MassDOT
Smart Work Zones Manager Application

APPENDIX A – VENDOR API V1.0 SPECIFICATION

Document Version 6.0

TABLE OF CONTENTS

1. Introduction 1
2. Security 2
1.1. Authorization Key Generation 2
1.2. Authorization Key Usage 2
1.2.1. Authorization Key Usage Example 2
3. Work Zone Definition 3
4. API Requests 4
4.1. Endpoint Summary 4
4.2. Vendor Information 5
4.2.1. Vendor Information 5
4.3. Work Zone Projects 6
4.3.1. Work Zone Project 7
4.3.2. Contractor Information 7
4.4. Road Events 8
4.4.1. Type of Work (from WZDx v2.0) 8
4.4.2. Lane (from WZDx v2.0) 9
4.4.3. Lane Restriction (from WZDx v2.0) 9
4.4.4. Road Event (from WZDx v2.0) 9
4.5. Field Devices Configuration 11
4.5.1. Field Device 12
4.5.2. DMS Properties 12
4.5.3. CCTV Properties 12
4.5.4. CTTV Preset 12
4.6. Road Event Dynamic Metrics 13
4.6.1. Road Event Dynamic Metrics 14
4.7. Field Devices Data 15
4.7.1. Field Device State 15
4.7.2. Dynamic Message Signs (DMS) 15
4.7.3. Vehicle Detection Sensors (VDS) 17
4.7.4. CCTV Cameras 19
4.8. Live Video Streaming *(Optional)* 20
4.9. DMS Message *(Optional)* 21
4.9.1. DMS Message Response 22
4.10. CCTV Preset *(Optional)* 23
4.10.1. CCTV Preset Response 24

Document Control

Client	:	Massachusetts Department of Transportation (MassDOT)
Project Name	:	Smart Work Zones Manager (SWZM) Application
Document Title	:	Appendix A – Vendor API v1.0 Specification
Contract #	:	88163
KCUS Reference	:	R14
Version	:	6.0
Digital Master	:	
Originator	:	Brian Delsey
Reviewer	:	Michelle Boucher, Jacob Brady
Authorization	:	Michelle Boucher
Circulation List	:	MassDOT: Neil Boudreau, Kevin Chiang, Douglas Small
History	:	8/1/17 – Document Version 1.0 submitted with SRS electronically to Neil Boudreau, Kevin Chiang And Douglas Small
	:	12/26/17 – Document Version 1.5 submitted electronically to Neil Boudreau, Kevin Chiang And Douglas Small
	:	1/30/18 – Document Version 1.6 for internal review with VerMAC
	:	8/31/18 – Document Version 2.0 for internal review with VerMAC
	:	5/1/19 – Document Version 3.0 for internal review with VerMAC
	:	1/30/20 – Document Version 4.0 for internal review with VerMAC
	:	1/31/20 – Document Version 4.2 for internal review with VerMAC
	:	2/3/20 – Document Version 4.2 for internal review with VerMAC
	:	2/16/20 – Document Version 4.3 for internal review with VerMAC
	:	2/25/20 – Document Version 4.4 for internal review with VerMAC
	:	4/28/20 - Document Version 5.0 submitted electronically to VerMAC
	:	6/19/20 – Document Version 6.0 submitted electronically to Neil Boudreau, Kevin Chiang And Douglas Small, and VerMAC

1. INTRODUCTION

The Application Programming Interface (API) defined herein is the API that must be implemented by any Smart Work Zone (SWZ) vendor application to be integrated with and consumed by the MassDOT Smart Work Zone Manager (SWZM) software. All functions of this API not marked as "*optional*" are required in order for the SWZ field application to be considered compliant. This API allows the MassDOT SWZM software to centrally monitor and control one or more SWZ implementations. The API utilizes Representational State Transfer (REST) principles and follows standard HyperText Transfer Protocol (HTTP) and rules. Every resource of the SWZ is exposed as a Unified Resource Locator (URL). The URL of each resource can be obtained by accessing the API Root Endpoint (to be defined by the SWZ vendor). All responses will be transmitted using JavaScript Object Notation (JSON).

The API content includes the core data as defined by the Work Zone Activity Data (WZAD) effort of FHWA's overall Work Zone Data Initiative. Specifically, the content as defined by the Work Zone Data Exchange (WZDx) specification is supported herein.

All mentions herein to "WZDx v2" refer to the documents that can be found at the following URL:

https://github.com/usdot-jpo-ode/jpo-wzdx/releases/tag/v2.0

All mentions herein to "TMDD" refer to the documents that can be found at the following URL:

https://www.ite.org/technical-resources/standards/tmdd/

2. SECURITY

The SWZ API will be secured using a simple “username/password” authentication key as part of the HTTP header of the request. From a security and authorization perspective, the SWZM software is considered the client and the SWZ field application is considered the server. All authorized clients will be provided a ‘username’ and ‘password’ so that they may authenticate and use the API.

1.1. AUTHORIZATION KEY GENERATION

Once a client (e.g. the SWZM application) has received their username and password, the client can generate an authorization key as follows:

a. Concatenate ‘username’ and ‘password’ separated by a colon: *username:password*

b. Generate a base-64 encoded string of the concatenated username and password value. This result is the required ‘basic’ authorization key to be used in all API requests that require authorization (as specified herein).

