SiteTrax.io API

The SiteTrax.io API enables seamless video processing for supply chain asset tracking by integrating with the SiteTrax backend primarily through data storage services such as Amazon S3 buckets. The API facilitates video uploads, typically captured using the SiteTrax.io Android app or other compatible cameras, to a specified data storage platform such as an Amazon S3 bucket for analysis. After uploading, SiteTrax.io processes the video and delivers the data to any REST API server, allowing for easy integration with other systems.

Key features include:

1. **Video Upload**: Videos up to 1 minute in length are uploaded to the designated data store, following specific guidelines such as a 1920x1080 resolution, 30 FPS, and embedded GPS data in subtitles.
2. **Processing and Output**: Once uploaded, the SiteTrax.io backend analyzes the video and pushes the extracted data to any REST API compatible server. The processing time depends on the number of assets in the video.
3. **Camera Types**: The API supports data capture from various camera types, including mobile apps and stationary cameras, with options for both dynamic and static GPS coordinates.

The SiteTrax.io API simplifies the process of video uploading and analysis, enabling efficient asset tracking and data integration.

- SiteTrax.io API - Input (Video)
- SiteTrax.io API - Output (JSON)
- Test Development Environment - Setup SiteTrax.io Test API and JSON

SiteTrax.io API - Input (Video)

Getting Started

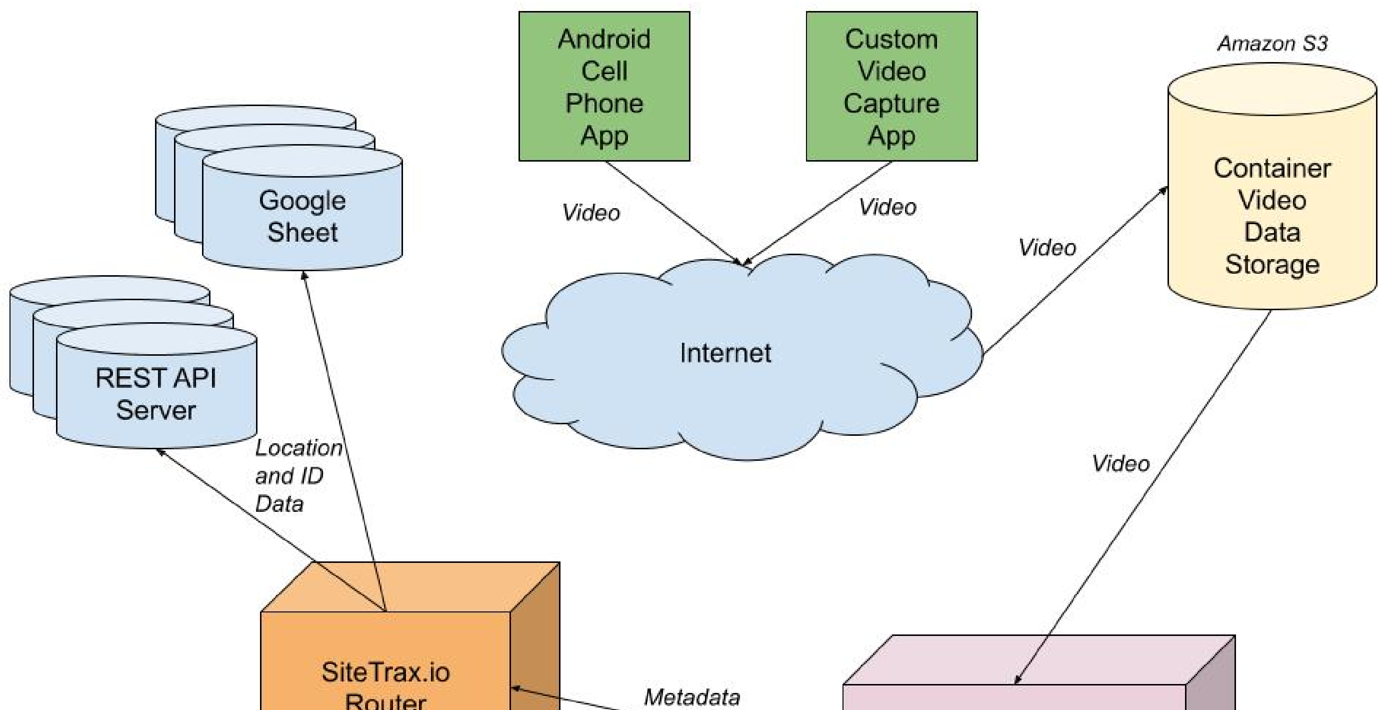
The SiteTrax.io backend requires a centralized data storage to transfer data to and from it. Currently, Amazon AWS S3 buckets are supported as the recommended data store. If needed, the bucket information and access keys can be provided by SiteTrax.io or your organization is welcome to set up their own S3 buckets. Please submit a support ticket to request a bucket and access keys to process videos.

