

# Flightdocs API Agreement

Prior to using this API to integrate with Flightdocs all 3rd parties will need to review the agreement. Please contact our integration team at [integrations@atp.com](mailto:integrations@atp.com) for this agreement.

## Flightdocs API Authentication

OAuth2 is a protocol that lets external apps request authorization to private customer or aircraft details in a user's Flightdocs account without getting the user's password.

Please coordinate with Flightdocs to setup your integration before getting started. A registered 3rd party will be assigned a unique Client ID and Client Secret. The Client Secret should not be shared. You will provide Flightdocs with a Redirect Uri where the user will be sent after authorization and your application will receive an authorization code.

### URLs

During development the following URL for our sandbox environment should be used in place of the production URL.

Sandbox: <https://api-sandbox.flightdocs.com/>

Production: <https://api.flightdocs.com/>

### Code Flow Authentication

Begin by directing your users to <https://api.flightdocs.com/oauth2/authorize> through a GET request with the following URL encoded parameters:

| Name          | Description   |
|---------------|---|
| response_type | Required. A value of code should be used to request an authentication code used later.  |
| client_id     | Required. The client ID you received from Flightdocs when you initially setup.  |
| redirect_uri  | Required. The URL in your app where users will be sent after authorization.   |
| state         | Optional parameter that is round tripped from the server back to your application. This is could be used to redirect the user to the correct resource on your site or to prevent cross-site-request-forgery by validating the state received from Flightdocs. |

An example request might look like this:

```
GET https://api.flightdocs.com/oauth2/authorize?response_type=code&client_id=YOUR_CLIENT_ID&redirect_uri=REDIRECT_URI &state=YOUR_STATE_INFORMATION
```

The user will be presented a login page to authenticate with Flightdocs.

Flightdocs will handle for validating the user's credentials and presenting relevant error messages.

On successful log in the user will be taken to a consent page to authorize your application to access their Flightdocs account.

Clicking "Grant access to Flightdocs" will redirect the user to the URI that you specified with a code parameter and a state parameter if you included one.

For example, Flightdocs might redirect to:

```
https://www.acme.com/flightdocstoken?code=RETURNED_AUTH_CODE  
&state=YOUR_STATE_INFORMATION
```

The code you receive as a query string parameter is used to get an access token. It is a single use code and is good for no more than 10 minutes.

To get an access token you'll need to make a POST request to <https://api.flightdocs.com/oauth2/token> with the following parameters:

| Name          | Description   |
|---------------|---|
| grant_type    | Required. A value of authorization_code must be used.   |
| code          | Required. The authorization code you receive previously.  |
| client_id     | Optional. The client ID you received from Flightdocs when you initially setup.  |
| client_secret | Optional. The client secret you received from Flightdocs when you initially setup.  |
| redirect_uri  | Required. The redirect URI you initially setup with Flightdocs and the same value that was passed in the initial user redirect. |

The client\_id and client\_secret can be supplied in the body of the request or in the authorization header. The preferred approach is in the authorization header and is done by combining the client\_id and client\_secret into a string like so "client\_id:client\_secret" and then base64 encoded.

A request for an access token using the authorization header may look like this:

```
POST /oauth2/token HTTP/1.1
Host: api.flightdocs.com
Content-Type: application/x-www-form-urlencoded
Authorization: Basic QWxhZGRpbjpvGVuIHNIc2FtZQ==
grant_type=authorization_code&code=RETURNED_AUTH_CODE &redirect_uri=REDIRECT_URI
```

Here is an example request with the `client_id` and `client_secret` in the body of the request:

```
POST /oauth2/token HTTP/1.1
Host: api.flightdocs.com
Content-Type: application/x-www-form-urlencoded
grant_type=authorization_code&client_id=YOUR_CLIENT_ID
&client_secret=YOUR_CLIENT_SECRET&code=RETURNED_AUTH_CODE &redirect_uri=REDIRECT_URI
```

An example of a successful response will be an HTTP response of 200 which will contain a JSON body like this:

```
{
  "access_token": "Hg3LELHaXu4E0B9R28wOdCqSjJQqqhTjK",
  "token_type": "bearer",
  "expires_in": 3599,
  "refresh_token":
    "adc96ab9f48740c8915b4ff2b55f18332fb852740d8a4e87ad46317a452e1d49"
}
```

The `access_token` is used to make an API request to Flightdocs. The `access_token` is valid for 1 hour. You can use a valid token for as many requests as needed but after an hour you will need to get a new valid token by using the provided `refresh_token`. A `refresh_token` is valid for only one use in 1 year. Every time you get a new `access_token` by using a `refresh_token` you receive a new `refresh_token` valid for another year. This means that if you get a new `refresh_token` at least once a year the user's login should be valid forever.

To use a `refresh_token` to get a new `access_token` you should make a POST request to <https://api.flightdocs.com/oauth2/token> with the following URL encoded parameters:

| Name                       | Description  |
|----------------------------|--|
| <code>grant_type</code>    | Required. A value of <code>refresh_token</code> must be used.                      |
| <code>client_id</code>     | Optional. The client ID you received from Flightdocs when you initially setup.     |
| <code>client_secret</code> | Optional. The client secret you received from Flightdocs when you initially setup. |

| Name          | Description   |
|---------------|---|
| refresh_token | Required. The redirect URI you initially setup with Flightdocs and the same value that was passed in the initial user redirect. |

The client\_id and client\_secret can be passed in the body but the preferred way to send these values is in the authorization header.

