

Dispatch1 Integration API

Last updated by | Saravanan Subramanian | Mar 25, 2024 at 5:47 PM EDT

Dispatch1 Integration API Documentation

Introduction

The Dispatch1 Integration API provides an interface for managing work orders within the Dispatch1 system. This API is used to create, update, and retrieve work order details, enabling seamless integration between Dispatch1 and external systems. This document outlines how to interact with the API endpoint responsible for handling work orders.

Base URL

The base URL for the API is: <https://api.example.com/v1/api/> ****Actual URL available soon.**

Workorder Endpoint Information

- **URL:** {baseUrl}/api/workorders
- **Method:** POST
- **Description:** Creates a new work order based on the provided information.

Model

The work order model consists of the following fields:

- `workOrderId` (string): The unique identifier for the work order sent by the client to refer back to their system.
- `issue` (string): A brief description of the issue or task.
- `statusType` (string): The status of the work order (e.g., Create, Revisit, Cancel)
- `scopeOfWork` (string): A description of the work to be performed, e.g., "Camera Install and possible PM."
- `contractId` (int): The contract identifier associated with the work order.
- `location` (object): An object containing location details.
 - `locationId` (int): The location identifier.
 - If new location then provide all other location-related fields
 - (Other location-related fields, all nullable)
- `startDateTime` (string): The start date and time for the work order.
- `specialInstructions` (string, nullable): Any special instructions for the work order.
- `poReferenceNumber` (string, nullable): The purchase order reference number.
- `serialNumber` (string, nullable): The serial number of the equipment or product.
- `parentWorkOrderId` (string, nullable): The identifier of the parent work order, if any.
- `clientId` (int): The client identifier.
- `endClientId` (int): The end client identifier.
- `ticketNumber` (string): This field is not part of the POST payload during the submission of the create work order. This will be sent back as a response once the work order is created successfully in Dispatch1 system. This number is a unique identifier to refer back to the work order in Dispatch1.

- `notes` (string, nullable): Any additional notes.
- `technicians` (array, nullable): A list of technician objects, if any.
- `technicianRequests` (array): An array of technician request objects.
 - `quantity` (int): The quantity of technicians requested.
 - `typeId` (int): The technician type identifier.
- `deliverables` (array, nullable): A list of deliverable objects, if any.
- `requiredTools` (string): Any tools required to complete the work order.
- `webhook` (nullable): Register a webhook to receive real-time updates on work order status changes.
 - `url` (string): The webhook URL provided by the client for communication of work order status updates.
 - `method` (string): The HTTP method for the webhook (POST/PUT).
 - `contentType` (string): The content type of the webhook request.
 - `headers` (string): Headers for the webhook request with comma separator values.
 - `authorizationKey` (string): Key for authorizing the webhook request.

Webhook Endpoints

The API uses webhooks to send real-time updates to clients when work order statuses change. Clients must provide a `webhookUrl` field in the request body when creating or updating a work order. The API will send a POST/PUT request to the provided webhook URL when the work order status changes. The webhook payload will contain the updated `WorkOrder` object.

Validation and Error Handling

The API validates the provided data in work order requests, such as the `location.locationId` if existing location. If it's a new location then the location address must include the [location.name](#), `location.address`, `location.city`, `location.state`, `location.zip`, and `location.country`. If the address is invalid, the API will not create a dispatch or project job and will respond with an HTTP 400 Bad Request Error.

Webhook Usage

Clients can use the webhooks to receive real-time updates on work order status changes. To use the webhooks, clients must provide a valid `webhookUrl` field in the request body when creating or updating a work order. The API will send a POST request to the provided webhook URL when the work order status changes. The webhook payload will contain the updated `WorkOrder` object.

Clients should implement a server-side endpoint to handle webhook POST requests from the Dispatch1Integration API. The endpoint should be capable of parsing the incoming JSON payload, extracting the updated `WorkOrder` object, and taking appropriate action based on the `statusType`.

Webhook Status Types

The following status types may be sent via webhooks:

- `Cancel`: Work order has been canceled.
- `Scheduled`: Work order has been scheduled and confirmed with the vendor.
- `WorkStarted`: Technician has checked in at the work location.
- `WorkEnded`: Technician has checked out of the work location.
- `DeliverablesCollected`: Deliverables have been collected, and the work order is complete.

