Massachusetts Department of Transportation (MassDOT)
Work Zone Manager (WZM) System

Appendix A – Vendor API Specification v4.0

2022-02-24 — Revision 1.1

IBI GROUP

TABLE OF CONTENTS

1. Introduction 2
2. Security 3
2.1. Authorization Key Generation 3
2.2. Authorization Key Usage 3
2.2.1. Authorization Key Usage Example 3
3. Work Zone Definition 4
4. API Requests 5
4.1. Endpoint Summary 5
4.2. Vendor Information 6
4.2.1. Vendor Information 6
4.3. Work Zone Projects 8
4.3.1. Work Zone Project 9
4.3.2. Contractor Information 10
4.4. WZDxFeed (Work Zone Road Events) 11
4.5. SwzDeviceFeed (Field Devices) 12
4.6. Road Event Metrics 13
4.6.1. Road Event Metrics 14

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 Work Zone Manager (WZM) software. This API allows the WZM software to centrally monitor and optionally control one or more SWZ implementations. All functions of this API not marked as "*optional*" are required to be provided by the implementer of the API to be considered compliant with the specification.

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 work zone definition and field device objects as defined by the Work Zone Data Exchange (WZDx) effort of FHWA's overall Work Zone Data Initiative. The Vendor API references via hyperlink enumerated types from the WZDx specification. Note that enumerated type values marked as "deprecated" in WZDx should not be used.

All mentions herein to "WZDx v4.0" refer to the documents that can be found at the following URL:

https://github.com/usdot-jpo-ode/jpo-wzdx/releases/tag/v4.0

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 API shall 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 WZM software is considered the client and the vendor application supporting the Vendor API is considered the server. All authorized clients shall be provided a ‘username’ and ‘password’ so that they can authenticate and use the API.

2.1. Authorization Key Generation

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

1. Concatenate ‘username’ and ‘password’ separated by a colon: *username:password*
2. 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).

2.2. 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)
```

2.2.1. Authorization Key Usage Example

```
Username: swzManager
Password: password
```

Authorization header value:

```
base64(”swzManager:password”) = c3d6TWFuYWdlcjpwYXNzd29yZA==
```

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 “password”).
3. Validate the received username and password.

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

3. WORK ZONE DEFINITION

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 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 the vendor's application.
- 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 multiple directions of travel.
    - Each direction of travel will contain 1 or more **road events**.
    - Each road event within a work zone project should have a unique name that distinguishes it from all other road events managed by the vendor's application, as well as a reference the work zone project to which the road event belongs.
        - All road event names should start with the name of the project to which they belong. For example: Project Name = "P1"… Road Event Name = "P1-RE1" or "P1_Event1", etc.
    - Each direction of travel must be represented by a separate road event. For example, if there is a work zone on a roadway with two lanes, each in an opposite travel direction, each direction must be a separate road event.
    - A change in lane configuration or roadway should result in creating a new road event, that is, separating the work zone into multiple road events.
    - Construction which requires alternating traffic flows within a lane must be represented with at least one road event in each direction. Similarly, roadways that during normal operation have an alternating flow of traffic direction based on time of day must be represented by at least one road event in each direction.

4. API REQUESTS

The API shall follow a request/response model where all communications are initiated by the client (i.e. the WZM software). Unless specified otherwise, the SWZ vendor applications shall only respond to authorized requests from the WZM 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 “4.0” (v4.0) of the Vendor API. The major version of the Vendor API will match the major version of WZDx specification from which it is based. Minor versions can evolve independently.

4.1. Endpoint Summary

Request Endpoint	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.
/roadEventMetrics	GET	Provides traffic static and dynamic traffic metrics for all road events in the vendor system.
/wzdxFeed	GET	Provides high-level information about work zone (and detour) road events occurring on roadways.
/swzDeviceFeed	GET	Provides information (location, status, live data) about field devices deployed on or near the roadway in work zones.

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/v4.0/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.

Function	Work Zone Projects Request			
Endpoint	/workZoneProjects			
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 projects was updated (e.g. "*yyyymmdd*T*hhmmss*Z"). This would include any of the following: • New project(s) added • Existing project(s) changed (any data associated with project) Project(s) deleted	Required
	work_zone_projects	Array	List of all work zone projects managed by this vendor. See ***Work Zone Project.***	Required

Sample Call	`$ curl`
	`-H 'Authorization: Basic c3d6TWFuYWdlcjptYXNzZG90'`
	`-H 'Accept: application/json'`
	`https://{root}/api/v4.0/workZoneProjects`			

Success Response	`Code: 200, Content (json):`
	`{`
	`"update_date": "20200101T070000Z",`
	`"work_zone_projects": [`
	`{ See Work Zone Project }`			