Here is a sample request that uses the authorization header to get an access token using a refresh token:

```
POST /oauth2/token HTTP/1.1
Host: api.flightdocs.com
Content-Type: application/x-www-form-urlencoded
Authorization: Basic QWxhZGRpbjpvcmVudHNC2FtZQ==
grant_type=refresh_token&refresh_token=REFRESH_TOKEN
```

Here is a sample of a request that put the client\_id and client\_secret in the request body to get an access token using a refresh token:

```
POST /oauth2/token HTTP/1.1
Host: api.flightdocs.com
Content-Type: application/x-www-form-urlencoded
grant_type=refresh_token&client_id=YOUR_CLIENT_ID
&client_secret=YOUR_CLIENT_SECRET&refresh_token=REFRESH_TOKEN
```

# Flightdocs API Services

## Versioning

The Flightdocs API supports versioning of the API to prevent 3rd parties from needing to rewrite their application when the structure or functionality of a request needs to change. By default the Flightdocs API will always conform to v1 of the API. It is recommended that your requests always explicitly include the version of the current API with the request. This is done by including an additional header to each of the requests made to the Flightdocs API.

A request for a versioned API method may look like this:

```
GET https://api.flightdocs.com/path/method HTTP/1.1
Host: api.flightdocs.com
Content-Type: application/x-www-form-urlencoded
Authorization: Bearer <bearer_token_here>
api-version: 1
```

## Timezone

The Flightdocs API supports handling for time zones. The use case is primarily for the due list. To get items that are due local to the user we allow the time zone to be set in the header according to the list of names from the Olson database.

A request that specifies a time zone may look like this:

```
GET https://api.flightdocs.com/path/method HTTP/1.1
Host: api.flightdocs.com
Content-Type: application/x-www-form-urlencoded
Authorization: Bearer <bearer_token_here>
Time-Zone: America/New_York
```

## Response Structure

### 200 Success Response

The Flightdocs API has a standard output that is followed for all requests (a few minor exceptions may occur like in the case of authentication or authorization errors outlined in the next section).

A typical response structure will look like this:

Example Response:

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Strict-Transport-Security: max-age=2592000
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Date: Wed, 05 Nov 2014 19:54:59 GMT
Content-Length: 1437
{
  "HttpStatusCode":200,
  "Data":[{"AircraftId":1808, "RegistrationNumber":N12345}],
  "Errors":[ ],
  "Messages":[ ],
  "IsSuccess":true,
  "AvailableRecordCount":0,
  "ErrorMessages":"No Errors Reported"
}
```

## Return Values

| Name                 | Description  |
|----------------------|--|
| HttpStatusCode       | Application status code  |
| Data                 | An array or single instance primitive or object data being returned to the client  |
| Errors               | List array of errors   |
| Messages             | List array of information messages   |
| IsSuccess            | Is the call successful and passes all authentication, and business validation      |
| AvailableRecordCount | Available number of records in the current list, or total records if paged         |
| ErrorMessages        | Concatenated list of error messages joined by a semi colon or "No Errors Reported" |

Note: Validation messages from the API are returned with 200 responses. Please check the `IsSuccess` flag and display the error messages to the user. An example of this is when the times being reported are not greater or equal to what is currently logged in Flightdocs.

### 401 Unauthorized Response

If the OAuth bearer token being used expires or is in some other way malformed a 401 HTTP response will be returned with a message in the body saying "Authorization has been denied for this request."

Example Response:

```
HTTP/1.1 401 Unauthorized
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Strict-Transport-Security: max-age=2592000
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
WWW-Authenticate: Bearer
Date: Fri, 07 Nov 2014 20:13:36 GMT
Content-Length: 61
{"Message": "Authorization has been denied for this request."}
```

Action on your part: Request a new OAuth token by using the saved refresh token or ask the user to authenticate with Flightdocs again.

## 403 Forbidden Response

If making a request to the API and a 403 forbidden result is returned there are two different reasons for this response.

1. The user does not have permission to call this method on the API. This return a header of “WWW-Authenticate: Bearer error=“insufficient\_rights” and a message body of “insufficient\_rights”.

Example Response

```
HTTP/1.1 403 Forbidden
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Strict-Transport-Security: max-age=2592000
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
WWW-Authenticate: Bearer error="insufficient_rights"
Date: Fri, 07 Nov 2014 20:19:20 GMT
Content-Length: 33
{"Message":"insufficient_rights"}
```

Action on your part: If you believe that this user should have access to call this method on the API contact Flightdocs to have the permission added to the user.

2. The client application does not have permission to call this method on the API. This return a header of “WWW-Authenticate: Bearer error=“insufficient\_scope” and a message body of “insufficient\_scope”.

