```
REST API Usage instructions:


1) ALWAYS use *HTTPS*, never HTTP!


2) Login must be performed before obtaining data. To login, perform the following steps:

    POST your assigned username and password as user=XXXXXX and pass=YYYYY
      user=USERNAME
      pass=PASSWORD
    to the following URL: https://portal.c-lockinc.com/api/login

    This will return a JSON Web Token (JWT) similar to this:

eyJ0eXAiOiJKV1QiLCJhbGciOiJ1NiJ9.eyJ1c2VyjE3NiwiaWF0IjzNjQ0NDIxLCJleHAiOjE2MDM2NDQ3MjF9.MKGLirTX
HwRG6f-ChD8LuHI


    This JWT will be valid for 300 seconds to request data related to your account.


3) Once the JWT is obtained, use it for all subsequent requests.
   POST your token as parameter: token=TOKEN
     or use the conventional header:  "Authorization: Bearer TOKEN"
   Use one of the following requests to obtain your account data.


*** For an interactive version of the API Documentation, please see: https://portal.c-
lockinc.com/api/help/

*** For an Example of logging in using Postman see this link: https://portal.c-
lockinc.com/api/auth-example.png


##############################################################################






    RAW DATA REQUESTS
    #################

    System Types:
        GreenFeed, QuickFlux, SmartFeed, SmartFeed Pro, Super SmartFeed, SmartScale

    Parameters:
        d:      Data type (meas, secs, feed, rfid, cmds)
        fid:    GreenFeed/QuickFlux/SmartFeed/SmartScale System ID number
        st:     Start time of the data request (YYYY-MM-DD_HH:mm:ss or YYYY-MM-DD format)
        et:     End time of the data request (YYYY-MM-DD_HH:mm:ss or YYYY-MM-DD format)
        zip:    Optional (set to 1 to receive data as a ZIP file, otherwise data will be
uncompressed CSV)
        includeowner: Optional (set to 1 to include OwnerID field of each returned result)
        h:      Optional  0, 1, 2    (2 = include Parameters header and column headers, 1 =
include column headers, 0 = include no headers)

        fids:   use instead of fid, for requesting data from multiple systems at once
              In this case, all data returned will also include the SystemID for each row of
```

```
data
              For example: https://portal.c-lockinc.com/api/getraw?d=rfid&fid=74 ...
                               returns headers: DateTime,RFIDTag,InOrOut,TrayOrStall
                           https://portal.c-lockinc.com/api/getraw?d=meas&fids=74,400 ...
                               returns headers: SystemID,DateTime,RFIDTag,InOrOut,TrayOrStall



   * Get secondly measurement data (limited to 30 system-days worth of data at a time):
     https://portal.c-lockinc.com/api/getraw?d=meas&fid=74&st=2020-01-01_00:00:00&et=2020-01-20_23:59:59
        Headers: DateTime,Sensor1,Sensor2,Sensor3,...Sensor24
        * Get names of each data measurement column:
            https://portal.c-lockinc.com/api/getraw?d=vars&fid=74
                Headers: Sensor1Name,Sensor2Name,Sensor3Name,...Sensor24Name
        * Get equations of each data measurement column:
            https://portal.c-lockinc.com/api/getraw?d=equs&fid=74
                Headers: Sensor1Equation,Sensor2Equation,Sensor3Equation,...Sensor24Equation


   * Get number of seconds recorded (aggregated by hour) (limited to 30 days worth of data at a time):
     https://portal.c-lockinc.com/api/getraw?d=secs&fid=74&st=2020-01-01_00:00:00&et=2020-01-20_23:59:59
        Headers: Date,Hour,SecondCount

   * Get feed times - (only available/needed for "GreenFeed" systems)
     https://portal.c-lockinc.com/api/getraw?d=feed&fid=74&st=2020-01-01_00:00:00&et=2020-01-20_23:59:59
        Headers: DateTime,RFIDTag,CurrentCup,MaxCups,CurrentPeriod,MaxPeriods,DropInterval,PeriodInterval,FeedType

   * Get rfid scans
     https://portal.c-lockinc.com/api/getraw?d=rfid&fid=74&st=2020-01-01_00:00:00&et=2020-01-20_23:59:59
        Headers: DateTime,RFIDTag,InOrOut,TrayOrStall

   * Get commands
     https://portal.c-lockinc.com/api/getraw?d=cmds&fid=74&st=2020-01-01_00:00:00&et=2020-01-20_23:59:59
        Headers: DateTime,IssuedCommand





    OWNED SYSTEMS REQUESTS
    ######################

    System Types:
        GreenFeed, SmartFeed, SmartFeed Pro, Super SmartFeed, Super SmartFeed Producer, SmartScale

    Parameters:
        d:              Optional (return only a certain type of system)
        includeowner:   Optional (set to 1 to include OwnerID field of each system)
        h:              Optional  0, 1, 2    (2 = include Parameters header and column headers, 1 = include column headers, 0 = include no headers)


   * Get list of owned system IDs and type of system (GreenFeed, SmartFeed, SmartFeed Pro, Super
SmartFeed, Super SmartFeed Producer, SmartScale)
     https://portal.c-lockinc.com/api/getownedsystems?d=greenfeed
     https://portal.c-lockinc.com/api/getownedsystems?d=smartfeed
     https://portal.c-lockinc.com/api/getownedsystems?d=smartfeedpro
     https://portal.c-lockinc.com/api/getownedsystems?d=supersmartfeed
     https://portal.c-lockinc.com/api/getownedsystems?d=supersmartfeedproducer
     https://portal.c-lockinc.com/api/getownedsystems?d=smartscale
     https://portal.c-lockinc.com/api/getownedsystems?d=all
         Headers: (OwnerID,)SystemID,SystemType,Location




    PROCESSED EMISSION MEASUREMENTS REQUESTS
    ########################################

    System Types:
        GreenFeed, QuickFlux

    Parameters:
        d:              Data type (currently visits is the only possible value)
        fids:           GreenFeed or QuickFlux ID numbers, separated by commas
        st:             Start time of the data request (YYYY-MM-DD_HH:mm:ss or YYYY-MM-DD
format)
        et:             End time of the data request (YYYY-MM-DD_HH:mm:ss or YYYY-MM-DD format)
        zip:            Optional (set to 1 to receive data as a ZIP file, otherwise data will be
uncompressed CSV)
        includeowner:   Optional (set to 1 to include OwnerID field of each emission
measurement)
        type:           Optional (type=1 -> only finalized data, type=2 -> only preliminary
data, all other values -> return both types)
        ts:             Optional (set to a Date+Time to filter data processed after that time.
(i.e. ts=2023-03-03 to show only data processed on or after March 3rd, 2023)
        h:              Optional  0, 1, 2    (2 = include Parameters header and column headers,
1 = include column headers, 0 = include no headers)


   * Get individual visit emission data from GreenFeeds 74 and 110, May 1st, 2020, midnight
until noon:
     https://portal.c-lockinc.com/api/getemissions?d=visits&fids=74,110&st=2020-05-
01_00:00:00&et=2020-05-01_12:00:00
        Headers: (OwnerID,)FeederID,AnimalName,RFID,StartTime,EndTime,GoodDataDuration,
                 CO2GramsPerDay,CH4GramsPerDay,O2GramsPerDay,H2GramsPerDay,H2SGramsPerDay,
                 AirflowLitersPerSec,AirflowCf,WindSpeedMetersPerSec,WindDirDeg,WindCf,
                 WasInterrupted,InterruptingTags,TempPipeDegreesCelsius,IsPreliminary,RunTime






    PROCESSED EMISSION WORKBOOK REQUESTS
    ####################################

    System Types:
        GreenFeed, QuickFlux

    Parameters:
        d:              Directive (action) - may be one of the following:     list    status
download    delete    create
        wid:            Workbook ID, obtained from the list directive, or from the web interface
        fids:           GreenFeed or QuickFlux ID numbers
        h:              Optional  0, 1, 2    (2 = include Parameters header and column headers,
1 = include column headers, 0 = include no headers)

        sd:             Only for d=create - start date of workbook
        ed:             Only for d=create - end date of workbook


   * List all processed workbook:
     https://portal.c-lockinc.com/api/workbook?d=list
        Headers:
WorkbookID,SystemIDs,Filename,FileSize,AddedTime,StartDate,StopDate,Days,Status,Permanent

   * List processed workbook that contain GreenFeed #400:
     https://portal.c-lockinc.com/api/workbook?d=list&fids=400
        Headers:
WorkbookID,SystemIDs,Filename,FileSize,AddedTime,StartDate,StopDate,Days,Status,Permanent

   * Get status of a particular workbook (Workbook ID #4405):
     https://portal.c-lockinc.com/api/workbook?d=status&wid=4405
        Headers: WorkbookID,Status

   * Download a particular workbook (Workbook ID #4405):
     https://portal.c-lockinc.com/api/workbook?d=download&wid=4405
        Headers: NONE, data will be output as a .XSLX workbook. Download as a binary file.