Request API Server Access>>

For instructions on setting up a test environment, please see the Test Development Environment - Setup SiteTrax Test API and JSON article.

Overview

The SiteTrax.io backend analyzes video stored in an Amazon S3 bucket and outputs that data to any REST API server.



Reference:

https://docs.google.com/drawings/d/1ZwDE18DMtbIDPIujX_M61jRm2SvanIolZJ7RsnY4XW0/edit

Step #1: Upload Video (Push) - See ***Input - Video Specifications*** *below for more details*

- Option #1: Using the SiteTrax.io Android app, capture video using any mode, which will upload capture and upload the video automatically for you into a storage bucket.
- Option #2: Upload video (no more than 1 minute in length) to the Amazon S3 bucket provided.

Step #2: REST API Server (Push)

After the video is uploaded in Step #1 above, the SiteTrax.io Backend will analyze the video and push the data analyzed to any REST API server via the SiteTrax.io Router. See ***Output - REST API*** page more information.

NOTE: Make sure to request API Server access via the link in the Overview section above.

Input - Video Specifications

The files that should be uploaded in `{bucket_name}/notprocessed` folder. **The maximum recommended length of a video is 1 minute**.

1. Video length: If the video is more than 1 minute in length, it is recommended to split the video into multiple 1 minute videos.
2. The recommended resolution of video is 1920x1080 with a frame rate of 30 frames per second.
3. The lens of 6mm or longer should be used. A lens less than 5mm could cause distortion and the backend may not be able to read the OCR.
4. The GPS information should be embedded in the subtitles of the video (this can be overwritten for static cameras)

SiteTrax.io Camera Types

SiteTrax.io can capture data from many different types of cameras including Android and basic security cameras. For implementation purposes, we recommend:

1. SiteTrax.io Mobile app (i.e. Android) - Many cameras for one bucket. The GPS coordinates are dynamically sent base on the location of the mobile app.
2. SiteTrax.io Gate or stationary camera - One camera to data store. The GPS coordinates have to be manually defined for each camera.

GPS Encoding

Below is an example of embedded subtitles in SRT format.

```
1
00:00:00,000 --> 00:00:00,150
36.8626459 -76.2314164

2
00:00:00,150 --> 00:00:00,300
36.8626459 -76.2314164

3
00:00:00,300 --> 00:00:00,450
36.8626459 -76.2314164
.
.
.
```

Video creation time should be present in both video stream and subtitle stream. In addition to this, creation time should also be present in file metadata. Video should be in mp4 format only. Below in an example of metadata as ffmpeg output.

```
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from '2_20211018T042608717Z_s00.mp4':
  Metadata:
    major_brand     : isom
    minor_version   : 512
    compatible_brands: isomiso2avc1mp41
    creation_time   : 2021-10-18T16:26:08.000000Z
    encoder         : Lavf58.67.100
  Duration: 00:00:14.55, start: 0.000000, bitrate: 29921 kb/s
  Stream #0:0(eng): Video: h264 (Baseline) (avc1 / 0x31637661), yuvj420p(pc, smpte170m/bt470bg/smpte170m),
    Metadata:
      creation_time   : 2021-10-18T16:26:08.000000Z
      handler_name    : VideoHandle
      vendor_id       : [0][0][0][0]
    Side data:
      displaymatrix: rotation of -90.00 degrees
  Stream #0:1(und): Subtitle: mov_text (tx3g / 0x67337874), 1 kb/s (default)
    Metadata:
      creation_time   : 2021-10-18T16:26:08.000000Z
      handler_name    : SubtitleHandler
```

Sample videos can be found here.