```
HTTP/1.1 403 Forbidden
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Strict-Transport-Security: max-age=2592000
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
WWW-Authenticate: Bearer error="insufficient_scope"
Date: Fri, 07 Nov 2014 20:19:20 GMT
Content-Length: 33
{"Message":"insufficient_scope"}
```

Action on your part: If you believe that your application should have access to call this method on the API contact Flightdocs to have the proper scope added to your client application.



## GET /Aircraft/GetMyAircraft

Get a list of aircraft available to the current user.

Parameters

**Note.** The current user is pulled from the auth token passed with the request.

Return Values

| Name               | Data Type  | Description  |
|--------------------|------------|--|
| Id                 | Integer    | Internal database identifier for this aircraft registration                            |
| RegistrationNumber | String(12) | The current registration or tail number of the aircraft as registered with Flightdocs. |
| SerialNumber       | String(40) | The current serial number of the aircraft as registered with Flightdocs.               |

Example Request:

```
GET https://api.flightdocs.com/Aircraft/GetMyAircraft HTTP/1.1
Host: api.flightdocs.com
User-Agent: {user_agent}
Authorization: Bearer {bearer_token_here}
```

Example Response:

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Strict-Transport-Security: max-age=2592000
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Date: Wed, 05 Nov 2014 19:54:59 GMT
Content-Length: 333
{
  "HttpStatusCode":200,
  "Data":[
```

```
{
  "Id":1808,
  "RegistrationNumber":"N12345",
  "SerialNumber":DEMO
},
{
  "Id":3471,
  "RegistrationNumber":"N12346",
  "SerialNumber":DEMO2
}
],
"Errors":[
],
"Messages":[
],
"IsSuccess":true,
"AvailableRecordCount":2,
"ErrorMessage":"No Errors Reported"
}
```

## GET /Aircraft/GetEquipment/{registrationNumber}

Get a list of equipment available to the requested tail number. This service is useful in cases where you need to present the user with a list of equipment on Flightdocs.

### Parameters

| Name               | Data Type  | Description  |
|--------------------|------------|--|
| registrationNumber | String(12) | Required. This is the current registration or tail number of the aircraft as registered with Flightdocs. |

### Return Values

| Name                       | Data Type  | Description  |
|----------------------------|------------|--|
| AircraftRegistrationNumber | String(12) | The current registration or tail number of the aircraft as registered with Flightdocs. |
| EquipmentId                | Integer    | Internal database identifier for this aircraft registration                            |
| EquipmentType              | String(50) | Indicator of the type of equipment this is i.e. Airframe, Engine                       |

| Name              | Data Type   | Description  |
|-------------------|-------------|--|
| AircraftPosition  | Integer     | Position of the equipment on the aircraft i.e. Engine 1 would be in position 1 |
| DisplayName       | String(Max) | Combined tail no, equipment type, position and serial number.                  |
| ManufacturerModel | String(100) | Manufacturers and model information available for the equipment.               |
| SerialNumber      | String(40)  | The current serial number of the equipment as registered with Flightdocs.      |

Example Request:

```
GET https://api.flightdocs.com/Aircraft/GetMyAircraft HTTP/1.1
Host: api.flightdocs.com
User-Agent: {user_agent}
Authorization: Bearer {bearer_token_here}
```

Example Response:

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Date: Mon, 08 Apr 2019 17:29:59 GMT
Content-Length: 797
```

```
{
  "HttpStatusCode": 200,
  "Data": [
    {
      "AircraftRegistrationNumber": "NHOOL1",
      "EquipmentId": 266,
      "EquipmentType": "Airframe",
      "AircraftPosition": 0,
      "DisplayName": "NHOOL1 - Airframe SN: 50000000",
      "ManufacturerModel": "EMBRAER EMB 500",
      "SerialNumber": "50000000"
    }
  ],
  {
    "AircraftRegistrationNumber": "NHOOL1",
```

```
"EquipmentId": 267,
"EquipmentType": "Engine",
"AircraftPosition": 1,
"DisplayName": "NHOOL1 - Engine 1 SN: PCE-LC0222",
"ManufacturerModel": "PW617F-E",
"SerialNumber": "PCE-LC0222"
},
{
  "AircraftRegistrationNumber": "NHOOL1",
  "EquipmentId": 268,
  "EquipmentType": "Engine",
  "AircraftPosition": 2,
  "DisplayName": "NHOOL1 - Engine 2 SN: PCE-LC011",
  "ManufacturerModel": "PW617F-E",
  "SerialNumber": "PCE-LC0111"
}
],
"Warnings": [],
"Errors": [],
"Messages": [],
"IsSuccess": true,
"AvailableRecordCount": 0,
"ErrorMessage": "No Errors Reported"
}
```

## GET /Aircraft/GetCurrentTimes/{registrationNumber}

Get the current aircraft times for an aircraft.

### Parameters

| Name               | Data Type  | Description  |
|--------------------|------------|--|
| registrationNumber | String(12) | Required. This is the current registration or tail number of the aircraft as registered with Flightdocs. |

### Return Values

| Name               | Data Type  | Description  |
|--------------------|------------|--|
| AircraftId         | Integer    | Internal database identifier for this registration (integer)                           |
| RegistrationNumber | String(12) | The current registration or tail number of the aircraft as registered with Flightdocs. |

---

| Name                   | Data Type  | Description  |
|------------------------|------------|--|
| Landings               | Integer    | Current number of landings   |
| LastUpdated            | DateTime   | Date/time in UTC of last update  |
| LastUpdatedBy          | Integer    | Id of the last user to update the aircraft   |
| RIN                    | Integer    | If model has been enabled for RIN tracking, current RIN  |
| ReportedDate           | Date       | Date of last reported times (Format YYYY-MM-DD)  |
| TotalTime              | Decimal(2) | Total time on airframe (2 significant digits)  |
| AircraftEquipmentTimes | Object     | Array of installed equipment on the airframe APU, Engines, Props (see AircraftEquipmentTimes object structure below) |

AircraftEquipmentTimes Object Structure:

| Name                    | Data Type  | Description   |
|-------------------------|------------|---|
| LookupId                | Guid       | Unique identifier for the equipment installed on the airframe   |
| AircraftEquipmentTypeId | Integer    | Id of the type of equipment. Standard types are: 1 – Airframe; 2 – Engine; 3 – APU; 4 – Heater; 5 – Air Conditioner; 6 – Propeller. Please note that Flightdocs supports custom equipment so customer can create their own equipment types so expect other integers to be returned. |
| Type                    | String(8)  | Text description of the type of equipment   |
| Make                    | String(40) | Make of equipment   |
| ModelNumber             | String(40) | Model number of equipment   |
| SerialNumber            | String(40) | Serial number of equipment  |
| InstallDate             | Date       | Install date of equipment (Format YYYY-MM-DD)   |
| Hours                   | Decimal    | Number of hours on the equipment (2 significant digits)   |
| Cycles                  | Decimal    | Number of cycles on the equipment (2 significant digits)  |
| C1C                     | Decimal    | If enabled for C1C/C2C, number of C1 cycles on the equipment (2 significant digits)   |
| C2C                     | Decimal    | if enabled for C1C/C2C, number of C2 cycles on the equipment (2 significant digits)   |
| IsTrackedByAirframe     | Boolean    | Specifies if these times generally track with the airframe. If true validation will require that hours go up if airframe total time goes up and vice versa. If false hours can be updated separately.   |

Example Request:

```
GET https://api.flightdocs.com/Aircraft/GetCurrentTimes/{RegistrationNumber} HTTP/1.1
Host: api.flightdocs.com
User-Agent: {user_agent}
Authorization: Bearer {bearer_token_here}
```

Example Response:

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Strict-Transport-Security: max-age=2592000
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Date: Wed, 05 Nov 2014 19:54:59 GMT
Content-Length: 1437
{
  "HttpStatusCode":200,
  "Data":{
    "AircraftId":1808,
    "RegistrationNumber":"N12345",
    "Landings":2502,
    "LastUpdated":"2014-09-03T16:00:48.54Z",
    "LastUpdatedBy":0,
    "RIN":0,
    "ReportedDate":"2014-07-28",
    "TotalTime":6004.57,
    "AircraftEquipmentTimes":[
      {
        "LookupId":"14d4b3c8-1f6d-428f-ac57-975ea4236d71",
        "AircraftEquipmentTypeId":2,
        "Type":"APU",
        "Make":null,
        "ModelNumber":"131-9B",
        "SerialNumber":"no_update",
        "InstallDate":null,
        "Hours":81198.00,
        "Cycles":71124.00,
        "C1C":0.00,
        "C2C":0.00,
        "IsTrackedByAirframe":false
      },
      {
        "LookupId":"280b0366-7df4-4d93-baed-2d9494e57eda",
        "AircraftEquipmentTypeId":3,
        "Type":"Engine 1",
        "Make":null,
        "ModelNumber":"CFM56-7B26/B1",
        "SerialNumber":"875922",
        "InstallDate":"1999-10-04",
```

```

    "Hours":6004.57,
    "Cycles":2502.00,
    "C1C":0.00,
    "C2C":0.00,
    "IsTrackedByAirframe":true
  }
]
},
"Errors":[
],
"Messages":[
],
"IsSuccess":true,
"AvailableRecordCount":0,
"ErrorMessage":"No Errors Reported"
}

```

## POST /Aircraft/ReportNewTimes

Updates the current times with a new set of times for an aircraft.

**Note:** Flightdocs will automatically increment any equipment that “Is Tracked By Airframe” if it is not supplied with the request. If the reported times for the airframe adds 2 hours then any equipment that tracks with the airframe would add 2 hours as well. Any equipment that does NOT track with the airframe times will need to be pass in the request to report new times.

### Parameters

| Name                   | Data Type  | Description  |
|------------------------|------------|--|
| RegistrationNumber     | String(12) | The current registration or tail number of the aircraft as registered with Flightdocs.                               |
| Landings               | Integer    | Current number of landings   |
| RIN                    | Integer    | If model has been enabled for RIN tracking, current RIN  |
| ReportedDate           | Date       | Date of last reported times (Format as YYYY-MM-DD)   |
| TotalTime              | Decimal    | Total time on airframe (2 significant digits)  |
| AircraftEquipmentTimes | Object     | Array of installed equipment on the airframe APU, Engines, Props (see AircraftEquipmentTimes object structure below) |