1.2. AUTHORIZATION KEY USAGE

The client may make any request to the API using the appropriate credentials as specified in the standard HTTP Authorization header of the request. The credentials shall be set as the base-64 encoded string that comprises the ‘username’ and ‘password’ separated by a colon, that is,

```
Authorization: Basic Base64(username:password)
```

1.2.1. Authorization Key Usage Example

Username: `swzManager`
Password: `massdot`

Authorization header value: `base64(”swzManager:massdot”) = c3d6TWFuYWdlcjptYXNzZG90`

On the server side, the validation process is as follows:

1. Decode the base-64 string in the HTTP header Authorization field.
2. Strip out the username, password, discarding the colon (i.e. “swzManager” and “massdot”).
3. Verify the received username and password.

Any attempt to use invalid credentials will result in a “401 Unauthorized” response from the API.

3. WORK ZONE DEFINITION

Version 1.1 of the WZDx specification referred to the road work effort in a single direction of traffic as a “work zone activity”, WZDx v2 refers to this as a more generic “road event”. For the purposes this API, “work zone activity” and “road event” are considered to be the same.

The following definition of a work zone shall be used by the Vendor to define the configuration of a work zone via the API. This definition accommodates many different scenarios and may at first look complex but for most deployments in practice will be quite simple.

- A **work zone project**, the highest-level organizational entity, will have a unique identifier that distinguishes it from all other work zone projects managed by SWZM.
- A work zone project may encompass multiple roadways.
- A work zone project may span multiple towns.
- A work zone project may span a single direction of travel only or may span two directions of travel.
  - Each direction of travel will contain 1 or more **road events**.
  - Each road event within a work zone project will have a unique identifier that distinguishes it from all other work zone activities managed by SWZM, as well as a sub-identifier that references the work zone project to which the activity belongs.
  - Any road event of a project may be distinguished from other road events of the same project based on a number of possible qualifiers such as:
    (1) Direction of travel
    (2) Lane configuration
    (3) Type of Work
    (4) Etc.

4. API REQUESTS

The API shall follow a request/response model where all communications are initiated by the client (i.e. the SWZM software). Unless specified otherwise, the SWZ vendor applications shall only respond to authorized requests from the SWZM and never initiate any unsolicited data transmission.

All endpoints defined herein are relative to an API Root Endpoint which is to be defined by the SWZ vendor.

The messages defined herein comprise version “1.0” (v1) of the API.

4.1. ENDPOINT SUMMARY

Request URL	HTTP Methods	Description
/vendor	GET	Provides vendor summary information such as contact details.
/workZoneProjects	GET	Provides an overview of all work zone projects. This request returns primarily static, project configuration data.
/roadEvents	GET	Provides the general details (excluding traffic metrics) of all roads events in the vendor system.
/fieldDevices	GET	Provides the configuration details of all intelligent field devices in the vendor system.
/roadEvents/dynamicMetrics	GET	Provides the operational metrics for all road events in the vendor system.
/fieldDevices/state	GET	Provides the operational state (dynamic data) of all field devices in the vendor system.
/fieldDevices/vds/trafficData	GET	Provides the dynamic traffic data information (speed, volume, etc.) for each vehicle detection sensor (VDS) in the system.
/fieldDevices/dms/:deviceId/message	GET, PUT, DELETE	Allows a client to post (PUT) a message to a DMS, release (DELETE) a user-posted messaged from a DMS, or view (GET) the message posted on a DMS, identified by ‘deviceId’.
/fieldDevices/cctv/:deviceId/preset	GET, PUT	Allows a client to set (PUT) a CCTV camera to a particular preset or get (GET) the preset set on a CCTV, identified by ‘deviceId’.

4.2. VENDOR INFORMATION

The API shall support the request for vendor information. A vendor (i.e. supplier of a SWZ field system) information request shall not require any authorization.

Function	Vendor Information Request
Endpoint	/vendor
HTTP Methods	GET
Headers	Accept: application/json
URL Parameters	n/a
Data Parameters	n/a
Response Object	{…} (see ***Vendor Information***)

Sample Call	`$ curl`
	`-H 'Accept: application/json'`
	`https://{root}/api/v1/vendor`

Success Response Example	`Code: 200, Content (json):`
	`{`
	`"name": "ABC Company",`
	`"contact_name": "Lisa Smith",`
	`"contact_phone": "888-111-1234",`
	`"contact_email": "lisa.smith@abc.com"`
	`}`

Error Response	n/a

4.2.1. Vendor Information

Field Name	JSON Type	Description	Conformance
name	String	Name of the smart work zone vendor.	Required
contact_name	String	Primary contact name.	Required
contact_phone	String	Primary contact phone number.	Required
contact_email	String	Primary contact email address.	Required
alternate_contact_name	String	Alternate contact name.	Optional
alternate_contact_phone	String	Alternate contact phone number.	Optional
alternate_contact_email	String	Alternate contact email address.	Optional
vendor_url	String	Vendor company URL.	Optional

4.3. WORK ZONE PROJECTS

The API shall offer an endpoint that provides a list of work zone projects configured by the vendor.