Video Upload

Once a video is generated, it can be uploaded to your designated bucket. The code to upload the video to the bucket is below.

```
import boto3



s3_client = boto3.client('s3', aws_access_key_id='',
                    aws_secret_access_key='',
                    region_name='eu-central-1'
                    )
s3_client.upload_file('/internal video path', bucket_name, 'notprocessed/' + video_name + '.mp4')
```

Once the video upload is complete, SiteTrax.io backend will start processing the video. The processing time is proportional to the number of trackable assets in the video.

If records from a video do not show up please check our status page or if all services are up then please file a support ticket at our support page.

SiteTrax.io API - Output (JSON)

The SiteTrax.io JSON API is a simple, JSON-backed push interface for sending SiteTrax.io captured geolocation and asset ID information to any REST API server. It is fully compatible with most REST API servers that accept JSON messages.

The JSON API is intended for software developers. To use it you should be familiar with web programming and be comfortable creating applications that consume web services through HTTP requests.

Getting Started

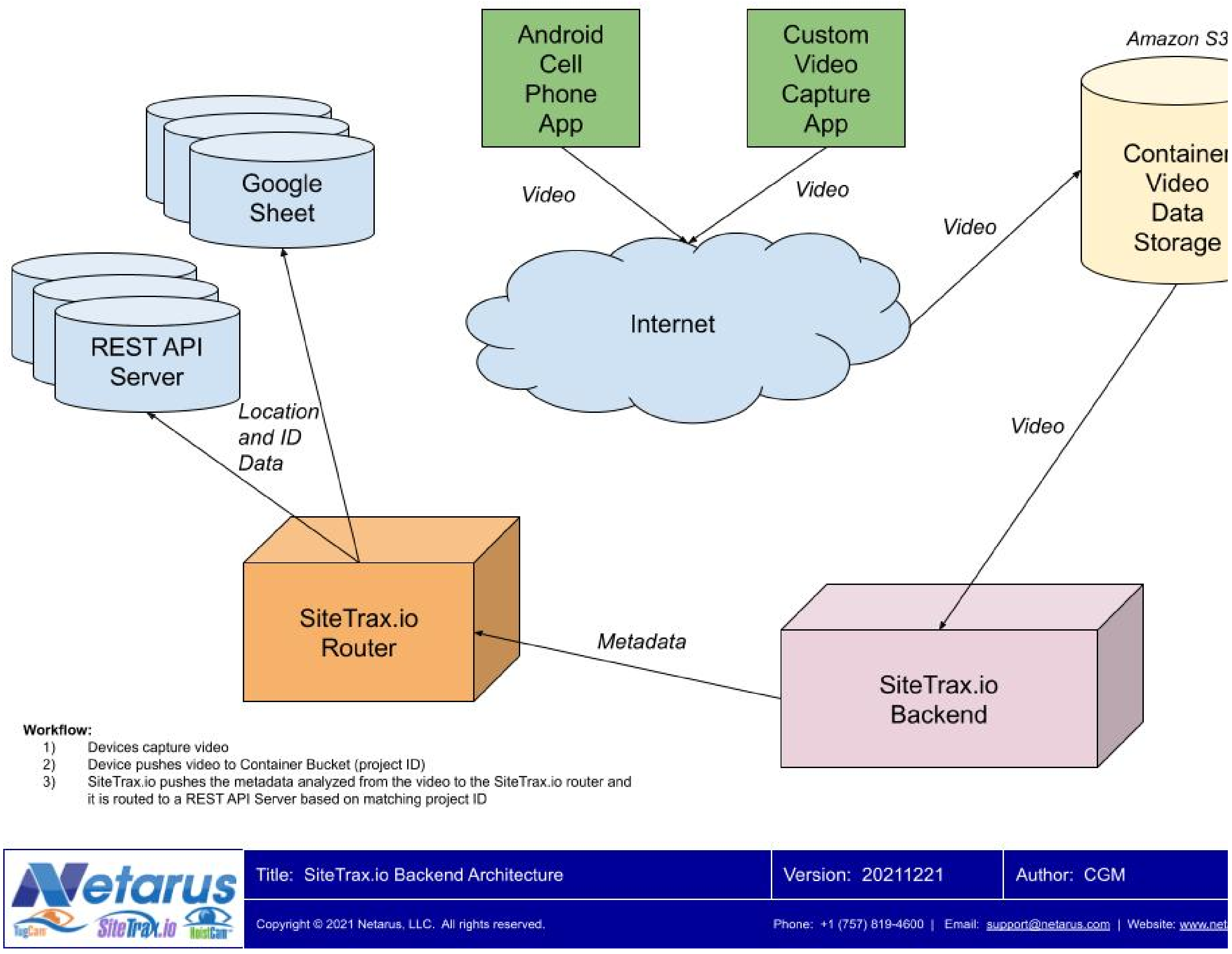
The SiteTrax.io backend requires an AWS bucket to be able to transfer data to and from it. The bucket information and access keys will be provided by SiteTrax. Please submit a support ticket to request a bucket and access keys to process videos.