AircraftEquipmentTimes Object Structure:

| <b>Name</b>             | <b>Data Type</b> | <b>Description</b>  |
|-------------------------|------------------|---|
| LookupId                | Guid             | Unique identifier for the equipment installed on the airframe                       |
| AircraftEquipmentTypeId | Integer          | Id of the type of equipment   |
| Type                    | String(8)        | Text description of the type of equipment   |
| Make                    | String(40)       | Make of equipment   |
| ModelNumber             | String(40)       | Model number of equipment   |
| SerialNumber            | String(40)       | Serial number of equipment  |
| InstallDate             | Date             | Install date of equipment (Format as YYYY-MM-DD)                                    |
| Hours                   | Decimal          | Number of hours on the equipment (2 significant digits)                             |
| Cycles                  | Decimal          | Number of cycles on the equipment (2 significant digits)                            |
| C1C                     | Decimal          | If enabled for C1C/C2C, number of C1 cycles on the equipment (2 significant digits) |
| C2C                     | Decimal          | if enabled for C1C/C2C, number of C2 cycles on the equipment (2 significant digits) |

Example Request:

POST https://api.flightdocs.com/Aircraft/ReportNewTimes HTTP/1.1

```
Host: api.flightdocs.com
Accept: application/json, text/plain, */*
Content-Type: application/json
User-Agent:
Authorization: Bearer <bearer_token_here>
api-version: 1
{
  "RegistrationNumber": "N12345",
  "Landings": 2502,
  "RIN": 0,
  "ReportedDate": "2014-07-28T00:00:00Z",
  "TotalTime": 6004.57,
  "AircraftEquipmentTimes": [
    {
      "LookupId": "14d4b3c8-1f6d-428f-ac57-975ea4236d71",
      "AircraftEquipmentTypeId": 2,
      "Type": "APU",
      "Make": null,
      "ModelNumber": "131-9B",
      "SerialNumber": "no_update",
      "InstallDate": null,
      "Hours": 81198.00,
      "Cycles": 71124.00,
      "C1C": 0.00,
      "C2C": 0.00
    },
    {
      "LookupId": "280b0366-7df4-4d93-baed-2d9494e57eda",
      "AircraftEquipmentTypeId": 3,
      "Type": "Engine 1",
      "Make": null,
      "ModelNumber": "CFM56-7B26/B1",
      "SerialNumber": "875922",
      "InstallDate": "1999-10-04",
      "Hours": 6004.57,
      "Cycles": 2502.00,
      "C1C": 0.00,
      "C2C": 0.00
    }
  ]
}
```

Example Response:

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Strict-Transport-Security: max-age=2592000
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Date: Wed, 05 Nov 2014 19:54:59 GMT
Content-Length: 1437
{
  "HttpStatusCode":200,
  "Data":{
    "AircraftId":1808,
    "RegistrationNumber":"N12345",
    "Flightdocs, Inc. 10 July 2014",
    "Landings":2502,
    "LastUpdated":"2014-09-03T16:00:48.54Z",
    "LastUpdatedBy":0,
    "RIN":0,
    "ReportedDate":"2014-07-28",
    "TotalTime":6004.57,
    "AircraftEquipmentTimes":[
      {
        "LookupId":"14d4b3c8-1f6d-428f-ac57-975ea4236d71",
        "AircraftEquipmentTypeId":2,
        "Type":"APU",
        "Make":null,
        "ModelNumber":"131-9B",
        "SerialNumber":"no_update",
        "InstallDate":null,
        "Hours":81198.00,
        "Cycles":71124.00,
        "C1C":0.00,
        "C2C":0.00,
        "IsTrackedByAirframe":false
      },
      {
        "LookupId":"280b0366-7df4-4d93-baed-2d9494e57eda",
        "AircraftEquipmentTypeId":3,
        "Type":"Engine 1",
        "Make":null,
        "ModelNumber":"CFM56-7B26/B1",
        "SerialNumber":"875922",
```

```
"InstallDate":"1999-10-04",
"Hours":6004.57,
"Cycles":2502.00,
"C1C":0.00,
"C2C":0.00,
"IsTrackedByAirframe":true
}
]
},
"Errors":[
],
"Messages":[
],
"IsSuccess":true,
"AvailableRecordCount":0,
"ErrorMessages":"No Errors Reported"
}
```

## GET /Aircraft/GetAvailabilityStatus/{registrationNumber}

Get the availability status for a given aircraft by the registration number.

### Parameters

| Name               | Data Type  | Description  |
|--------------------|------------|--|
| registrationNumber | String(12) | Required. This is the current registration or tail number of the aircraft as registered with Flightdocs. |

### Return Values

| Name               | Data Type  | Description  |
|--------------------|------------|--|
| RegistrationNumber | String(12) | The current registration or tail number of the aircraft as registered with Flightdocs.   |
| Status             | String(30) | Internal database identifier for this aircraft registration <ul style="list-style-type: none"><li>• Available</li><li>• Unavailable</li><li>• Out for service</li><li>• Grounded</li></ul> |

Example Request:

```
GET https://api.flightdocs.com/Aircraft/GetAvailabilityStatus/NHOOLI1 HTTP/1.1
Host: api.flightdocs.com
User-Agent: {user_agent}
Authorization: Bearer {bearer_token_here}
```

Example Response:

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Date: Wed, 10 Apr 2019 14:59:55 GMT
Content-Length: 204
```

```
{
  "HttpStatusCode": 200,
  "Data": {
    "RegistrationNumber": "NHOOLI1",
    "Status": "Unavailable"
  },
  "Warnings": [],
  "Errors": [],
  "Messages": [],
  "IsSuccess": true,
  "AvailableRecordCount": 0,
  "ErrorMessages": "No Errors Reported"
}
```

## PUT /Aircraft/UpdateAvailabilityStatus/

Set the availability status for a given aircraft by the registration number.

