

# AutoSpot Batch API

## Overview

The AutoSpot Batch API provides HTTPS endpoints for transmitting data from source systems to AutoSpot. Source systems can call these endpoints to create a new Repair Order (POST) in AutoSpot and update an existing Repair Order (PATCH) in AutoSpot. Source systems authenticate with an API key.

## Data Schema

JSON is the expected format. The endpoints accept an array, and each object contains the `shop_id` at the top level. This allows source systems to send multiple Repair Orders for multiple shops in a single request. A minimum of one shop with one repair order is required. Note that the `obs_event_id` property is required if you want OBS to receive stage change notifications for the Repair Order.

The `document_links` property is a JSON array containing a list of document link objects. Each document link object contains a `url` property string with a value set to the Web accessible link to a document that will appear in the repair order detail view in the shop production board. The `type` and `description` properties are string values set to user defined values for the document type and description. The description is the value that appears as the name for the document in the repair order detail view list of documents associated to the repair order.

```
[
  {
    "shop_id": 33,
    "repair_orders":
    [
      {
        "repair_number": "606471",
        "scheduled_arrival": "2020-03-11T00:00:00Z",
        "scheduled_complete": "2020-03-
12T00:00:00Z", "scheduled_delivery": "2020-
03-13T00:00:00Z",
        "obs_event_id": "783A61C0-DDAA-4B11-A758475C419B5927",
        "rental_car": true,
        "total_estimated_hours":48,
        "document_links": [
          {"url":"<INSERT_URL_LINK_TO_A_DOCUMENT_ON_THE_WEB>",
            "type":"pdf", "description": "Parts Doc"},
          {"url":"<INSERT_URL_LINK_TO_A_DOCUMENT_ON_TH
E_WEB>", "type":"pdf", "description":"Inventory Doc"}
        ],
        "status": "Active",
        "vehicle": {
          "year": 2000,
          "model":
            "Silverado",
          "make":
            "Chevrolet",
          "color": "Red"
        },
        "customer": {
          "first_name": "Bilbo",
          "last_name":
            "Baggins",
          "email": "test@email.com",
          "phone_number":
            "5553337777",
```

```
},
"insurance": {
  "name": "State Farm"
},
"repair_team": [
  {
    "id": "4849843",
    "first_name": "Meriadoc",
    "last_name": "Brandybuck",
    "title": "Estimator"
  },
  "repair_plan": [
    {
      "stage_id": 1,
      "estimate": 12780,
      "supplemental_estimate": 10000
    },
    {
      "stage_id": 2,
      "estimate": 64800,
      "supplemental_estimate": 10
    },
    {
      "stage_id": 3,
      "estimate": 15210,
      "supplemental_estimate": 0
    },
    {
      "stage_id": 4,
      "estimate": 85500,
      "supplemental_estimate": 0
    },
    {
      "stage_id": 5,
      "estimate": 14400,
      "supplemental_estimate": 0
    }
  ]
}
```

## Adding a Repair Plan

The optional repair plan array can be added to a repair order to add estimates for each repair plan stage. The repair plan consists of an array of stage objects containing one of the Stage Ids listed in Autospot “stage\_ids” Table in the OBS Schema reference of this document and an estimate property specifying the stage labor hours estimate in minutes. If no repair plan is added to the repair order a default plan will be created in AutoSpot with no labor hours estimates.

## Updating Repair Orders

Notice that you don't need to pass all the properties when updating a Repair Order, just pass the piece of information that you need to update along with the repair\_number. See the sample request below for updating customer phone\_number only.

```
[
  {
    "shop_id": 33,
    "repair_orders": [
      {
        "repair_number": "606471",
      },
      "customer": {
        "phone_number": "5553337777",
      }
    ]
  }
]
```

When editing the repair\_team, you must send the entire repair team in the request body. If a specialist that is currently in the repair\_team is not sent in the request, it will be deleted.

When editing the document\_links, you must send the entire document\_links array in the request body. If a document link that is currently in the document\_links for a repair order is not sent in the request, it will be deleted. An empty document\_links array in the Patch request will delete all document links for the specified repair order.

## Updating Repair Order Repair Plans

The AutoSpot Batch API update **PATCH** endpoint includes the capability to update Repair Order labor hours.