Request API Server Access>>

For instructions on setting up a test environment, please see the Test Development Environment - Setup SiteTrax Test API and JSON article.

Overview

The SiteTrax.io backend analyzes video stored in an Amazon S3 bucket and outputs that data to any REST API server.



Workflow:

1) Devices capture video
2) Device pushes video to Container Bucket (project ID)
3) SiteTrax.io pushes the metadata analyzed from the video to the SiteTrax.io router and it is routed to a REST API Server based on matching project ID

Title: SiteTrax.io Backend Architecture	Version: 20211221	Author: CGM
Copyright © 2021 Netarus, LLC. All rights reserved.	Phone: +1 (757) 819-4600 | Email: support@netarus.com | Website: www.net	

Reference:

https://docs.google.com/drawings/d/1ZwDE18DMtbIDPIujX_M61jRm2SvanIolZJ7RsnY4XW0/edit

Step #1: Upload Video (Push) - See ***Input - Video Specifications*** *below for more details*

- Option #1: Using the SiteTrax.io Android app, capture video using any mode, which will upload capture and upload the video automatically for you into a storage bucket.
- Option #2: Upload video (no more than 1 minute in length) to the Amazon S3 bucket provided.

Step #2: REST API Server (Push)

After the video is uploaded in Step #1 above, the SiteTrax.io Backend will analyze the video and push the data analyzed to any REST API server via the SiteTrax.io Router. See ***Output - REST API*** section below for more information.

NOTE: Make sure to request API Server access via the link in the Overview section above.

Input - Video Specifications

The files that should be uploaded in `{bucket_name}/notprocessed` folder. **The maximum recommended length of a video is 1 minute**.

1. Video length: If the video is more than 1 minute in length, it is recommended to split the video into multiple 1 minute videos.
2. The recommended resolution of video is 1920x1080 with a frame rate of 30 frames per second.
3. The lens of 6mm or longer should be used. A lens less than 5mm could cause distortion and the backend may not be able to read the OCR.
4. The GPS information should be embedded in the subtitles of the video (this can be overwritten for static cameras)

SiteTrax.io Camera Types

SiteTrax.io can capture data from many different types of cameras including Android and basic security cameras. For implementation purposes, we recommend:

1. Mobile App (i.e. Android) - Many cameras for one bucket. The GPS coordinates are dynamically sent base on the location of the mobile app.
2. Stationary camera - One camera to one bucket. The GPS coordinates have to be manually defined for each camera.

Output - REST API

The orientation of output JSON is in records format. These records represent each detected asset and its metadata. These records of a single video are sent in series one after another. Samples of each type of record are given below.

International Intermodal Shipping Container ID

```
{
"video_name": "video24-05-07_09-43-03-39_01424",
"type": "International Intermodal Vertical ID",
```

"text": "TRBU5341840",
"datetime": "2024-05-07T13:43:43.701000Z",
"datetime_original": "2024-05-07T13:43:43.701000Z",
"datetime_digitized": "2024-05-07T13:43:43.701000Z",
"gps_lat": 30.4178413846,
"gps_lon": -81.5642421112,
"container_company": "CJ DARCL LOGISTICS LIMITED",
"container_country": "India",
"status": "Success",
"status_code": "A0",
"asset_image": "https://d11isnr6z7061v.cloudfront.net/video24-05-07_09-43-03-39_01424_TRBU5341840.jpg",
"asset_heading": "R2L",
"camera": "B8A44F9E3A9E"
}