<table>
<tr><th>Function</th><th colspan="4">Work Zone Projects Request</th></tr>
<tr><td>Endpoint</td><td colspan="4">/workZoneProjects</td></tr>
<tr><td>HTTP Methods</td><td colspan="4">GET</td></tr>
<tr><td>Headers</td><td colspan="4">Authorization: Basic Base64(username:password)
Accept: application/json</td></tr>
<tr><td>URL Parameters</td><td colspan="4">n/a</td></tr>
<tr><td>Request Data</td><td colspan="4">n/a</td></tr>
<tr><td rowspan="3">Response Object</td><td>Field Name</td><td>JSON Type</td><td>Description</td><td>Conformance</td></tr>
<tr><td>update_date</td><td>String</td><td>ISO 8601 formatted UTC date/time indicating the last date and time at which the list of projects was updated (e.g. “yyyymmddThhmmss+z”). This would include any of the following:
New project(s) added
Existing project(s) changed (any data associated with project)
Project(s) deleted</td><td>Required</td></tr>
<tr><td>work_zone_projects</td><td>Array</td><td>List of all work zone projects managed by this vendor. See Work Zone Project.</td><td>Required</td></tr>
<tr><td>Sample Call</td><td colspan="4">$ curl
-H ‘Authorization: Basic c3d6TWFuYWdlcjptYXNzZG90’
-H ‘Accept: application/json’
https://{root}/api/v1/workZoneProjects</td></tr>
<tr><td>Success Response</td><td colspan="4">Code: 200, Content (json):
{
“update_date”: “20200101T070000+z”,
“work_zone_projects”: [
{…}, (See Work Zone Project)
…
]
}</td></tr>
<tr><td>Error Response</td><td colspan="4">Code: 401 UNAUTHORIZED, Content {json}:
{
“error”: “Invalid User Credentials”
}</td></tr>
</table>

4.3.1. Work Zone Project

Field Name	JSON Type	Description	Conformance
update_date	String	ISO 8601 formatted UTC date/time indicating the last date and time at which the details of this project were updated.	Required
project_id	String	The unique project ID (e.g. “Wilson Bridge”). This ID maps directly to the WZDx road event “subidentifier” element.	Required
description	String	Overall description of this project.	Required
start_date	String	ISO 8601 formatted UTC date indicating the start date of the project (e.g. “*yyyymmdd*T*hhmmss*+z”)	Required
end_date	String	ISO 8601 formatted UTC date indicating the end date of the project (e.g. “*yyyymmdd*T*hhmmss*+z)	Required
district	String	The district associated with the project.	Required
comments	String	Free-form comments/notes.	Optional
contractor	Object	Details about the prime contractor responsible for the work zone project. See **Contractor Information**.	Required

4.3.2. Contractor Information

Field Name	JSON Type	Description	Conformance
name	String	Name of the smart work zone contractor.	Required
contact_name	String	Primary contact name.	Required
contact_phone	String	Primary contact phone number.	Required
contact_email	String	Primary contact email address.	Required
alternate_contact_name	String	Alternate contact name.	Optional
alternate_contact_phone	String	Alternate contact phone number.	Optional
alternate_contact_email	String	Alternate contact email address.	Optional
contractor_url	String	Contractor company URL.	Optional

4.4. ROAD EVENTS

The API shall offer an endpoint that provides the details of all road events (i.e. work zone activities) in the vendor system. The response returned from this API request will be fully conformant with the WZDx v2.0 specification.

Function	Road Events Request			
Endpoint	/roadEvents			
HTTP Methods	GET			
Headers	Authorization: Basic Base64(*username:password*) Accept: application/json			
URL Parameters	n/a			
Request Data	n/a			
Response Object	**Field Name**	**JSON Type**	**Description**	**Conformance**
	update_date	Object	ISO 8601 formatted UTC date/time indicating the last date and time at which the list of road events was updated (e.g. "*yyyymmdd*T*hhmmss*+z").	Required
	road_events	Array	List of all road events managed by this vendor (see ***Road Event***)	Required

Sample Call	`$ curl`
	`-H 'Authorization: Basic c3d6TWFuYWdlcjptYXNzZG90'`
	`-H 'Accept: application/json'`
	`https://{root}/api/v1/roadEvents`			

Success Response Example	`Code: 200, Content (json):`
	`{`
	`"update_date": "20200101T070000+z",`
	`"road_events": [`
	`{…}, (See Road Event)`
	`…`
	`]`
	`}`			

Error Response	`Code: 401 UNAUTHORIZED, Content {json}: { "error": "Invalid User Credentials" }`			

4.4.1. Type of Work (from WZDx v2.0)

Field Name	JSON Type	Description	Conformance
type_name	String	A high-level text description of the type of work being done. See Work Type Name Enumerated Type from WZDx 2.0.	Required
is_architectural_change	Boolean	A flag indicating whether the type of work will result in an architectural change to the roadway. See here for details.	Optional