The labor hours for a Repair Order can be updated in a **PATCH** request to the **/repair-orders** Batch API. Updated labor hours should be added to the

**"supplemental\_estimate"** property in

units of minutes to a stage in the **"repair\_plan"** array. In the example below, Stage 3 & 5 in the

show values added to the **"supplemental\_estimate"** property in minutes. The original **"estimate"** property remains unchanged from the original POST request value and is ignored in the update and can be omitted.

```
[
  {
    "shop_id": 1,
    "repair_orders": [
      {
        "repair_number": "606477",
        "scheduled_arrival": "2022-03-11T00:00:00Z",
        "scheduled_complete": "2022-03-12T00:00:00Z",
        "scheduled_delivery": "2022-03-13T00:00:00Z",
        "rental_car": true,
        "total_estimated_hours": 48,
        "vehicle": {
          "year": 2000,
          "model": "Silverado",
          "make": "Chevrolet",
          "color": "Blue"
        },
        "customer": {
          "first_name": "Millie",
          "last_name": "Jackson",
          "email": "test@detroitlabs.com",
          "phone_number": "5554449999"
        },
        "insurance": {
          "name": "State Farm"
        },
        "repair_team": [
          {
            "id": "4849843",
            "first_name": "Meriadoc",
            "last_name": "Brandybuck",
            "title": "Estimator"
          },
          {
            "id": "9794854",
            "first_name": "Peregrin",
            "last_name": "Took",
            "title": "Body Technician"
          }
        ]
      }
    ]
  }
],
```

```
"repair_plan": [  
  {  
    "stage_id": 1,  
    "estimate": 0,  
    "supplemental_estimate": 0  
  },  
  {  
    "stage_id": 2,  
    "estimate": 100,  
    "supplemental_estimate": 140  
  },  
  {  
    "stage_id": 3,  
    "estimate": 2,  
    "supplemental_estimate": 360  
  },  
  {  
    "stage_id": 4,  
    "estimate": 350,  
    "supplemental_estimate": 0  
  },  
  {  
    "stage_id": 5,  
    "estimate": 60,  
    "supplemental_estimate": 120  
  },  
  {  
    "stage_id": 7,  
    "estimate": 70,  
    "supplemental_estimate": 0  
  },  
  {  
    "stage_id": 8,  
    "estimate": 90,  
    "supplemental_estimate": 0  
  },  
  {  
    "stage_id": 9,  
    "estimate": 0,  
    "supplemental_estimate": 0  
  }  
]  
}  
]  
}
```

## Stage Change Notifications from Autospot to OBS

A manual or hardware beacon stage change event sends “obs\_event\_id” and the current stage to the OBS Scheduler Update endpoint whenever a stage change is made to an OBS repair order in Autospot. The “obs\_event\_id” is the same value provided by OBS when the Repair Order was created using the Repair Order (POST) message from OBS to Autospot.

### Data Schema

JSON is the expected format. Autospot sends the following (POST) JSON message to the

OBS Scheduler Update endpoint at [https://onboardscheduler.com/api/ver\\_1/updateevent/](https://onboardscheduler.com/api/ver_1/updateevent/).

### OBS Schema reference

<https://kb.onboardscheduler.com/index.php/knowledge-base/onboard-scheduler-update-api/>

```
[
  {
    "obs_event_id": "fromOBS obs_event_id",
    "status": "Current Status from Autospot possible stage values"
  }
]
```

The “status” property is one of any one of the available Autospot “stage\_ids” from the following table:

stage_id	stage_name
1	SCHEDULED IN
2	VEHICLE ARRIVE
3	PRE-WASH
5	DISASSEMBLY
4	BLUEPRINT

6	STAGED FOR PRODUCTION
8	MECHANICAL
7	BODY
9	REFINISH
10	REASSEMBLY
11	DETAIL
12	FINAL INSPECTION
13	DELIVERY READY
15	DELIVERY UNCONFIRMED
14	DELIVERED/CLOSED

### **Error Handling**

If an error is returned in the endpoint response, none of the data in the request will be saved. For example, if 10 repair orders are included in a request and one has an invalid Repair Order number, none of the 10 repair orders will be saved in AutoSpot. After receiving an error, you will need to fix the specified issue or remove the Repair Orders with an issue and re-try the request.

#### Supported Error Handling

- Shop ID Does Not Exist (POST + PATCH)
- Repair Order Number is Duplicated (POST)
- Repair Order Number Does Not Exist (PATCH)
- Internal Server Error (POST + PATCH)