International Intermodal Shipping Container Chassis

{
"video_name": "1_20210810T103622681Z_s00",
"type": "International Intermodal Chassis ID",
"text": "MCCZ406697",
"datetime": "2024-05-07T13:43:43.701000Z",
"datetime_original": "2024-05-07T13:43:43.701000Z",
"datetime_digitized": "2024-05-07T13:43:43.701000Z",
"gps_lat": 30.4178413846,
"gps_lon": -81.5642421112,
"container_company": "CJ DARCL LOGISTICS LIMITED",
"container_country": "India",
"status": "Success",
"status_code": "A0",
"asset_image": "https:\/\/container-thumbnails-public.s3.amazonaws.com\/1_20210810T103622681Z_s00_MCCZ40
"asset_heading": "R2L",
"camera": "B8A44F9E3A9E"
}

Shipping Container and Chassis

{
"video_name": "video24-05-07_09-43-03-39_01424",
"type": "International Intermodal Vertical ID",
"text": "TRBU5341840",
"datetime": "2024-05-07T13:43:43.701000Z",

```
"datetime_original": "2024-05-07T13:43:43.701000Z",
"datetime_digitized": "2024-05-07T13:43:43.701000Z",
"gps_lat": 30.4178413846,
"gps_lon": -81.5642421112,
"container_company": "CJ DARCL LOGISTICS LIMITED",
"container_country": "India",
"status": "Success",
"status_code": "A0",
"asset_image": "https://d11isnr6z7061v.cloudfront.net/video24-05-07_09-43-03-39_01424_TRBU5341840.jpg",
"asset_heading": "R2L",
"stacking": "314875212301145617478946594181356160171_000_000",
"camera": "B8A44F9E3A9E"
}
```

Generic Text Asset Tracking

```
{
"video_name": "video24-05-07_09-43-03-39_01424",
"type": "Generic OCR",
"text": "TRBU5341840",
"datetime": "2024-05-07T13:43:43.701000Z",
"datetime_original": "2024-05-07T13:43:43.701000Z",
"datetime_digitized": "2024-05-07T13:43:43.701000Z",
"gps_lat": 30.4178413846,
"gps_lon": -81.5642421112,
"container_company": "CJ DARCL LOGISTICS LIMITED",
"container_country": "India",
"status": "Success",
"status_code": "A0",
"asset_image": "https://d11isnr6z7061v.cloudfront.net/video24-05-07_09-43-03-39_01424_TRBU5341840.jpg",
"asset_heading": "R2L",
"camera": "B8A44F9E3A9E"
}
```

Each record has a variety of metadata to track and localize an asset across the globe. There are also status codes included in international intermodal shipping container records to represent how sure we are of each input container.

API Reference

Field	Type	Description
`video_name`	`str`	Name of the video in which container was detected.
`type`	`str`	Type of detected asset ID. (Horizontal of Vertical)

Field	Type	Description
`text`	`str`	Detected asset ID.
`datetime`	`str`	Time in ISO8601 of when the camera was passing in front of the asset.
`datetime_original`	`str`	Time in ISO8601 of when the video was created.
`datetime_digitized`	`str`	Time in ISO8601 of when the video was digitized.
`gps_lat`	`float`	Latitude of detected asset in decimal format.
`gps_lon`	`float`	Longitude of detected asset in decimal format.
`container_company`	`str`	Company associated with the container. For more information please check BIC database. (International Intermodal Exclusive Entry)
`container_country`	`str`	Country of the company. (International Intermodal Exclusive Entry)
`status`	`str`	Verbal description of the container entry detection status. (International Intermodal Exclusive Entry)
`status_code`	`str`	Return code associated with the status of the entry. (International Intermodal Exclusive Entry)
`asset_image`	`str`	A public URL to the image of the asset that we are trying to track.
`asset_heading [optional]`	`str`	Specifies the asset direction within the camera frame. R2L: Asset moving from right to left. L2R: Asset moving from left to right. U2D: Asset moving from up to down. D2U: Asset moving from down to up.
sorting [optional]	str	[Deprecated] [Replaced by below field] If more than one processing type is selected, this field is populated. Users can sort the records via this field to match assets that may pair together.

