

参考見積募集要領

次のとおり参考見積を募集します。
令和7年2月17日

独立行政法人水資源機構
琵琶湖開発総合管理所長 石橋 一恭

1. 目 的

この参考見積の募集は、琵琶湖開発総合管理所で予定している業務の積算の参考とするため、参考見積りを募集するものです。

なお、この参考見積書をご提出いただいたことで、業務の指名又は競争参加資格をお約束するものではありませんので、予めご了承下さい。

見積条件について不明な点がございましたら、担当者にご連絡下さい。

2. 参考見積書提出の資格

- (1) 水資源機構における令和3・4・5・6 年度一般競争（指名競争）参加資格業者のうち、物品製造等の業種区分「ソフトウェア、情報処理、提供サービス」又は「試験・測量・測定・観測・監視機器」の認定を受けていることとします。
- (2) 営業に関し法律上必要とされる資格を有していることとします。
- (3) 水資源機構から「工事請負契約に係る指名停止等の措置要領」（平成6年5月31日付け6経契第443号）に基づき、淀川水系関連区域において指名停止を受けていないこととします。

3. 参考見積書の提出等

参考見積書は次に従い提出してください。

- (1) 参考見積書は、別紙-1「業務内容」に示す項目について必要な費用を別紙-2「LPWA水位計WEBサービス用API導入運用 見積書記入例」を参考に記載願います。
但し、様式は特に定めておりませんので自社の様式で提出頂いて構いません。
- (2) 消費税を除いた費用を記載願います。
- (3) 見積有効期限は「令和8年3月31日まで」として下さい。
なお、これによりがたい場合は、任意の期限とし、理由を明記下さい。
- (4) 見積の宛名は「独立行政法人水資源機構 琵琶湖開発総合管理所長 宛」として下さい。
- (5) 提出期間：令和7年2月25日（火）から令和7年3月4日（火）まで
持参する場合は、上記までの土曜日、日曜日及び祝日を除く毎日、9時から17時まで
- (6) 提出先：独立行政法人水資源機構 琵琶湖開発総合管理所長 宛
【担当】 管理課 村井（むらい）
〒520-0243 滋賀県大津市堅田2丁目1-10 TEL:077-574-0680
FAX:077-574-1739
- (7) 提出方法
書面は持参、郵送又はFAX（社印があること）により提出して下さい。

4. 募集要領に対する質問

この依頼書に対する質問がある場合においては、次に従い、書面(様式は自由)により提出してください。

(1) 提出期間:令和7年2月18日(火) から令和7年2月20日(木) まで

持参する場合は、上記までの土曜日、日曜日及び祝日を除く毎日、9時から17時まで

(2) 提出場所:3. (6)に同じ。

(3) 提出方法:3. (7)に同じ。

5. 質問に対する回答

質問に対する回答書は、次のとおり閲覧に供します。

(1) 閲覧期間:令和7年2月25日(火) から令和7年3月4日(火) まで

(2) 閲覧方法:ホームページに掲載します。

6. 参考見積書作成及び提出に要する費用

誠に勝手ながら見積提出者の負担とします。

7. ヒアリング

提出していただいた参考見積書についてヒアリングを実施することがあります。

－ 以 上 －

(別紙ー1)

業務内容

第 1 節 目的

本件は、内水排除支援等に資するために、機構がクラウドでデータを管理している LPWA 簡易水位計のデータについて WEB 表示のカスタマイズするために API 導入運用するものである。

第 2 節 納入場所

滋賀県大津市堅田 2-1-10
琵琶湖開発総合管理所

第 3 節 見積対象期間

令和 7 年 4 月 1 日から令和 8 年 3 月 31 日まで

第 4 節 WEB サービス用 API 導入運用

水資源機構がクラウドで運用している株式会社 farmo 製の LPWA 簡易水位計にかかるクラウドシステムを運用するウェブサービスに対して API を導入する。本 API は、ウェブサービスとして提供されるもので、以下に機能を外部から利用するためのインターフェースを提供するものとする。クライアントは HTTP リクエストを通じてデータの取得等ができるものとする。

4-1 ログイン API 機能

ログイン API は、ユーザーがウェブサービスにログインするためのインターフェースを提供するものとする。クライアントは HTTP リクエストを通じてユーザー名とパスワードを提供し、成功時にはアクセストークンを取得する。