Create Workorder

- **URL:** {baseUrl}/api/workorders
- **Method:** POST
- **Description:** create new workorder

Sample Request

```
{
  "workOrderId": "228",
  "issue": "POST Create Testing",
  "statusType": "Create",
  "scopeOfWork": "Camera Install and possible PM",
  "contractId": 123,
  "location": {
    "locationId": 123
  },
  "startDateTime": "01/02/2024 06:47:00",
  "clientId": 123,
  "endClientId": 123,
  "technicianRequests": [
    {
      "quantity": 1,
      "typeId": 8
    }
  ],
  "requiredTools": "No required Tools",
  "webhook": {
    "url": "http://test.com/updateworkorder",
    "method": "PUT",
    "contentType": "application/json",
    "headers": "Authorization: Basic {token},Id:{id}",
    "authorizationKey": "{token base64encoding value}"
  }
}
```

Sample Response

Error Response

Status Code: 400

Internal Server Error

Status Code: 500

Success Response

Status Code: 201

```

{
  "workOrderId": "228",
  "issue": "POST Create Testing",
  "statusType": "Create",
  "scopeOfWork": "Camera Install and posible PM",
  "contractId": 123,
  "location": {
    "locationId": 123,
    "locationName": null,
    "locationAddress": null,
    "locationCity": null,
    "locationState": null,
    "locationZip": null,
    "locationCountry": null,
    "locationContactFirstName": null,
    "locationContactLastName": null,
    "locationContactPhone": null,
    "locationContactEmail": null
  },
  "startDateTime": "01/02/2024 06:47 pm ",
  "specialInstructions": null,
  "poReferenceNumber": null,
  "serialNumber": null,
  "parentWorkOrderId": null,
  "clientId": 123,
  "endClientId": 123,
  "ticketNumber": "FSDP-123",
  "notes": null,
  "technicians": [
    {
      "name": "Tech 1",
      "phone": "(111) 111-1111",
      "scheduledDateTime": "2024-01-02 13:31:00.000",
      "startDateTime": "2024-01-02 13:15:00.000",
      "endDateTime": "2024-01-02 15:30:00.000"
    }
  ],
  "technicianRequests": [
    {
      "quantity": 1,
      "typeId": 8
    }
  ],
  "deliverables": [
    {
      "title": "Deliverable 1",
      "documentUrl": "https://www.google.com",
      "comments": "Deliverable 1 description",
      "timestamp": "2024-01-02 15:30:00.000"
    },
    {
      "name": "Deliverable 2",
      "documentUrl": "https://www.google.com",
      "comments": "Deliverable 2 description",
      "timestamp": "2024-01-02 15:30:00.000"
    }
  ],
  "requiredTools": "No required Tools",
  "webhook": {
    "url": "http://test.com/updateworkorder",
    "method": "PUT",
    "contentType": "application/json",
    "headers": "Authorization: Basic {token},Id:{id}"
  }
}

```

Update Workorder Endpoint Information:

- **URL:** {baseUrl}/api/workorders
- **Method:** PUT
- **Description:** PUT only support Cancel statusType.

Sample Request

```
{
  "workOrderId": "228",
  "issue": "POST Create Testing",
  "statusType": "Cancel",
  "scopeOfWork": "Camera Install and possible PM",
  "contractId": 123,
  "location": {
    "locationId": 123
  },
  "startDateTime": "01/02/2024 06:47 pm",
  "clientId": 123,
  "endClientId": 123,
  "ticketNumber": "FSDP-123",
  "technicianRequests": [
    {
      "quantity": 1,
      "typeId": 8
    }
  ],
  "requiredTools": "No required Tools"
}
```

Error Response

Status Code: 404

Internal Server Error

Status Code: 500

Success Response

Status Code: 200

```

{
  "workOrderId": "228",
  "issue": "POST Create Testing",
  "statusType": "Cancel",
  "scopeOfWork": "Camera Install and posible PM",
  "contractId": 123,
  "location": {
    "locationId": 123,
    "locationName": null,
    "locationAddress": null,
    "locationCity": null,
    "locationState": null,
    "locationZip": null,
    "locationCountry": null,
    "locationContactFirstName": null,
    "locationContactLastName": null,
    "locationContactPhone": null,
    "locationContactEmail": null
  },
  "startDateTime": "01/02/2024 06:47 pm ",
  "specialInstructions": null,
  "poReferenceNumber": null,
  "serialNumber": null,
  "parentWorkOrderId": null,
  "clientId": 123,
  "endClientId": 123,
  "ticketNumber": "FSDP-123",
  "notes": null,
  "technicians": [],
  "technicianRequests": [
    {
      "quantity": 1,
      "typeId": 8
    }
  ],
  "deliverables": [],
  "requiredTools": "No required Tools",
  "webhook": {
    "url": "http://test.com/updateworkorder",
    "method": "PUT",
    "contentType": "application/json",
    "headers": "Authorization: Basic {token},Id:{id}"
  }
}