Field	Type	Description
stacking [optional]	str	For containers and chassis detection. This field pairs a container record with it's corresponding chassis record.
camera [optional]	str	Unique serial number of the camera. Only available to virtual gate applications.

Status Codes (International Intermodal Exclusive)

Code	Description
A0	No errors were detected for the container entry.
A1	Check digit of the entry is interpolated.
I1	BIC repository cannot verify the company of the container.
I2	Check digit of the container cannot be verified. A record containing this code can be classified as incorrect right away.
I3	We detected the container but its ID cannot be determined. This can be due to damaged or scratched ID or the asset might not be a standard asset.
I4	The provided database of assets does not contain the listed asset.
I5	When processing for both container and chassis is selected, if there is no container for a chassis because it is a bare chassis then the entry for that missing container will have this status code.
I6	Low confidence detection.
I7	Low confidence detection. ID has been interpolated.

The status code, status text, country and company are only available for international intermodal shipping container IDs. For chassis ID and generic text detection, these entries are not available.

Generating Test Sample Data

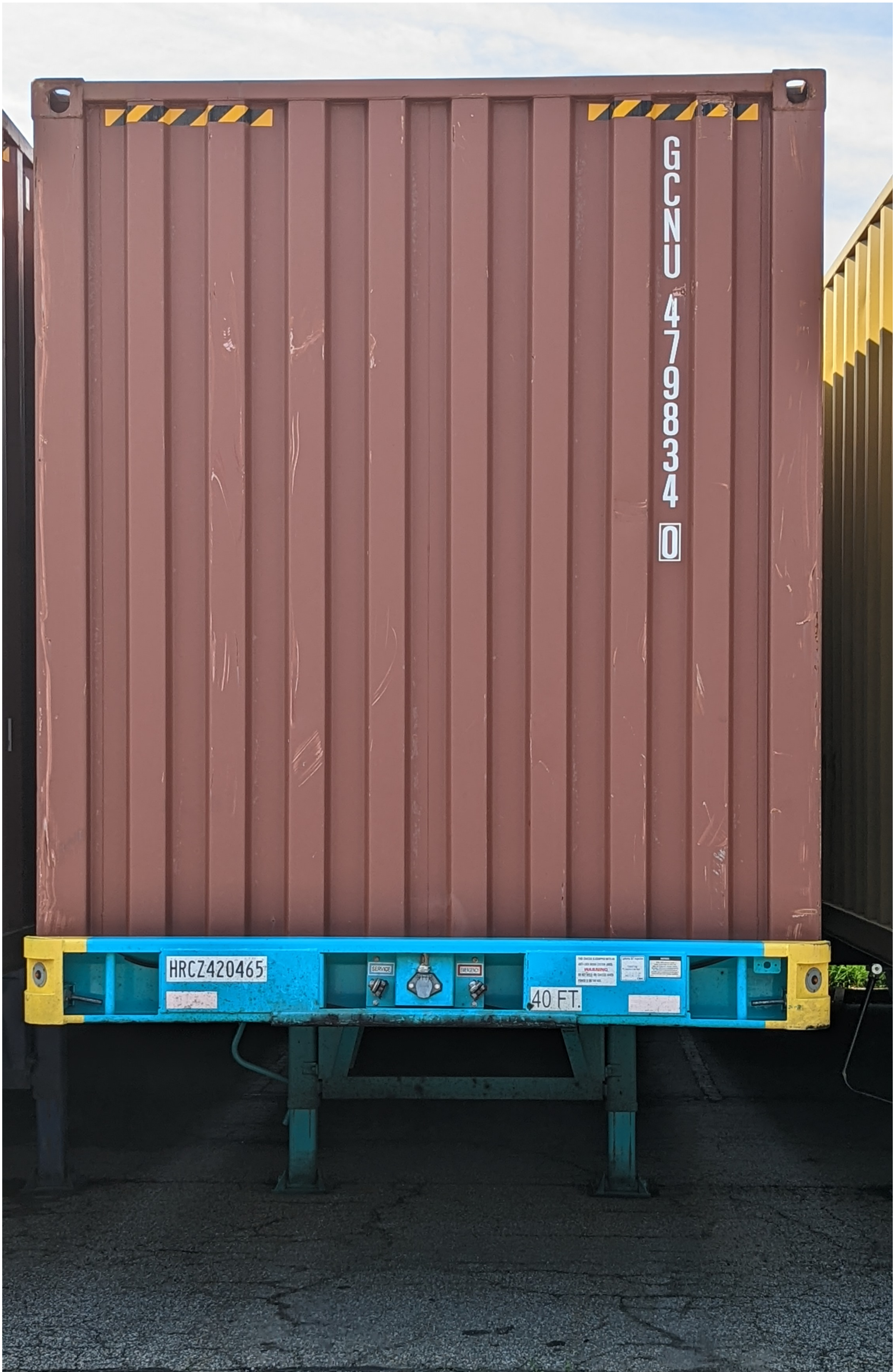
We recommend using SiteTrax.io directly to generate test data. This can be done by scanning a picture of an intermodal containers to trigger events whenever you wish. It is recommended to scan a complete container so that the object detection and first identify the container and then scan the ID.