### Parameters

| Name                 | Data Type  | Description   |
|----------------------|------------|---|
| RegistrationNumber   | String(12) | Required. This is the current registration or tail number of the aircraft as registered with Flightdocs.  |
| AvailabilityStatusId | Integer    | Required. The new availability status to set this aircraft to. Valid values are: <ul style="list-style-type: none"><li>• 1 – Available</li><li>• 2 – Unavailable</li><li>• 3 – Out for service</li><li>• 4 – Grounded</li></ul> |

### Return Values

200 HTTP Response with property "IsSuccess" true.

### Example Request:

```
PUT https://api.flightdocs.com/Aircraft/UpdateAvailabilityStatus HTTP/1.1
Host: api.flightdocs.com
Accept: application/json, text/plain, */*
User-Agent: {user_agent}
Content-Type: application/json
Authorization: Bearer {bearer_token_here}
```

```
{
  "RegistrationNumber": "NHOOL1",
  "AvailabilityStatusId": 2
}
```

### Example Response:

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Date: Wed, 10 Apr 2019 17:21:41 GMT
```

Content-Length: 120

```
{
  "Warnings": [],
  "Errors": [],
  "Messages": [],
  "IsSuccess": true,
  "AvailableRecordCount": 0,
  "ErrorMessages": "No Errors Reported"
}
```

## GET /MaintenanceItem/GetDueList /{registrationNumber}

**Version 1** – Get a minimal version of the current due list for an aircraft. This is appropriate for a listing of all items coming due.

Note - See version 2 for a more detailed response useful for display on a calendar or off-line calculations.

### Parameters

| Name               | Data Type  | Description  |
|--------------------|------------|--|
| RegistrationNumber | String(12) | Required. This is the current registration or tail number of the aircraft as registered with Flightdocs. |

### Return Values (Array)

| Name                        | Data Type   | Description  |
|-----------------------------|-------------|--|
| AircraftId                  | Int         | Internal database identifier for this registration (integer)   |
| ATACode                     | Int         | The ATA code for the item. 101 indicates a MEL item.   |
| ATADisplay                  | String(100) | Formatted field combining ATACode, Mfg Maint Code, Amendment, and Version for display on reports.  |
| CyclesOnPart                | Decimal     | Number of cycles the part has at installation  |
| DueStatus                   | Int         | Status Code to categorize items. (1 - No Categorization, 2 - Greater than 10 days till due, 3 - Within 10 days but not overdue, 4 - Past due but within tolerance, 5 - Past due) |
| Disposition                 | String(50)  | Disposition of the maintenance task.   |
| HasTaskcard                 | Bool        | Indicator if the maintenance item has any task cards.  |
| Id                          | Guid        | Unique identifier for a maintenance item.  |
| GroupName                   | String(200) | Item Grouping Name   |
| IntervalFormatted           | String(max) | A textual formatted version of all intervals   |
| IntervalAdjustmentFormatted | String      | A textual formatted version of all interval adjustments  |
| ItemDescription             | String(255) | The textual description of the item (name).  |

| Name                         | Data Type   | Description   |
|------------------------------|-------------|---|
| ItemNumber                   | Int         | The item number identifier  |
| ItemType                     | Int         | The type of item this is, which could be part, inspection, AD, etc.   |
| LandingsOnPart               | Int         | Number of landings the part has at installation   |
| LastCWFormatted              | String(max) | The last complied with formatted  |
| HoursOnPart                  | Decimal     | The number of hours the part has at installation  |
| ManufacturingMaintenanceCode | String(40)  | The manufacturer maintenance code   |
| NextDueDate                  | Date        | The next due date (format YYYY-MM-DD)   |
| NextDueHours                 | Decimal     | The next due hours  |
| NextDueLandings              | Int         | The next due landings   |
| NextDueCycles                | Decimal     | The next due cycles   |
| NextDueRIN                   | Int         | The next due RIN  |
| NextDueFormatted             | String(max) | A formatted text-version of the next due date   |
| PartNumberOn                 | String(40)  | The part number currently on the aircraft, if applicable for the item   |
| PartNumberOff                | String(40)  | The part number that came off the aircraft on last maintenance, if applicable for the item  |
| Position                     | String(30)  | Information about the position for the item. Open text field for customers to defined, not currently used in reporting.   |
| Reference                    | String(50)  | Reference field typically used for storing the AMM reference number.  |
| RegularNotes                 | String(max) | Text notes  |
| RemainingFormatted           | String(max) | The formatted collection of metrics remaining before the next maintenance is due. For date-based items it is a hard date. For all others it is based on the usage preferences for the airframe which can be set in the application. |
| RINOnPart                    | Int         | The number of RIN the part has at installation  |
| SerialNumberOn               | String(40)  | The serial number of the part currently on the aircraft, if applicable for the item   |
| SerialNumberOff              | String(40)  | The serial number of the part that came off the aircraft on last maintenance, if applicable for the item  |
| Tolerance                    | String(50)  | The tolerance or grace period of a maintenance item   |
| TrackedBy                    | String(8)   | The major component this item is tracked by   |
| TrackedByFormatted           | String(max) | The tracked by formatted for display.   |
| Aircraft_RegistrationNumber  | String(12)  | The current registration or tail number of the aircraft as registered with Flightdocs.  |



Example Request (API Version 1):