4.4.2. Lane (from WZDx v2.0)

Field Name	JSON Type	Description	Conformance
lane_edge_reference	String	The roadside edge from which lane are assigned numbers. See Lane Edge Reference Enumerated Type from WZDx 2.0. NOTE: this value must be the same for all lanes!	Required
lane_number	Number	The number assigned to a lane (starting from 1, counting away from the lane edge reference).	Required*
lane_status	String	Status of the lane for the traveling public. See Lane Status Enumerated Type from WZDx 2.0.	Required
lane_type	String	Type of the lane in accordance with Lane Type Enumerated Type from WZDx 2.0.	Required
lane_restrictions	Array	A list of one or more Lane Restriction objects.	Optional

** Indicates items are marked as Optional in the WZDx specification but are required by this API.*

4.4.3. Lane Restriction (from WZDx v2.0)

Field Name	JSON Type	Description	Conformance
restriction_type	String	The specific lane restriction in accordance with Road Restriction Enumerated Type from WZDx v2.0.	Required
restriction_value	Number	A numerical value describing the restriction, if applicable to the type of restriction. (E.g. “reduced-width” restriction may specify a value of 10 and units of ‘feet’)	Optional
restriction_unit	String	The units of measure for the lane restriction type in accordance with Lane Restriction Unit Enumerated Type from WZDx v2.0.	Conditional

4.4.4. Road Event (from WZDx v2.0)

Field Name	JSON Type	Description	Conformance
road_event_id	String	A unique ID of this road event/work zone activity	Required
subidentifier	String	If specified, must be the same as the “project_id” field on the ***Work Zone Project***.	Required*
geometry_type	String	GeoJSON geometry type. Must be either “LineString” or “Multipoint”.	Required
geometry	Array	A collection of 2 or more locations in accordance with the GeoJSON specification. When only the start and end location of the road event/activity is provided, the “type” field must be “Multipoint”. When more than 2 locations of the road event/activity path are provided, the “geometry_type” field must be “LineString”.	Required
road_name	String	Publicly known name of the road on which the event occurs.	Required
road_number	String	The road number designated by a jurisdiction such as a county, state or interstate. Examples: I-5, VT 133.	Optional
direction	String	The digitization direction of the road that is impacted by the event. Example North (for I-5 North). This field is restricted to the Direction Enumerated Type from WZDx 2.0.	Required
beginning_cross_street	String	Name or number of the nearest cross street along the roadway where the event begins.	Optional
ending_cross_street	String	Name or number of the nearest cross street along the roadway where the event ends.	Optional
beginning_milepost	String	The linear distance measured against a milepost marker along a roadway where the event begins.	Optional
ending_milepost	String	The linear distance measured against a milepost marker along a roadway where the event ends.	Optional

beginning_accuracy	String	Either “Estimated” or “Verified”. Indicates how the beginning coordinate was defined. See Spatial Verification Enumerated Type from WZDx v2.0.	Required
ending_accuracy	String	Either “Estimated” or “Verified”. Indicates how the ending coordinate was defined. See Spatial Verification Enumerated Type from WZDx v2.0.	Required
start_date	String	The UTC time and date when the event begins, in ISO 8601 format.	Required
end_date	String	The UTC time and date when the event ends, in ISO 8601 format.	Required
start_date_accuracy	String	Either “Estimated” or “Verified.” A measure of how accurate the start Date Time is. See Time Verification Enumerated Type from WZDx v2.0	Required
end_date_accuracy	String	Either “Estimated” or “Verified.” A measure of how accurate the end Date Time is. See Time Verification Enumerated Type from WZDx v2.0	Required
event_status	String	The status of the event. See Event Status Enumerated Type from WZDx v2.0.	Optional
total_num_lanes	Number	The total number of lanes associated with the road segment designated by the event geometry.	Required*
vehicle_impact	String	The impact to vehicular lanes along a single road in a single direction. See Vehicle Impact Enumerated Type from WZDx v2.0.	Required
workers_present	Boolean	A flag indicating that there are workers present in the event space.	Optional
reduced_speed_limit	Number	The reduced speed limit posted within the event space.	Optional
restrictions	Array	One or more road restrictions applying to the work zone road segment associated with the work zone. See Road Restriction Enumerated Type from WZDx v2.0.	Optional
description	String	Short, free text description of work zone.	Optional
issuing_organization	String	The organization issuing the data feed.	Optional
creation_date	String	The UTC time and date when the activity or event was created, in ISO 8601 format.	Optional
update_date	String	The UTC time and date when the activity or event was updated, in ISO 8601 format.	Required*
types_of_work	Array	A list of 1 or more Type Of Work objects.	Required
lanes	Array	A list of N Lane(***s***) where N = total_num_lanes.	Required

** Indicates items are marked as Optional in the WZDx specification but are required by this API.*

4.5. FIELD DEVICES CONFIGURATION

The API shall offer an endpoint that provides the configuration details of all field in the vendor system. This API request returns static, configuration data associated with each field device, not the dynamic operational state, which is retrieved using a different request (see Field Devices Operational State).