   * Download a particular workbook (with a filename):
     https://portal.c-lockinc.com/api/workbook?
d=download&wid=0&fn=GreenFeed_Summarized_Data_400.xlsx
        Headers: NONE, data will be output as a .XSLX workbook. Download as a binary file.
        * Note: fn should only be used if the Workbook ID (wid) is 0. This will only apply to
daily generated files.
                Custom created workbooks will always have a Workbook ID and should be downloaded
with &wid=### parameter.

   * Create a new workbook consisting of GreenFeed 74, 400, and QuickFlux 5500, from Jan 1, 2024
until Jan 31, 2024:
     https://portal.c-lockinc.com/api/workbook?d=create&fids=74,400,5500&sd=2024-01-01&ed=2024-
01-31
        Headers: on success:  WorkbookID
                 on failure:  Error

   * Delete a workbook (Workbook ID #4405):
     https://portal.c-lockinc.com/api/workbook?d=delete&wid=4405
        Headers: NONE,
                 on success: will be message "4405 deleted."
                 on failure: Error





    ANIMAL WEIGHT REQUESTS
    ######################

    System Types:
        SmartScale, SmartFeed (Pro) with SmartScale Add-On

    Parameters:
        d:              Data type (daily, dailytab, or visits)
        fids:           SmartScale ID numbers, separated by commas
        st:             Start time of the data request (YYYY-MM-DD_HH:mm:ss or YYYY-MM-DD
format)
        et:             End time of the data request (YYYY-MM-DD_HH:mm:ss or YYYY-MM-DD format)
        full:           Report weights as estimated animal full-body weights, rather than half-body scale weights (for daily data only)
        zip:            Optional (set to 1 to receive data as a ZIP file, otherwise data will be uncompressed CSV)
        includeowner:   Optional (set to 1 to include OwnerID field of each weight measurement)
        h:              Optional  0, 1, 2    (2 = include Parameters header and column headers, 1 = include column headers, 0 = include no headers)


   * Get daily scale weights from SmartScales 1000001 and 1000027, May 1st - 20th, 2020:
     https://portal.c-lockinc.com/api/getweight?d=daily&fids=1000001,1000027&st=2020-05-01&et=2020-05-20&full=1
        Headers: RFID,Date,WeightKG

   * Same as above, but in a 2D table format:
     https://portal.c-lockinc.com/api/getweight?d=dailytab&fids=1000001,1000027&st=2020-05-01&et=2020-05-20&full=1
        Headers: RFID,Date1KG,Date2KG,Date3KG,...

   * Get individual scale visits from SmartScales 1000001 and 1000027, May 1st, 2020, midnight until noon:
     https://portal.c-lockinc.com/api/getweight?d=visits&fids=1000001,1000027&st=2020-05-01_00:00:00&et=2020-05-01_12:00:00
        Headers: ScaleID,RFID,DateTime,Duration,WeightKG,StallNumber




    ANIMAL INTAKE REQUESTS
    ######################

    System Types:
        SmartFeed, SmartFeed Pro, Super SmartFeed, Super SmartFeed Producer

    Parameters:
        d:              Data type (daily or visits)
        fids:           SmartFeed ID numbers, separated by commas
        st:             Start time of the data request (YYYY-MM-DD_HH:mm:ss or YYYY-MM-DD format)
        et:             End time of the data request (YYYY-MM-DD_HH:mm:ss or YYYY-MM-DD format)
        ts:             Optional (set to a Date+Time to filter data processed after that time. (i.e. ts=2023-03-03 to show only data processed on or after March 3rd, 2023)
        zip:            Optional (set to 1 to receive data as a ZIP file, otherwise data will be uncompressed CSV)
        includeowner:   Optional (set to 1 to include OwnerID field of each intake measurement)
        h:              Optional  0, 1, 2    (2 = include Parameters header and column headers, 1 = include column headers, 0 = include no headers)


   * Get daily intake from SmartFeeds 10036, 10055, and 10171, May 1st - 20th, 2020:
     https://portal.c-lockinc.com/api/getintake?d=daily&fids=10036,10055,10171&st=2020-05-01&et=2020-05-20
        Headers: RFID,Date,IntakeKG

   * Same as above, but in a 2D table format:
     https://portal.c-lockinc.com/api/getintake?d=dailytab&fids=10036,10055,10171&st=2020-05-01&et=2020-05-20
        Headers: RFID,Date1KG,Date2KG,Date3KG,...