4-2 センサーリスト取得 API 仕様

水位センサーの詳細情報を取得する API であり、以下の仕様とする。

① リクエストパラメータ

ログイン API で取得したファーモ ID
ログイン API で取得した clientSecret を利用

② レスポンスパラメータ

水位センサーの製品 ID
水位センサーを設置した場所名
水位センサーを設置した緯度
水位センサーを設置した経度
超音波センサーから地面までの高さ
水位センサーの基準水位

水位センサーの危険水位
水位センサーの警戒水位
水位センサーの最低水位
エラーの原因に関する情報

4-3 水位データ取得 API 仕様

測定した水位データを取得する API であり、以下の仕様とする。

① リクエストパラメータ

ログイン API で取得したファーム ID
ログイン API で取得した clientSecret を利用

② レスポンスパラメータ

水位センサーの製品 ID
測定日時(YYYYMMDDhhmm 形式)
水位(単位:cm):センサー測定範囲外であれば、異常値として返す。
基準水位との差分(単位:cm): 基準水位が登録されていない場合は、基準水位は 0cm として計算する。water_level が異常値であれば、異常値として返す
エラーの原因に関する情報

4-4 水位センサー登録 API 仕様

水位センサーを登録する API であり、以下の仕様とする。

① リクエストパラメータ

ログイン API で取得したファーム ID
ログイン API で取得した clientSecret を利用

② レスポンスパラメータ

水位センサーの製品 ID
登録結果(成功時 失敗時)
登録結果詳細(成功時 失敗時)
エラー原因に関する情報

4-5 LPWA 水位計 WEB サービス用 API 導入運用のための対象機器等

LPWA 簡易水位計:14個
使用月数:令和7年4月～令和8年3月(12ヶ月)