Function	Field Devices Request			
Endpoint	/fieldDevices			
HTTP Methods	GET			
Headers	Authorization: Basic Base64(*username:password*) Accept: application/json			
URL Parameters	n/a			
Request Data	n/a			
Response Object	**Field Name**	**JSON Type**	**Description**	**Conformance**
	update_date	String	ISO 8601 formatted UTC date/time indicating the last date and time at which the list of field devices was updated (e.g. "*yyyymmdd*T*hhmmss*+z").	Required
	field_devices	Array	List of all field devices managed by this vendor. See ***Field Device***.	Required

Sample Call	`$ curl`
	`-H 'Authorization: Basic c3d6TWFuYWdlcjptYXNzZG90'`
	`-H 'Accept: application/json'`
	`https://{root}/api/v1/fieldDevices`			

Success Response Example	`Code: 200, Content (json):`
	`{`
	`"update_date": "20200101T070000+z",`
	`"field_devices": [`
	`{…},` *(See **Field Device**)*
	`...`
	`]`
	`}`			

Error Response	`Code: 401 UNAUTHORIZED, Content {json}:`
	`{`
	`"error": "Invalid User Credentials"`
	`}`			

4.5.1. Field Device

Field Name	JSON Type	Description	Conformance
device_type	String	One of “dms”, “cctv”, or “vds”	Required
device_id	String	The globally (within the context of a vendor) unique ‘fixed’ device ID. This ID does not have to have meaning to a general user of the system as they will typically refer to it by the ‘device_name’.	Required
road_event_id	String	The ID of the road event this device is associated with.	Required
device_name	String	User friendly name of the device (e.g. “VDS-A-SB”).	Required
latitude	Number	EPSG 4326 latitude value of device location.	Required
longitude	Number	EPSG 4326 longitude value of device location.	Required
road_name	String	Roadway name of device location.	Required
road_number	Number	The road number designated by a jurisdiction such as a county, state or interstate of the device location. Examples I-5, VT 133.	Optional
direction	String	The roadway direction of the device location; see Direction Enumerated Type from WZDx 2.0.	Required
milepost	Number	The linear distance measured against a milepost marker along a roadway where the device is located.	Required
device_properties	Object	A collection of key/value pairs unique to a specific device type. Either ***DMS Properties*** or ***CCTV Properties*** depending on if device type is “dms” or “cctv” respectively.	Required
update_date	String	The UTC time and date when the device configuration details were updated, in ISO 8601 format.	Required

4.5.2. DMS Properties

Field Name	JSON Type	Description	Conformance
sign_width_pixels	Number	Width of the sign in units of pixels. Numeric value as per TMDD.	Required
sign_height_pixels	Number	Height of the sign in units of pixels. Numeric value as per TMDD.	Required
is_sign_control_supported	Boolean	Indicates whether or not the optional camera preset control is supported by the devices deployed for this project. Defaults to ‘false’ if not specified.	Optional.

4.5.3. CCTV Properties

Field Name	JSON Type	Description	Conformance
presets	Array	Array of zero or more ***CCTV Preset***s for the CCTV device.	Required
is_live_video_supported	Boolean	Indicates whether or not the optional live video feed is supported by the devices deployed for this project. Defaults to ‘false’ if not specified.	Optional
is_camera_control_supported	Boolean	Indicates whether or not the optional camera preset control is supported by the devices deployed for this project. Defaults to ‘false’ if not specified.	Optional

4.5.4. CTTV Preset

Field Name	JSON Type	Description	Conformance
preset_number	Number	Array of zero or more ***CCTV Preset***s for the CCTV device.	Required
description	String	Description of the preset.	Optional

4.6. ROAD EVENT DYNAMIC METRICS

The API shall offer an endpoint that provides a list of dynamic metrics for all of the road events (work zone activities) managed by the vendor.

<table>
<tr><th>Function</th><th colspan="4">Activity Metrics Request</th></tr>
<tr><td>Endpoint</td><td colspan="4">/roadEvents/dynamicMetrics</td></tr>
<tr><td>HTTP Methods</td><td colspan="4">GET</td></tr>
<tr><td>Headers</td><td colspan="4">Authorization: Basic Base64(username:password)
Accept: application/json</td></tr>
<tr><td>URL Parameters</td><td colspan="4">n/a</td></tr>
<tr><td>Request Data</td><td colspan="4">n/a</td></tr>
<tr><td>Response Object</td><td>Field Name</td><td>JSON Type</td><td>Description</td><td>Conformance</td></tr>
<tr><td></td><td>update_date</td><td>String</td><td>ISO 8601 formatted UTC date/time indicating the last date and time at which the list of dynamic road event metrics was updated (e.g. “yyyymmddThhmmss+z”)</td><td>Required</td></tr>
<tr><td></td><td>update_rate</td><td>Number</td><td>How often the data will be updated on the vendor side in seconds (e.g. 60). This provides clients with a sense of how often they should be polling the API endpoint for data.</td><td>Required</td></tr>
<tr><td></td><td>dynamic_metrics</td><td>Array</td><td>List of all road event (work zone activity) metrics. See Road Event Metrics and all associated sub-frame details below.</td><td>Required</td></tr>
<tr><td>Sample Call</td><td colspan="4">$ curl
-H ‘Authorization: Basic c3d6TWFuYWdlcjptYXNzZG90’
-H ‘Accept: application/json’
https://{root}/api/v1/roadEvents/dynamicMetrics</td></tr>
<tr><td>Success Response Example</td><td colspan="4">Code: 200, Content (json):
{
“update_date”: “20200101T070000+z”,
“update_rate”: 60,
“dynamic_metrics”: [
{…}, (See Road Event Metrics)
…
]
}</td></tr>
<tr><td>Error Response(s)</td><td colspan="4">Code: 401 UNAUTHORIZED, Content {json}:
{
“error”: “Invalid User Credentials”
}</td></tr>
</table>