   * Get individual visits from SmartFeeds 10036, 10055, and 10171, May 1st, 2020, midnight
until noon (with and without lastupdated timestamps and OwnerID):

     https://portal.c-lockinc.com/api/getintake?d=visits&fids=10036,10055,10171&st=2020-05-
01_00:00:00&et=2020-05-01_12:00:00&ts=2020-01-01_12:00:00
        Headers:
FeederID,RFID,StartTime,EndTime,Duration,StartMassKG,EndMassKG,IntakeKG,FeedTypeNum,WarningCode,
WarningMsg,SSFTray,LastUpdated

     https://portal.c-lockinc.com/api/getintake?d=visits&fids=10036,10055,10171&st=2020-05-
01_00:00:00&et=2020-05-01_12:00:00&ts=0&includeowner=1
        Headers:
OwnerID,FeederID,RFID,StartTime,EndTime,Duration,StartMassKG,EndMassKG,IntakeKG,FeedTypeNum,Warn
ingCode,WarningMsg,SSFTray


#############################################################################
```

CODE EXAMPLES:

The following examples show how to login and authenticate using your username and password
Then get 2 hours worth of raw secondly data from GreenFeed unit #74 - Nov 1st, 2020 midnight until 01:59:59am

```
Bash with cURL:
---------------
# First Authenticate to receive token:
USER=my_username
PASS=my_password
FID=74
# Spaces must be replaced with _
ST="2020-11-01_00:00:00"
ET="2020-11-01_01:59:59"
TOK=`curl -s -d "user=$USER&pass=$PASS" "https://portal.c-lockinc.com/api/login"`

# Now get data using the login token
curl -s -d "token=$TOK" "https://portal.c-lockinc.com/api/getraw?d=meas&fid=$FID&st=$ST&et=$ET"
> REQ.txt

# Output the data
cat REQ.txt
```

```
R using httr library:
---------------------
USER <- "my_username"
PASS <- "my_password"
FID  <- "74"
# Spaces must be replaced with _
ST   <- "2020-11-01_00:00:00"
ET   <- "2020-11-01_01:59:59"
library(httr) # Must first install  httr with:   install.packages("httr")
```

```
# First Authenticate to receive token:
req <- POST("https://portal.c-lockinc.com/api/login", body=list(user=USER, pass=PASS))
stop_for_status(req)
TOK <- trimws(content(req))

# Now get data using the login token
URL <- paste0("https://portal.c-lockinc.com/api/getraw?d=meas&fid=", FID, "&st=", ST, "&et=",
ET)
req <- POST(URL, body=list(token=TOK))
stop_for_status(req)

a <- content(req)

#Since we have paramaters on the first line, and headers on the second line,
#We can remove the first line, then create a df from csv using the rest of the text
lines <- strsplit(a, "\n")[[1]]
remaining_text <- paste(lines[-1], collapse = "\n")  #Join the remaining lines back into a
single string

#Read as a CSV
df <- read.csv(text = remaining_text, header=TRUE)



Python 3 using urllib:
----------------------
from urllib import request
from urllib import parse

# First Authenticate to receive token:
USER = "my_username"
PASS = "my_password"
FID  = "74"
# Spaces must be replaced with _
ST   = "2020-11-01_00:00:00"
ET   = "2020-11-01_01:59:59"
req = request.urlopen("https://portal.c-lockinc.com/api/login",
bytes('user='+USER+'&pass='+PASS, 'ascii'))
TOK=req.read().decode('ascii').strip()

# Now get data using the login token
URL = "https://portal.c-lockinc.com/api/getraw?d=meas&fid="+FID+"&st="+ST+"&et="+ET
req = request.urlopen(URL, bytes('token='+TOK, 'ascii'))
data = req.read()

# Output the data
print(data.decode("ascii"))




PowerShell:
-----------
# First Authenticate to receive token:
$Body = @{
    user='my_username'
    pass='my_password'
}
$LoginResponse = Invoke-WebRequest 'https://portal.c-lockinc.com/api/login' -UseBasicParsing -
SessionVariable 'Session' -Body $Body -Method 'POST'
$TOK=$LoginResponse.Content.Trim()
```

```
# Now get data using the login token
$Body = @{ token=$TOK }
# Spaces must be replaced with _
$Data = Invoke-WebRequest 'https://portal.c-lockinc.com/api/getraw?d=meas&fid=74&st=2020-11-
01&et=2020-11-01_01:59:59' -UseBasicParsing -Body $Body -Method 'POST'

# Output the data
echo $Data.Content




Excel Visual Basic Macros using XMLHTTP Object:
-----------------------------------------------
Sub DownloadData()