※LPWA 簡易水位計の数については、運用状況により変更する場合がある。

第 5 節 保守サービス等

本件については、電子メールや電話等により適時対応するものとする。

第 6 節 疑義

疑義が生じた場合は、速やかに機構と協議しなければならない。

以上

```
// @ts-nocheck
function routine_dev() {
  // 水位計のID情報
  // var spreadsheetId = "◆◆◆◆◆◆◆◆◆◆";
  var WATER_METERS = [
    { sheetName: '新川', productId: 'pc36y', baseWaterLevel: 90 },
    { sheetName: '外ヶ浜', productId: 'pdu61', baseWaterLevel: 110 },
    { sheetName: '菅沼', productId: 'phgnv', baseWaterLevel: 130 },
    { sheetName: '五反田', productId: 'ppyv7', baseWaterLevel: 120 },
    { sheetName: '永竜', productId: 'pe7d1', baseWaterLevel: 0 },
    { sheetName: '札立', productId: 'pmvrt', baseWaterLevel: 0 },
    { sheetName: '佃新川', productId: 'p774y', baseWaterLevel: 0 }
  ]
  // ... 他の水位計の情報を追加 ...
  // 10分に1回
  // ●ログインAPIを叩いて水位計データ取得のためのファームIDとclientSecretを取得
  var jsonResponse = login();
  var farmID = jsonResponse.farmo_id;
  var clientSecret = jsonResponse.client_secret;
  console.log(jsonResponse);

  // ●センサーリスト取得APIから各センサーの基準水位を取得
  var jsonResponse2 = getSensorList(farmID, clientSecret);
  console.log(jsonResponse2);
  for (var i = 0; i < WATER_METERS.length; i++) {
    var meter = WATER_METERS[i];
    var sensorInfo = jsonResponse2.sensor_list.find(s => s.sensor_id ===
meter.sensor_id);
    if (sensorInfo && sensorInfo.normal_water_level !== null) {
      meter.baseWaterLevel = sensorInfo.normal_water_level;
    }
  }

  // APIから取得したセンサーリストでWATER_METERSのbaseWaterLevelを更新
  WATER_METERS.forEach(meter => {
    var sensorInfo = jsonResponse2.sensor_list.find(s => s.sensor_id ===
meter.sensor_id);
    if (sensorInfo) {
      meter.baseWaterLevel = sensorInfo.normal_water_level;
    }
  });

  // ●現在時刻の取得
  var currentTime = new Date();
  console.log(currentTime);

  ///●以下、各水位計ごとに処理
  for (var i = 0; i < WATER_METERS.length; i++) {
    var spreadsheet = SpreadsheetApp.getActiveSpreadsheet();
    var sheet = spreadsheet.getSheetByName(WATER_METERS[i].sheetName);
    // 最下行の日時を取得
    var lastRow = sheet.getLastRow();
    var lastDateCell = sheet.getRange(lastRow, 2).getValue(); // 2列目に格納
    var lastDate = new Date(lastDateCell);
    // 現在時刻と最下行の日時との差を計算
    var timeDifference = (currentTime - lastDate) / 1000 / 60; // 差分を分単位
    で計算
    // 5分以上経過していればデータ取得
  }
}
```

```

        if (timeDifference >= 5) {
            // 水位計データ取得APIを叩いてデータを取得
            // 最下行の日時+1秒~現在時刻まで
            var data = fetchWaterLevelData(farmoID, clientSecret, lastDate,
currentTime, WATER_METERS[i].productId);
            console.log(WATER_METERS[i].sheetName);
            console.log(data);
            // data_listが空でない場合のみ、スプレッドシートの更新処理を実行
            if (data.data_list && data.data_list.length > 0) {
                ///各水位計のスプレッドに更新された分を記載
                appendToSheet(sheet, data, WATER_METERS[i].baseWaterLevel);
            }
        }
    }
};

function login() {
    //R5 API使用契約のURL、ログインIDとパスワード
    var url='https://api.farmo.tech/login'
    var headers = {
        "Content-Type": "application/json",
        "Accept-Charset": "UTF-8"
    };
    var payload = {
        "farmo_id": "uAvSRWmHXJv3",
        "farmo_pass": "9ns3kMg8egYu"
    };
    var options = {
        "method": "post",
        "headers": headers,
        "payload": JSON.stringify(payload),
        "muteHttpExceptions": true // エラーをミュートにして、レスポンスの内容を自分
でハンドルする
    };
    var response = UrIFetchApp.fetch(url, options);
    ///console.log(response);
    ///console.log(response.getResponseCode());
    ///console.log(response.getContentText());
    if (response.getResponseCode() == 200) {
        var jsonResponse = JSON.parse(response.getContentText());
        return jsonResponse;
    }
};

function getSensorList(farmoID, clientSecret) {
    var url = "https://api.farmo.tech/aqua/sensor";
    var headers = {
        "Content-Type": "application/json",
        "Accept-Charset": "UTF-8",
        "X-Farmo-Id": farmoID,
        "X-Farmo-Client-Secret": clientSecret
    };
    var options = {
        "method": "GET",
        "headers": headers
    };
    var response = UrIFetchApp.fetch(url, options);
    var jsonResponse = JSON.parse(response.getContentText());

```

```

    return jsonResponse; // APIのレスポンスに基づいて調整が必要
};

```

```

function fetchWaterLevelData(farmoID, clientSecret, lastDate, currentTime,
productId) {
    var baseUrl = 'https://api.farmo.tech/aqua/data';
    // 日時をAPIが期待する形式に変換
    function formatDate(date) {
        return Utilities.formatDate(date, "JST", "yyyyMMddHHmm");
    }
    var start = formatDate(new Date(lastDate.getTime() + 60000)); // 最後に取得し
    た日時の1分後
    var end = formatDate(currentTime);
    var url = baseUrl + '?sensor_id=' + productId + '&start=' + start + '&end=' +
end;
    console.log(url)
    var headers = {
        "Content-Type": "application/json",
        "Accept-Charset": "UTF-8",
        "X-Farmo-Id": farmoID,
        "X-Farmo-Client-Secret": clientSecret
    };
    var options = {
        "method": "get",
        "headers": headers,
        "muteHttpExceptions": true
    };
    var response = UrlFetchApp.fetch(url, options);
    if (response.getResponseCode() == 200) {
        var jsonResponse = JSON.parse(response.getContentText());
        return jsonResponse;
    } else {
        console.error("Error fetching water level data:",
response.getContentText());
        return null;
    }
}