4.6.1. Road Event Dynamic Metrics

Field Name	JSON Type	Description	Conformance
road_event_id	String	The unique road event (work zone activity) ID (e.g. “Wilson Bridge - Deck”). This ID maps directly to the WZDx work zone activity “road_event_id” element.	Required
road_event_update_date	String	ISO 8601 formatted UTC date/time indicating the last date and time at which the parent road event details were updated. This value is the same as the “update_date” value in the ***Road Event***. Clients can use this to determine whether or not there have been any changes to the largely static road event information and thus should re-fetch this information.	Required
update_date	String	The UTC time and date when the metrics were updated, in ISO 8601 format.	Required
volume	Number	Volume measured in vehicles per hour	Require
travel_time	Number	Travel time in seconds through the work zone activity. As per TMDD “link-travel-time”.	Required
delay	Number	Delay in seconds through the work zone activity. As per TMDD “link-delay”.	Optional
average_speed	Number	Average speed through the work zone activity in KPH as per TMDD “link-speed-average”.	Required
capacity	Number	Capacity in vehicles-per-hour as per TMDD “link-capacity”.	Optional
speed_limit	Number	Free flow speed through the work zone activity in KPH as per TMDD “link-speed-limit”.	Required
queue_length	Number	Length of queue formed upstream of the work zone activity start location, specified in meters.	Optional
average_occupancy	Number	Average occupancy throughout the work zone activity specified as a percent.	Optional

4.7. FIELD DEVICES DATA

These device data endpoints provide a means by which a client can monitor the operational state and status details of the various field devices deployed for the SWZ installation.

4.7.1. Field Device State

The below Field Device State object is used in the response to all field device data requests described in section 4.7, regardless of device type.

Field Name	JSON Type	Description	Conformance
device_id	String	The unique 'fixed' device ID (e.g. "DMS-1234ab56").	Required
road_event_id	String	The ID of the road event this device is associated with.	Required
road_event_update_date	String	ISO 8601 formatted UTC date/time indicating the last date and time at which the parent road event details were updated. This value is the same as the "update_date" value in the ***Road Event***. Clients can use this to determine whether or not there have been any changes to the largely static road event information and thus should re-fetch this information.	Required
status	String	Device status as per TMDD "device-status".	Required
error	String	Device error as per TMDD "device-error".	Required
update_date	String	ISO 8601 formatted UTC date/time with time zone indicating the date and time at which the device data was last updated.	Required

4.7.2. Dynamic Message Signs (DMS)

Function	Field Device State Request			
Endpoint	/fieldDevices/dms/data			
HTTP Methods	GET			
Headers	Authorization: Basic Base64(*username:password*) Accept: application/json			
URL Parameters	n/a			
Request Data	n/a			
Response Object	**Field Name**	**JSON Type**	**Description**	**Conformance**
	update_date	String	ISO 8601 formatted UTC date/time indicating the last date and time at which the device data was updated (e.g. "*yyyymmdd*T*hhmmss*+z")	Required
	update_rate	Number	How often the data will be updated on the vendor side in seconds (e.g. 60). This provides clients with a sense of how often they should be polling the API endpoint for this type of data.	Required
	dms_data	Array	An array of ***DMS Data*** objects. One object for every DMS configured by this vendor.	Required

Sample Call	`$ curl`
	`-H 'Authorization: Basic c3d6TWFuYWdlcjptYXNzZG90'`
	`-H 'Accept: application/json'`
	`https://{root}/api/v1/fieldDevices/dms/data`			

Success Response	`Code: 200, Content (json):` `{`			

Example	"update_date": "20200101T070000+z", "update_rate": 60, "dms_data": [{…}, (See ***DMS Data***) …] }

Error Response	Code: 401 UNAUTHORIZED, Content {json}:
	{
	"error": "Invalid User Credentials"
	}

DMS Data

Field Name	JSON Type	Description	Conformance
device_state	Object	The **Device State** details for this particular DMS.	Required
message_data	Object	The details of the message displayed on this DMS. See **Message Data**.	Required

Message Data

Field Name	JSON Type	Description	Conformance
current_message_multi	String	The current message displayed, specified as an NTCIP multi-string. (e.g. "[fo1]CAUTION[nl]SLOW[nl]AHEAD")	Required
message_set_date	String	ISO 8601 formatted UTC date/time with time zone indicating the date and time at which the message posted.	Required
message_source	String	The source of the message currently posted on the sign. This may be one of the following: • "external": when a client of the vendor system has commanded the sign to post a specific message via this API. • "internal": when the vendor system itself is the source of the posted message. • "field": when the posted message has been set locally in the field at the device itself.	Required