```
GET https://api.flightdocs.com/MaintenanceItem/GetDueList/N12345 HTTP/1.1
Host: api.flightdocs.com
Accept: application/json, text/plain, */*
User-Agent: {user_agent}
Authorization: Bearer {bearer_token_here}
api-version: 1
Time-Zone: America/New_York
```

Example Response (API Version 1):

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Strict-Transport-Security: max-age=2592000
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Date: Wed, 05 Nov 2014 19:54:59 GMT
Content-Length: 1437
{
  "HttpStatusCode":200,
  "Data":[
    {
      "AircraftId":1808,
      "ATACode":5,
      "ATADisplay":"5 0110",
      "CyclesOnPart":0.00,
      "Disposition":"",
      "DueStatus":1,
      "HasTaskcard":false,
      "GroupName":"1 MO",
      "HoursOnPart":0.00,
      "Id":"11842571-8ce3-40b6-9967-a3c8abc569a3",
      "IntervalFormatted":"M: 1\r\n",
      "IntervalAdjustmentFormatted":"",
      "ItemDescription":"1 MONTH CHECK",
      "ItemNumber":2534,
      "ItemType":1,
      "LandingsOnPart":0,
      "LastCWFormatted":"23/07/2014\r\nH: 81198\r\nL: 71141\r\n",
      "ManufacturingMaintenanceCode":"0110",
      "NextDueDate":"2014-08-31",
      "NextDueHours":null,
```

```
"NextDueLandings":null,
"NextDueCycles":null,
"NextDueRIN":null,
"NextDueFormatted":"31/08/2014\r\n",
"PartNumberOn":null,
"PartNumberOff":null,
"Position":"One",
"Reference":"","
"RegularNotes":null,
"RemainingFormatted":"M: -2\r\n",
"RINOnPart":0,
"SerialNumberOn":null,
"SerialNumberOff":null,
"Tolerance":"","
"TrackedBy":"Airframe",
"TrackedByFormatted":"Airframe",
"Aircraft_RegistrationNumber":"N12345"
},
{
"AircraftId":1808,
"ATACode":5,
"ATADisplay":"5 0135",
"CyclesOnPart":0.00,
"Disposition":"","
"HasTaskcard":false,
"HoursOnPart":0.00,
"GroupName":"1m",
"Id":"a683962b-fb74-4f4e-8edd-839d94d3d327",
"IntervalFormatted":"D: 30\r\n",
"IntervalAdjustmentFormatted":"","
"ItemDescription":"FAR 135 AIRWORTHINESS RELEASE",
"ItemNumber":5439,
"ItemType":1,
"LandingsOnPart":0,
"LastCWFormatted":"23/07/2014\r\nH: 82000\r\nL: 71141\r\n",
"ManufacturingMaintenanceCode":"0135",
"NextDueDate":"2014-08-22",
"NextDueHours":null,
"NextDueLandings":null,
"NextDueCycles":null,
"NextDueRIN":null,
"NextDueFormatted":"22/08/2014\r\n",
"PartNumberOn":null,
"PartNumberOff":null,
"Position":"Two",
"Reference":"","
```

```

"RegularNotes":null,
"RemainingFormatted":"D: -15\r\n",
"RINOnPart":0,
"SerialNumberOn":null,
"SerialNumberOff":null,
"Tolerance":"D: 20",
"TrackedBy":"Airframe",
"TrackedByFormatted":"Airframe",
"Aircraft_RegistrationNumber":null
}
],
"Errors":[
],
"Messages":[
],
"IsSuccess":true,
"AvailableRecordCount":2,
"ErrorMessages":"No Errors Reported"
}

```

## GET /MaintenanceItem/GetDueList /{registrationNumber}

**Version 2** – Get a complete version of the current due list for an aircraft. This is appropriate for a calendar view of the when items come due, off-line calculations of due date, or simply a more detailed listing of all items coming due.

To use this API add a header value of api-version and set this to a value of “2”. This is shown in an example request below.

### Parameters

| Name               | Data Type  | Description  |
|--------------------|------------|--|
| RegistrationNumber | String(12) | Required. This is the current registration or tail number of the aircraft as registered with Flightdocs. |

### Return Values (Array)

| Name         | Data Type   | Description   |
|--------------|-------------|---|
| AircraftId   | Integer     | Internal database identifier for this registration (integer)                                      |
| ATACode      | Integer     | The ATA code for the item. 101 indicates a MEL item.  |
| ATADisplay   | String(100) | Formatted field combining ATACode, Mfg Maint Code, Amendment, and Version for display on reports. |
| CyclesOnPart | Decimal     | Number of cycles the part has at installation   |