    Dim USER, PASS, TOK, pageURL As String
    Dim FID, ST, ET As String

    ' Clear some cells '
    Range("A1:Z1000").Value = ""

    USER = "my_username"
    PASS = "my_password"
    FID = "74"
    ' Spaces must be replaced with _ '
    ST = "2020-11-01_00:00:00"
    ET = "2020-11-01_01:59:59"


    ' First Authenticate to receive token '
    pageURL = "https://portal.c-lockinc.com/api/login"
    Dim objXmlHttpReq As Object
    Set objXmlHttpReq = CreateObject("Microsoft.XMLHTTP")
    objXmlHttpReq.Open "POST", pageURL, False
    objXmlHttpReq.setRequestHeader "Content-Type", "application/x-www-form-urlencoded"
    objXmlHttpReq.send "user=" + USER + "&pass=" + PASS
    If objXmlHttpReq.Status <> 200 Then Exit Sub
    TOK = objXmlHttpReq.responseText


    ' Now get data using the login token '
    pageURL = "https://portal.c-lockinc.com/api/getraw?d=meas&fid=" & FID & "&st=" & ST & "&et="
& ET
    objXmlHttpReq.Open "POST", pageURL, False
    objXmlHttpReq.setRequestHeader "Content-Type", "application/x-www-form-urlencoded"
    objXmlHttpReq.send "token=" + TOK
    If objXmlHttpReq.Status <> 200 Then Exit Sub


    'This will paste the results into cells starting at A1 of the current worksheet '
    Dim x As Variant
    x = Split(objXmlHttpReq.responseText, vbLf)
    Range("A1:A" & UBound(x) + 1).Value2 = Application.WorksheetFunction.Transpose(x)

    Range("A1:A" & UBound(x) + 1).TextToColumns _
        Destination:=Range("A1"), DataType:=xlDelimited, _
        TextQualifier:=xlDoubleQuote, ConsecutiveDelimiter:=False, Tab:=False, _
        Semicolon:=False, Comma:=True, Space:=False, Other:=False
End Sub
```

```
C#:
---
using System;
using System.IO;
using System.Net;
using System.Text;

class Program
{
    static void Main(string[] args)
    {
        // First Authenticate to receive token:
        string USER = "my_username";
        string PASS = "my_password";
        string FID = "74";
        // Spaces must be replaced with _
        string ST = "2020-11-01_00:00:00";
        string ET = "2020-11-01_00:01:59";

        string token;

        // Authenticate and receive token
        string loginUrl = "https://portal.c-lockinc.com/api/login";
        string loginData = "user=" + USER + "&pass=" + PASS;
        byte[] loginBytes = Encoding.ASCII.GetBytes(loginData);
        WebRequest loginRequest = WebRequest.Create(loginUrl);
        loginRequest.Method = "POST";
        loginRequest.ContentType = "application/x-www-form-urlencoded";
        loginRequest.ContentLength = loginBytes.Length;
        using (Stream stream = loginRequest.GetRequestStream())
        {
            stream.Write(loginBytes, 0, loginBytes.Length);
        }
        using (WebResponse loginResponse = loginRequest.GetResponse())
        using (Stream responseStream = loginResponse.GetResponseStream())
        using (StreamReader reader = new StreamReader(responseStream))
        {
            token = reader.ReadToEnd().Trim();
        }

        // Get data using the login token
        string dataUrl = "https://portal.c-lockinc.com/api/getraw?d=meas&fid=" + FID + "&st=" +
ST + "&et=" + ET;
        string dataParams = "token=" + token;
        byte[] dataBytes = Encoding.ASCII.GetBytes(dataParams);
        WebRequest dataRequest = WebRequest.Create(dataUrl);
        dataRequest.Method = "POST";
        dataRequest.ContentType = "application/x-www-form-urlencoded";
        dataRequest.ContentLength = dataBytes.Length;
        using (Stream stream = dataRequest.GetRequestStream())
        {
            stream.Write(dataBytes, 0, dataBytes.Length);
        }


        string responseData;
        using (WebResponse dataResponse = dataRequest.GetResponse())
        using (Stream responseStream = dataResponse.GetResponseStream())
        using (StreamReader reader = new StreamReader(responseStream))
        {
            responseData = reader.ReadToEnd();
            Console.WriteLine(responseData);
        }
    }
}
```