4.7.3. Vehicle Detection Sensors (VDS)

Function	Field Device State Request			
Endpoint	/fieldDevices/vds/data			
HTTP Methods	GET			
Headers	Authorization: Basic Base64(*username:password*) Accept: application/json			
URL Parameters	n/a			
Request Data	n/a			
Response Object	**Field Name**	**JSON Type**	**Description**	**Conformance**
	update_date	String	ISO 8601 formatted UTC date/time indicating the last date and time at which the device data was updated (e.g. "*yyyymmdd*T*hhmmss*+z")	Required
	update_rate	Number	How often the data will be updated on the vendor side in seconds (e.g. 60). This provides clients with a sense of how often they should be polling the API endpoint for this type of data.	Required
	vds_data	Array	An array of ***VDS Data*** objects. One object for every VDS configured by this vendor.	Required

Sample Call	`$ curl`
	`-H 'Authorization: Basic c3d6TWFuYWdlcjptYXNzZG90'`
	`-H 'Accept: application/json'`
	`https://{root}/api/v1/fieldDevices/vds/data`			

Success Response Example	`Code: 200, Content (json):`
	`{`
	`"update_date": "20200101T070000+z",`
	`"update_rate": 60,`
	`"vds_data": [`
	`{…},` (See ***VDS Data***)
	`…`
	`]`
	`}`			

Error Response	`Code: 401 UNAUTHORIZED, Content {json}:`
	`{`
	`"error": "Invalid User Credentials"`
	`}`			

VDS Data

Field Name	JSON Type	Description	Conformance
device_state	Object	The **Device State** details for this particular DMS.	Required
traffic_data	Object	The details of the traffic data measured by this VDS. See **Traffic Data**.	Required

Traffic Data

Field Name	JSON Type	Description	Conformance
from_date	String	ISO 8601 formatted UTC date/time with time zone indicating the start of the period that defines the traffic data collection interval.	Required
to_date	String	ISO 8601 formatted UTC date/time with time zone indicating the end of the period that defines the traffic data collection interval.	Required
vehicle_count	Number	Number of vehicles counted at this location, across all lanes, during the from/to interval. As per TMDD “detector-vehicle-count”.	Required
average_speed	Number	Average speed measured at this location, across all lanes, during the from/to interval. Specified in KMH as per TMDD “detector-speed”.	Required
average_occupancy	Number	Average occupancy measured at this location, across all lanes, during the from/to interval. Specified as a percent as per TMDD “detector-occupancy”.	Optional
lane_level_data	Array	A list of 0, 1 or more ***Lane Level Traffic Data*** objects.	Optional

Lane Level Traffic Data

Field Name	JSON Type	Description	Conformance
lane_edge_reference	String	The roadside edge from which lane are assigned numbers. See Lane Edge Reference Enumerated Type from WZDx 2.0.	Required
lane_number	Number	The number assigned to a lane (starting from 1, counting away from the lane edge reference).	Required
vehicle_count	Number	Number of vehicles counted at this location, in the specified, during the from/to interval. As per TMDD “detector-vehicle-count”.	Required
speed	Number	The speed measured at this location, in the specified lane, during the from/to interval. Specified in KPH as per TMDD “detector-speed”.	Required
Occupancy	Element	The occupancy measured at this location, in the specified lane, during the from/to interval. Specified as a percent as per TMDD “detector-occupancy”.	Optional

4.7.4. CCTV Cameras

Function	Field Device State Request			
Endpoint	/fieldDevices/cctv/data			
HTTP Methods	GET			
Headers	Authorization: Basic Base64(*username:password*) Accept: application/json			
URL Parameters	n/a			
Request Data	n/a			
Response Object	**Field Name**	**JSON Type**	**Description**	**Conformance**
	update_date	String	ISO 8601 formatted UTC date/time indicating the device data was last updated (e.g. "*yyyymmdd*T*hhmmss*+z")	Required
	update_rate	Number	How often the data will be updated on the vendor side in seconds (e.g. 60).	Required
	cctv_data	Array	An array of **CCTV *Data*** objects. One object for every camera configured by this vendor.	Required

Sample Call	`$ curl`
	`-H 'Authorization: Basic c3d6TWFuYWdlcjptYXNzZG90'`
	`-H 'Accept: application/json'`
	`https://{root}/api/v1/fieldDevices/cctv/data`			

Success Response Example	`Code: 200, Content (json):`
	`{`
	`"update_date": "20200101T070000+z",`
	`"update_rate": 60,`
	`"cctv_data": [`
	`{…}, (See **CCTV Data**)`
	`…`
	`]`
	`}`			

Error Response	`Code: 401 UNAUTHORIZED, Content {json}:`
	`{`
	`"error": "Invalid User Credentials"`
	`}`			

CCTV Data

Field Name	JSON Type	Description	Conformance
device_state	Object	The **Device State** details for this particular camera.	Required
image_data	Object	The details of the last image captured by this camera. See **Image Data**.	Required

Image Data

Field Name	JSON Type	Description	Conformance
camera_preset_number	Number	The current preset number.	Optional
encoded_image	String	Base64 encoding of the JPEG image bytes.	Optional
image_capture_date	String	ISO 8601 formatted UTC date/time with time zone indicating when the image was captured. (Required only if encoded_image is provided.)	Conditional

4.8. LIVE VIDEO STREAMING *(OPTIONAL)*

To be determined.

