControlCommand:sudo
/usr/sbin/i2cset -y 1 0x40 40 11 i
[___main___] 2017-07-08 16:01:03,741 INFO : I2C command success: 1 time retry. System
will powerdown
**** above-mentioned is previous log ****
2017-07-08T16:01:53 re.local (re.local:33:main) Started logging to :
/home/pi/Documents/mochimugi/daily_timelapseSandbox/mochimugi.log
**** re.local ver: 1.2 更新: 2017/06/13 01:14 ****
2017-07-08T16:02:03 re.local (re.local:33:main) PROGRAM SWITCH is off. Now system start
normally
2017-07-08T16:02:04 re.local (re.local:130:main) crontab is off
  
```

勉強方法

- サイト
- トレーニングコース（電子本）
- Facebook グループ
- Slack コミュニティ

どうもお疲れさま！！
さあ、ビールだ！