```

```

function appendToSheet(sheet, data, baseWaterLevel) {
    var dataList = data.data_list;
    // 空の配列を作成して新しいデータ行を収集するためのバッファとして使用
    var rowsToAppend = [];
    // 各データポイントに対して
    for (var i = 0; i < dataList.length; i++) {
        var entry = dataList[i];
        // datetime を Google スプレッドシートの形式に変換します
        var dateTime = entry.datetime;
        var dateFormatted = dateTime.substring(0, 4) + '/' + dateTime.substring(4,
6) + '/' + dateTime.substring(6, 8);
        var timeFormatted = dateTime.substring(8, 10) + ':' + dateTime.substring(10,
12);
        // -----以下、water_levelからbaseWaterLevelを差し引くコード-----
        //var waterLevelValue;
        // entry.water_levelが数値かどうかをチェック
        //if (entry.water_level === null) {
        //    waterLevelValue = "異常値";
        //} else {

```

```

    // waterLevelValue = entry.water_level - baseWaterLevel;
    //}
    //-----
    // water_levelではなくwater_level_diffを使用
    var waterLevelValue = entry.water_level_diff !== null ?
entry.water_level_diff : "異常値";
    var row = [
        '', // 1列目は空白として残します
        dateFormatted + ' ' + timeFormatted,
        waterLevelValue
    ];
    rowsToAppend.push(row);
}
// シートに新しい行を追加します
sheet.getRange(sheet.getLastRow() + 1, 1, rowsToAppend.length,
3).setValues(rowsToAppend);
}