```

Get Workorder Details

This function is used to get workorder information

Request

- **Method:** GET
- **URL:** {baseUrl}/api/workorders

Response

The response is a JSON object containing a list of workorders.

Success Response

Status Code: 200

Error Response

Status Code: 500

Content: Internal server error. Something went wrong during the processing of this request. Please try again.

Get Workorder Details By Ticketnumber

This function is used to get workorder by ticket number

Request

- **Method:** GET
- **URL:** {baseUrl}/api/workorders/{ticketNumber}

Response

The response is a JSON object containing a list of workorders.

Success Response

Status Code: 200

Error Response

Status Code: 404

Content: Resource not found

Status Code: 500

Content: Internal server error. Something went wrong during the processing of this request. Please try again.

Utilities

Get All EndClients

This function is used to get all end clients for a specific client.

Request

- **Method:** GET
- **URL:** {baseUrl}/api/utilities/endclients/{clientId}
- **Parameters:**
 - `clientId`: The ID of the client.

Response

The response is a JSON object containing a list of end clients.

Success Response

Status Code: 200

Content: List of end clients

Error Response

Status Code: 404

Content: Endclient information not found for the clientid.

Status Code: 500

Content: Internal server error. Something went wrong during the processing of this request. Please try again.

Sample Data:

```
[
  {
    "endClientId": 1,
    "endClientName": "End Client 1"
  },
  {
    "endClientId": 2,
    "endClientName": "End Client 2"
  }
]
```

Get All Country

This function is used to get all countries.

Request

- **Method:** GET
- **URL:** {baseUrl}/api/utilities/countries

Response

The response is a JSON object containing a list of countries.

Success Response

Status Code: 200

Content: List of countries

Error Response

Status Code: 500

Content: Internal server error. Something went wrong during the processing of this request. Please try again.

Sample Data:


```
[
  {
    "countryId": 1,
    "countryName": "Country 1"
  },
  {
    "countryId": 2,
    "countryName": "Country 2"
  }
]
```

Get All TechnicianTypes

This function is used to get all technician types.

Request

- **Method:** GET
- **URL:** {baseUrl}/api/utilities/techniciantypes

Response

The response is a JSON object containing a list of technician types.

Success Response

Status Code: 200

Content: List of technician types

Error Response

Status Code: 500

Content: Internal server error. Something went wrong during the processing of this request. Please try again.

Sample Data:

```
[
  {
    "technicianTypeId": 2,
    "technicianTypeName": "advanced technician"
  },
  {
    "technicianTypeId": 4,
    "technicianTypeName": "Smart Hands"
  }
]
```

Get All Location

This function is used to get all locations for a specific client.

Request

- **Method:** GET

- **Route:** {baseUrl}/api/utilities/locations/{clientId}
- **Parameters:**
 - ⊗ `clientId`: The ID of the client.

Response

The response is a JSON object containing a list of locations.

Success Response

Status Code: 200

Content: List of locations

Error Response

Status Code: 404

Content: Location not found for the clientId

Status Code: 500

Content: Internal server error. Something went wrong during the processing of this request. Please try again.

Sample Data:

```
[
  {
    "locationId": 1,
    "locationName": "Location 1",
    "locationAddress": "Address 1",
    "locationCity": "City 1",
    "locationState": "state 1",
    "locationZip": "Zip 1",
    "locationPhone": "phone 1",
    "countryName": "Country 1",
    "clientId": 1,
    "clientName": "Client 1"
  }
]
```

Conclusion

The Dispatch1Integration API is a powerful tool for managing work orders in the field services industry. By adhering to RESTful best practices and utilizing webhooks for real-time updates, the API provides a seamless integration experience for clients. This technical overview has covered the primary aspects of the API, including the WorkOrder resource, available endpoints, webhook functionality, and error handling. By leveraging the Dispatch1Integration API, clients can efficiently create, update, and manage work orders, leading to improved service delivery and customer satisfaction.