```
                    …
                ]
            }
```

Error Response

```
Code: 401 UNAUTHORIZED, Content {json}:
{
   “error”: “Invalid User Credentials”
}
```

4.3.1. Work Zone Project

Field Name	JSON Type	Description	Conformance
id	String	A Universally Unique Identifier (following RFC 4122) for the work zone project.	Required
name	String	A human-readable name for the project.	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*”)	Required
end_date	String	ISO 8601 formatted UTC date indicating the end date of the project (e.g. “*yyyymmdd"*)	Required
region	String	The region (district) the project is associated with (e.g. "District 1").	Required
road_event_ids	Array	A list of road events that are part of (children of) this work zone project. These should correspond the ID of a "RoadEventFeature" in the Vendor's WZDxFeed.	Required
contractor	Object	Details about the prime contractor responsible for the work zone project. See **Contractor Information**.	Required
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
comments	String	Free-form comments/notes.	Optional

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. WZDxFeed (Work Zone Road Events)

The API shall offer an endpoint that provides the details of all work zone road events (i.e. work zone segments) in the vendor system. The response to this request is a valid WZDxFeed object as defined by the WZDx v4.0 Specification. The WZDx documentation provides a JSON Schema to facilitate validating the conformance of the response to this endpoint.

Function	WZDxFeed Request
Endpoint	/wzdxFeed
HTTP Methods	GET
Headers	`Authorization: Basic Base64(username:password)` `Accept: application/json`
URL Parameters	n/a
Request Data	n/a
Response Object	***WZDxFeed***

Sample Call	`$ curl`
	`-H 'Authorization: Basic c3d6TWFuYWdlcjptYXNzZG90'`
	`-H 'Accept: application/json'`
	`https://{root}/api/v4.0/wzdxFeed`

Success Response Example	`Code: 200, Content (json):`
	`{`
	See ***WZDxFeed***
	`}`

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

4.5. SwzDeviceFeed (Field Devices)

The API shall offer an endpoint that provides the configuration and operational state details of all field devices in the vendor system. The response to this request is a valid SwzDeviceFeed object as defined in the WZDx v4.0 Specification. The WZDx documentation provides a JSON Schema to facilitate validating the conformance of the response to this endpoint.

Function	SwzDeviceFeed Request
Endpoint	/swzDeviceFeed
HTTP Methods	GET
Headers	`Authorization: Basic Base64(username:password)` `Accept: application/json`
URL Parameters	n/a
Request Data	n/a
Response Object	**SwzDeviceFeed**

Sample Call	`$ curl`
	`-H 'Authorization: Basic c3d6TWFuYWdlcjptYXNzZG90'`
	`-H 'Accept: application/json'`
	`https://{root}/api/v4.0/swzDeviceFeed`

Success Response Example	`Code: 200, Content (json):`
	`{`
	See SwzDeviceFeed
	`}`

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

4.6. Road Event Metrics

The API shall offer an endpoint that provides a list of static and dynamic metrics for all road events managed by the vendor.

Function	Road Event Metrics Request			
Endpoint	/roadEventMetrics			
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 road event metrics was updated (e.g. "*yyyymmdd*T*hhmmss*Z")	Required
	update_frequency	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 data.	Required
	road_event_metrics	Array	List of all road event metrics. See Road Event Metrics and all associated sub-frame details below.	Required

Sample Call	`$ curl`
	`-H 'Authorization: Basic c3d6TWFuYWdlcjptYXNzZG90'`
	`-H 'Accept: application/json'`
	`https://{root}/api/v4.0/roadEventMetrics`			

Success Response Example	`Code: 200, Content (json):`
	`{`
	`"update_date": "20200101T070000Z",`
	`"update_frequency": 60,`
	`"road_event_metrics": [ { See Road Event Metrics } … ]`
	`}`			

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

4.6.1. Road Event Metrics

Field Name	JSON Type	Description	Conformance
road_event_id	String	The unique ID of the road event. This ID maps directly to the ***Road Event Feature*** "id" property.	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's ***Road Event Core Details***. Clients can use this to determine whether 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 that the metrics apply to, in ISO 8601 format.	Required
volume_vph	Number	Volume measured in vehicles per hour	Optional
travel_time_seconds	Number	Travel time in seconds through the road event. As per TMDD "link-travel-time".	Required
delay_seconds	Number	Delay in seconds through the road event. As per TMDD "link-delay".	Optional
average_speed_kph	Number	Average speed through the road event in KPH as per TMDD "link-speed-average".	Required
capacity_vph	Number	Capacity in vehicles-per-hour as per TMDD "link-capacity".	Optional
speed_limit_kph	Number	Free flow speed through the road event in KPH as per TMDD "link-speed-limit".	Required
queue_length_meters	Number	Length of queue formed upstream of the road event start location, specified in meters.	Optional
average_occupancy_percent	Number	Average occupancy throughout the road event specified as a percent.	Optional