```

```
# coding: UTF-8
# 作成R5.2 作成者・松橋輝明
# 事前にGoogle Consoleで認証キーのjsonを取得
# 目的のスプレッドシートでjsonに記載のメールアドレスを権限許可しておく

import gspread
from datetime import datetime
import lxml
import re
import time
from bs4 import BeautifulSoup
from selenium import webdriver
from selenium.webdriver.chrome.options import Options

''' データ取得リスト設定'''
# スプレッドシートのファイル名「排水機場水位」
# 書き込むシート名、琵琶湖サイトのURL、琵琶湖サイトの列番号のリスト)
# 気象局は別ファイル
# 湖西管理所：針江(時間・累計)、入道沼(時間・累計)
# 湖北管理所：米原、大同川(同)
# 湖南管理所：野田、安治、津田江(同)
Datalist = [['雨量_湖西', 'http://biwako-mizukanri.jp/rain_w_h.html', '^ana0_',
'^ana1_', '^ana2_', '^ana3_'],
['雨量_湖北', 'http://biwako-mizukanri.jp/rain_n_h.html', '^ana6_',
'^ana7_', '^ana0_', '^ana1_'],
['雨量_湖南', 'http://biwako-mizukanri.jp/rain_s_h.html', '^ana6_',
'^ana7_', '^ana4_', '^ana5_', '^ana0_', '^ana1_']]

def GetRain(Data):
    gc =
gspread.service_account(filename='biwako-project-51269-a17c562706a5.json')

    '''1. スプレッドシートの最新の日時を入手する'''
    ws = gc.open('排水機場水位').worksheet(Data[0])
    # 最終行を取得したい列の列番号を指定する(例えば、A列の場合は1、B列の場合は2
    となる)
    column_number = 2
    last_row = len(ws.col_values(column_number))
    LatestDT_SS = ws.cell(last_row, column_number).value
    LatestDT_SS = datetime.strptime(LatestDT_SS, "%Y/%m/%d %H:%M:%S")
    print(LatestDT_SS)

    '''2. 琵琶湖サイト読み込み、最新日時を取得'''
    # たまに失敗するので繰り返し処理
    for trial in range(7):
        if trial == 0 or DTs_Act[0].get_text() == '':
            time.sleep(3)
            options = Options()
            options.add_argument('--headless') # ヘッドレスモードを設定
            driver = webdriver.Chrome(options=options)
            url = Data[1]
            driver.get(url)
            soup = BeautifulSoup(driver.page_source, 'lxml') # soupオブジェクト
            作成
            DTs_Act = soup.find_all('span', id=re.compile('^datestr_')) # 日時リ
            ストの取得
            Rain1s_Act = soup.find_all('span', id=re.compile(Data[2])) # 雨量リス
            トの取得
```

```

Rain2s_Act = soup.find_all('span', id=re.compile(Data[3])) #雨量リス
トの取得
Rain3s_Act = soup.find_all('span', id=re.compile(Data[4])) #雨量リス
トの取得
Rain4s_Act = soup.find_all('span', id=re.compile(Data[5])) #雨量リス
トの取得
# 3地点目がある場合
if len(Data) == 8:
    Rain5s_Act = soup.find_all('span', id=re.compile(Data[6])) #雨量
リストの取得
    Rain6s_Act = soup.find_all('span', id=re.compile(Data[7])) #雨量
リストの取得
else: #成功
    print(trial)
    break

'''3. 日付の比較をして条件分岐、雨量データの書き込み'''
# 備考：琵琶湖サイトの日付がすべて新しい場合は書き込まない設定（そこまで放置
しないという想定）
if datetime.strptime(DTs_Act[24].get_text(), "%Y-%m-%d %H:%M") <=
LatestDT_SS: #リスト最後尾を指定
    pass # 琵琶湖サイトの最新日付がスプレッドシートの最新日付より古い場合
→ 何もしない
else:
    # 琵琶湖サイトの日付がスプレッドシートの日付より新しい場合、スプレッドシ
ートを更新
    for i in range(25):
        DTi_Act = datetime.strptime(DTs_Act[25-1-i].get_text(), "%Y-%m-%d
%H:%M")
        if DTi_Act > LatestDT_SS:
            #print(DTi_Act, LatestDT_SS)
            pass # 琵琶湖サイトリストを遡っていく
        else:
            #琵琶湖サイトの日付がすでにスプレッドシートにあったタイミングで遡り
をやめ、シートに書き込み
            for ii in range(i):
                DTii_Act = datetime.strptime(DTs_Act[25-i+ii].get_text(),
"%Y-%m-%d %H:%M")
                Rain1ii_Act = Rain1s_Act[25-i+ii].get_text()
                Rain2ii_Act = Rain2s_Act[25-i+ii].get_text()
                Rain3ii_Act = Rain3s_Act[25-i+ii].get_text()
                Rain4ii_Act = Rain4s_Act[25-i+ii].get_text()
                ws.update_cell(last_row+1+ii, 2, DTii_Act.strftime("%Y/%m/%d
%H:%M:%S"))
                ws.update_cell(last_row+1+ii, 3, Rain1ii_Act)
                ws.update_cell(last_row+1+ii, 4, Rain2ii_Act)
                ws.update_cell(last_row+1+ii, 5, Rain3ii_Act)
                ws.update_cell(last_row+1+ii, 6, Rain4ii_Act)
            # 3地点目がある場合
            if len(Data) == 8:
                Rain5ii_Act = Rain5s_Act[25-i+ii].get_text()
                Rain6ii_Act = Rain6s_Act[25-i+ii].get_text()
                ws.update_cell(last_row+1+ii, 7, Rain5ii_Act)
                ws.update_cell(last_row+1+ii, 8, Rain6ii_Act)
            break
# Chromeドライバーを終了
driver.close()