4.9. DMS MESSAGE *(OPTIONAL)*

The API shall offer an endpoint that allows a client to post (PUT) a message to a DMS, release (DELETE) a user-posted messaged from a DMS, or view (GET) the message posted on a DMS.

Function	DMS Message Display Request			
Endpoint	/fieldDevices/dms/:deviceId/message			
HTTP Methods	GET (retrieve the message currently posted to the DMS) PUT (post a message to the DMS) DELETE (release a user-posted message)			
Headers	Authorization: Basic Base64(*username:password*) Accept: application/json			
URL Parameters	**Parameter Name**	**Description**		**Conformance**
	deviceId	Device ID of DMS to interact with.		Required
Request Data (for PUT only)	**Field Name**	**JSON Type**	**Description**	**Conformance**
	message_multi	String	MULTI message to post to DMS.	Required
Response Object	{…} *See* ***DMS Message Response***			

Sample Call	`$ curl -X PUT`
	`-H 'Authorization: Basic c3d6TWFuYWdlcjptYXNzZG90'`
	`-H 'Accept: application/json'`
	`-D '{ "message_multi": "[fo1]CAUTION[nl]SLOW[nl]AHEAD" }'`
	`https://{root}/api/v1/fieldDevices/dms/dms-1293/message`			

Success Response Example	`Code: 200, Content (json):`
	`{`
	`"device_id": "dms-1293",`
	`"road_event_id: "wilbert-bridge-north",`
	`"message_multi": "[fo1]CAUTION[nl]SLOW[nl]AHEAD"`
	`}`			

Error Response(s)	If credentials are invalid:
	`Code: 401 UNAUTHORIZED, Content {json}:`
	`{`
	`"error": "Invalid User Credentials"`
	`}`
	
	If the device ID is invalid, or the requested message is invalid (bad format, too large, etc.):
	`Code: 400 BAD REQUEST, Content {json}:`
	`{`
	`"error": "Invalid Request Format"`
	`}`
	
	If the device does not support message posting:
	`Code: 501 NOT SUPPORTED Content {json}:`
	`{`
	`"error": "DMS Control Is Not Supported"`
	`}`			

4.9.1. DMS Message Response

Field Name	JSON Type	Description	Conformance
device_id	String	The unique device ID of the DMS (e.g. “Sign_1”).	Required
road_event_id	String	The unique activity ID (e.g. “Wilson Bridge - Deck”) associated with this sign.	Required
message_multi	String	The NTCIP-MULTI formatted string representing: • For GET requests, the posted message. • For PUT requests, the requested message that is queued for posting. • For DELETE requests, this field shows the message that was removed (as confirmation).	Required

4.10.CCTV PRESET *(OPTIONAL)*

The API shall offer an endpoint that allows a client to set (PUT) a CCTV camera to a particular preset or get (GET) the preset set on the CCTV.

Function	DMS Message Release Request			
URL	/fieldDevices/cctv/:deviceId/preset			
HTTP Methods	GET (view the preset currently set on the CCTV) PUT (set the preset on the CCTV)			
Headers	Authorization: Basic Base64(*username:password*) Accept: application/json			
URL Parameters	**Parameter Name**	**Description**		**Conformance**
	deviceId	Device ID of CCTV to interact with.		Required
Request Data	**Field Name**	**JSON Type**	**Description**	**Conformance**
	preset_number	Number	The number of one of the configured preset positions for the specified CCTV.	Required
Response Object	{…} (See ***CCTV Preset Response***)			

Sample Call	`$ curl -X PUT`
	`-H 'Authorization: Basic c3d6TWFuYWdlcjptYXNzZG90'`
	`-H 'Accept: application/json'`
	`-D '{ "preset_number": 1 }'`
	`https://{root}/api/v1/cctv/cctv-12-purple/preset`			

Success Response	`Code: 200, Content (json):`
	`{`
	`"device_id": "cctv-12-purple",`
	`"road_event_id: "wilbert-bridge-north",`
	`"preset_number": 1`
	`}`			

Error Response(s)	If credentials are invalid:
	`Code: 401 UNAUTHORIZED, Content {json}:`
	`{`
	`"error": "Invalid User Credentials"`
	`}`
	
	If the device ID is invalid, or the requested preset number is invalid:
	`Code: 400 BAD REQUEST, Content {json}:`
	`{`
	`"error": "Invalid Request Format"`
	`}`
	
	If the device does not support preset control:
	`Code: 501 NOT SUPPORTED Content {json}:`
	`{`
	`"error": "CCTV Control Is Not Supported"`
	`}`			

4.10.1. CCTV Preset Response

Field Name	JSON Type	Description	Conformance
device_id	String	The unique camera ID (e.g. “CCTV_1.ABC-Dynamo”).	Required
road_event_id	String	The unique activity ID (e.g. “Wilson Bridge - Deck”) associated with this camera.	Required
preset_number	Number	The camera preset number: • For GET requests, the current preset number. • For PUT requests, the requested preset number that the CCTV is queued to move to (setting on device may not be immediate).	Required