Preparing test data to be pushed to you for testing:

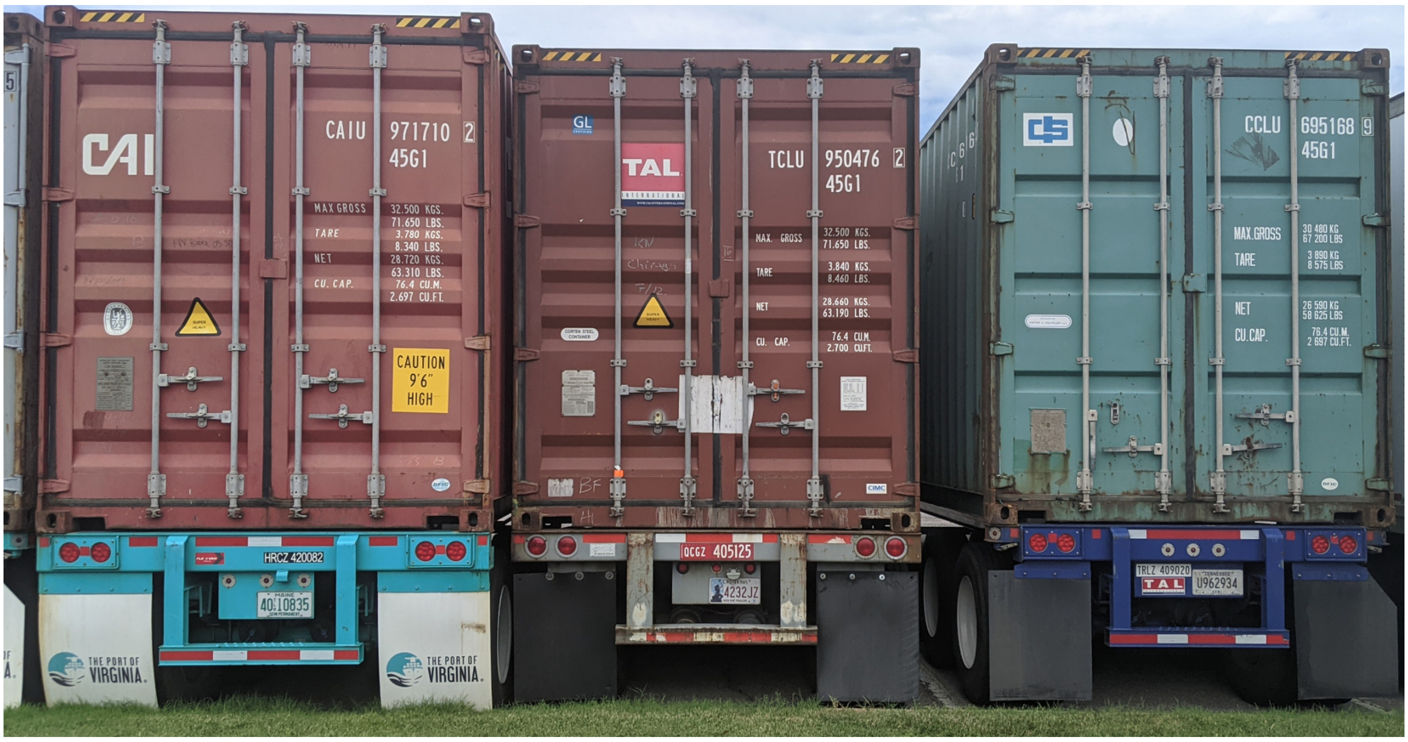
1. **Locate** an Android 10 or later device
2. **Register** for a free SiteTrax.io account - https://www.sitetrax.io (Click ***Register***)
3. **Download** the SiteTrax.io Android app (link provided after registering your account above).
4. **Scan** an international intermodal container and/or chassis.
5. **View** the results in a Google Spreadsheet.