```

```
#関数実行
for Data in Datalist:
    GetRain(Data)
```

```
# coding: UTF-8
# 作成R5.2 作成者・松橋輝明
# 事前にGoogle Consoleで認証キーのjsonを取得
# 目的のスプレッドシートでjsonに記載のメールアドレスを権限許可しておく

import gspread
from datetime import datetime
import lxml
import re
import time
from bs4 import BeautifulSoup
from selenium import webdriver
from selenium.webdriver.chrome.options import Options

''' データ取得リスト設定'''
# スプレッドシートのファイル名「排水機場水位」
# 書き込むシート名、琵琶湖サイトのURL、琵琶湖サイトの列番号のリスト)
# 気象局は別ファイル
# 湖西管理所：針江、入道沼
# 湖北管理所：米原、大同川
# 湖南管理所：野田、安治、津田江
Datalist = [['針江', "http://biwako-mizukanri.jp/water_w_h.html", '^ana0_',
'^ana1_'],
['入道沼', "http://biwako-mizukanri.jp/water_w_h.html", '^ana3_', '^ana4_'],
['米原', "http://biwako-mizukanri.jp/water_n1_h.html", '^ana13_', '^ana14_'],
['大同川', "http://biwako-mizukanri.jp/water_n1_h.html", '^ana0_', '^ana1_'],
['野田', "http://biwako-mizukanri.jp/water_s2_h.html", '^ana3_', '^ana4_'],
['安治', "http://biwako-mizukanri.jp/water_s2_h.html", '^ana0_', '^ana1_'],
['津田江', "http://biwako-mizukanri.jp/water_s1_h.html", '^ana0_', '^ana1_']]

def GetWL(Data):
    gc =
gspread.service_account(filename='biwako-project-51269-a17c562706a5.json')

    '''1. スプレッドシートの最新の日時を入手する'''
    ws = gc.open('排水機場水位').worksheet(Data[0])
    # 最終行を取得したい列の列番号を指定する（例えば、A列の場合は1、B列の場合は2
    となる）
    column_number = 2
    last_row = len(ws.col_values(column_number))
    LatestDT_SS = ws.cell(last_row, column_number).value
    LatestDT_SS = datetime.strptime(LatestDT_SS, "%Y/%m/%d %H:%M:%S")
    print(LatestDT_SS)

    '''2. 琵琶湖サイト読み込み、最新日時を取得'''
    # たまに失敗するので繰り返し処理
    for trial in range(7):
        if trial == 0 or DTs_Act[0].get_text() == '':
            time.sleep(3)
            options = Options()
            options.add_argument('--headless') # ヘッドレスモードを設定
            driver = webdriver.Chrome(options=options)
            url = Data[1]
            driver.get(url)
            soup = BeautifulSoup(driver.page_source, 'lxml') # soupオブジェクト
            を作成
            DTs_Act = soup.find_all('span', id=re.compile('^datestr_')) #日時リ
```

```

ストの取得      WL1s_Act = soup.find_all('span', id=re.compile(Data[2])) #内水位リス
トの取得      WL2s_Act = soup.find_all('span', id=re.compile(Data[3])) #外水位リス
トの取得
    else: #成功
        print(trial)
        break

''' 3. 日付の比較をして条件分岐、水位データの書き込み'''
if datetime.strptime(DTs_Act[24].get_text(), "%Y-%m-%d %H:%M") <=
LatestDT_SS: #リスト最後尾を指定
    pass # 琵琶湖サイトの最新日付がスプレッドシートの最新日付より古い場合
→ 何もしない
    else:
        # 琵琶湖サイトの日付がスプレッドシートの日付より新しい場合、スプレッドシ
        ートを更新
        for i in range(25):
            DTi_Act = datetime.strptime(DTs_Act[25-1-i].get_text(), "%Y-%m-%d
%H:%M")
            if DTi_Act > LatestDT_SS:
                #print(DTi_Act, LatestDT_SS)
                pass # 琵琶湖サイトリストを遡っていく
            else:
                #琵琶湖サイトの日付がすでにスプレッドシートにあったタイミングで遡り
                をやめ、シートに書き込み
                for ii in range(i):
                    DTii_Act = datetime.strptime(DTs_Act[25-i+ii].get_text(),
"%Y-%m-%d %H:%M")
                    WL1ii_Act = WL1s_Act[25-i+ii].get_text()
                    WL2ii_Act = WL2s_Act[25-i+ii].get_text()
                    ws.update_cell(last_row+1+ii, 2, DTii_Act.strftime("%Y/%m/%d
%H:%M:%S"))
                    ws.update_cell(last_row+1+ii, 3, WL1ii_Act)
                    ws.update_cell(last_row+1+ii, 4, WL2ii_Act)
                break
        # Chromeドライバーを終了
        driver.close()

#関数実行
for Data in Datalist:
    GetWL(Data)

```

独立行政法人 水資源機構
琵琶湖開発総合管理所長 殿

見 積 書

会 社 名	印
電 話 番 号	
FAX 番 号	
担 当 者 名	

見積有効期限は「令和8年3月31日まで」 (消費税相当額は除く)

名称	数量・単位	単価	備考
API利用料 月額料金	1式		※諸経費等含む

- ※特記事項
- ・見積書については、自社の様式で提出頂いて構いません。
 - ・月額料金は諸経費等含む費用を記載してください。
 - ・その他、注意事項・必要事項等がある場合は適宜記載してください。