Next, use the one or more images of containers on a full screen computer and scan the ISO numbers. Make sure to have the whole object in frame and make sure the picture is taking up the entire screen. Below are some sample images (click to view in a New Window in your web browser).

Sample Image #1 - International Intermodal Container:


GCNU 479834 0
HRCZ420465
SERVICE
EMERGENCY
WARNING
40 FT.

Sample Image #2 - Multiple International Intermodal Containers:



Use the images above in a full screen monitor to test scan with the Android app. The data should push to the spreadsheet that you set up to see the sample payload within a spreadsheet when you registered.

Reviewing the Data:

Please visit https://www.sitetrax.io and login to the SiteTrax.io Service Portal if you need to troubleshoot or monitor what you scanned (along with the results). A sample JSON payload is also available with each record in the SiteTrax.io Service Portal if needing to troubleshoot further.

NOTE: The default settings for free accounts is typically limited to 100 scans per month and scanning one asset (or image) at a time. If you wish to scan continuously with the app, please submit a support ticket at our support page.

Custom Endpoint: When you're ready to have live data pushed to your endpoint, please submit a support ticket at our support page and supply your registered username and project to request that the data is sent to your webhook destination.

Troubleshooting

If records from a video do not show up please check our status page or if all services are up then please submit a support ticket at our support page.

Test Development Environment - Setup SiteTrax.io Test API and JSON

The following article shows how to set up and configure a test environment to configure the SiteTrax.io API to push JSON messages to your REST API server via a webhook call (POST).

Prerequisites:

The following prerequisites are required in order to testing communication from SiteTrax.io REST API Client to your REST API server.

1. **SiteTrax.io Storage** - For most accounts, SiteTrax.io uses Amazon S3 Buckets for storage. We recommend using your own buckets if you want to store your data long term. Provide us with the following information about your bucket so that we can configure the storage:
   1. Storage bucket name,
   2. region (default us-east-1),
   3. client key, and
   4. secret key
2. **REST API Server** - REST API Server set up to receive messages. For example:
   1. Postman - www.postman.co
   2. Nightingale - www.nightingale.rest
3. **SiteTrax.io Router Configured** - SiteTrax.io configured REST API router to send messages to the REST API Server above. This is done by the SiteTrax.io staff
4. **Sample Video Payload** - Sample video payload to upload to storage.
5. **Bucket Browser Utility** - upload into the **notprocessed/** folder of the storage container (i.e. AWS S3 Bucket)
   - IOS: https://www.stratospherix.com/amazon-s3/
   - Android: https://play.google.com/store/apps/details?id=lysesoft.s3anywhere&hl=en_US&gl=U

S

Configure Test REST API or JSON Message Trigger

The following steps define how to trigger a REST API message from the SiteTrax.io REST API Client to your REST API Server.

Step #1: Configure AWS Bucket to Receive Video Payload

You will receive AWS bucket credentials from Netarus. This includes a bucket name, region, client key and secret key. We recommend using a AWS bucket browser app (referenced above) to upload sample payload to your bucket.

Alternatively, you can download the SiteTrax.io Capture Android app here. This is configure easily using a predefined PIN.

Step #2: Configure your REST API Server (Webhook)

A REST API Server will need to be set up to receive JSON messages from SiteTrax.io in order to ingest the data. See our article on **SiteTrax.io - Video (Input) REST API and JSON (Output)** documentation**.**

Step #3: Provide REST API Endpoint URL

Once a REST API endpoint URL is provided we will be able to POST our JSON payload of the asset data to the REST API.

Step #4: Upload Video to Bucket to Trigger JSON Payload

You may use the bucket credentials provided or the SiteTrax.io Capture app to upload videos of assets to your bucket, after processing the videos the data will be POSTed to the REST API server.