| Name                         | Data Type   | Description   |
|------------------------------|-------------|---|
| DueStatus                    | Integer     | Status Code to categorize items. (1 - No Categorization, 2 - Greater than 10 days till due, 3 - Within 10 days but not overdue, 4 - Past due but within tolerance, 5 - Past due, 6 – Aircraft On Ground)                            |
| Disposition                  | String(50)  | Disposition of the maintenance task.  |
| HasTaskcard                  | Boolean     | Indicator if the maintenance item has any task cards.   |
| Id                           | Guid        | Unique identifier for a maintenance item.   |
| GroupName                    | String(200) | Item Grouping Name  |
| IntervalFormatted            | String(max) | A textual formatted version of all intervals  |
| IntervalAdjustmentFormatted  | String      | A textual formatted version of all interval adjustments   |
| ItemDescription              | String(255) | The textual description of the item (name).   |
| ItemNumber                   | Integer     | The item number identifier  |
| ItemType                     | Integer     | The type of item this is, which could be part, inspection, AD, etc.   |
| LandingsOnPart               | Integer     | Number of landings the part has at installation   |
| LastCWFormatted              | String(max) | The last complied with formatted  |
| HoursOnPart                  | Decimal     | The number of hours the part has at installation  |
| ManufacturingMaintenanceCode | String(40)  | The manufacturer maintenance code   |
| NextDueDate                  | Date        | The next due date (Format YYYY-MM-DD)   |
| NextDueHours                 | Decimal     | The next due hours  |
| NextDueLandings              | Integer     | The next due landings   |
| NextDueCycles                | Decimal     | The next due cycles   |
| NextDueRIN                   | Integer     | The next due RIN  |
| NextDueFormatted             | String(max) | A formatted text-version of the next due date   |
| PartNumberOn                 | String(40)  | The part number currently on the aircraft, if applicable for the item   |
| PartNumberOff                | String(40)  | The part number that came off the aircraft on last maintenance, if applicable for the item (Version 1 only)   |
| Position                     | String(30)  | Information about the position for the item. Open text field for customers to defined, not currently used in reporting.   |
| Reference                    | String(50)  | Reference field typically used for storing the AMM reference number.  |
| RegularNotes                 | String(max) | Text notes  |
| RemainingFormatted           | String(max) | The formatted collection of metrics remaining before the next maintenance is due. For date-based items it is a hard date. For all others it is based on the usage preferences for the airframe which can be set in the application. |
| RINOnPart                    | Integer     | The number of RIN the part has at installation  |
| SerialNumberOn               | String(40)  | The serial number of the part currently on the aircraft, if applicable for the item   |

| Name                        | Data Type   | Description  |
|-----------------------------|-------------|--|
| SerialNumberOff             | String(40)  | The serial number of the part that came off the aircraft on last maintenance, if applicable for the item |
| Tolerance                   | String(50)  | The tolerance or grace period of a maintenance item  |
| TrackedBy                   | String(8)   | The major component this item is tracked by  |
| TrackedByFormatted          | String(max) | The tracked by formatted for display   |
| Aircraft_RegistrationNumber | String(12)  | The current registration or tail number of the aircraft as registered with Flightdocs.                   |
| ToleranceFormatted          | String(max) | Tolerance formatted for display  |
| CalculatedDate              | Date        | Calculated due date accounting for anticipated utilization   |
| CalculatedMaxDueDate        | Date        | Calculated due date accounting for anticipated utilization and tolerance                                 |
| IntervalDays                | Integer     | Required maintenance interval in days  |
| IntervalMonths              | Integer     | Required maintenance interval in months  |
| IntervalHours               | Decimal     | Required maintenance interval in hours   |
| IntervalLandings            | Integer     | Required maintenance interval in landings  |
| IntervalCycles              | Decimal     | Required maintenance interval in cycles  |
| IntervalRIN                 | Decimal     | Required maintenance interval in RIN   |
| IntervalAdjustmentHours     | Decimal     | Adjust the interval hours  |
| IntervalAdjustmentDays      | Integer     | Adjust the interval days   |
| IntervalAdjustmentMonths    | Integer     | Adjust the interval months   |
| IntervalAdjustmentLandings  | Integer     | Adjust the interval landings   |
| IntervalAdjustmentCycles    | Decimal     | Adjust the interval cycles   |
| IntervalAdjustmentRIN       | Decimal     | Adjust the interval RIN  |
| LogDate                     | Date        | Current log date   |
| LogHours                    | Decimal     | Total logged hours   |
| LogLandings                 | Integer     | Total logged landings  |
| LogCycles                   | Decimal     | Total logged cycles  |
| LogRIN                      | Decimal     | Total logged RIN   |
| RemainingHours              | Decimal     | Aircraft hours remaining before due  |
| RemainingLandings           | Integer     | Aircraft landings remaining before due   |
| RemainingRIN                | Decimal     | Aircraft RIN remaining before due  |
| RemainingCycles             | Decimal     | Aircraft cycles remaining before due   |
| RemainingMonths             | Integer     | Months remaining before due  |
| RemainingDays               | Integer     | Days remaining before due  |
| ToleranceMonths             | Integer     | Acceptable variance in months from due date  |
| ToleranceDays               | Integer     | Acceptable variance in days from due date  |
| ToleranceHours              | Decimal     | Acceptable variance in hours   |
| ToleranceLandings           | Integer     | Acceptable variance in landings  |
| ToleranceCycles             | Decimal     | Acceptable variance in cycles  |
| ToleranceRIN                | Decimal     | Acceptable variance in RIN   |
| ToleranceRemainingMonths    | Integer     | Months remaining in tolerance  |

| Name                       | Data Type | Description  |
|----------------------------|-----------|--|
| ToleranceRemainingDays     | Integer   | Days remaining in tolerance                          |
| ToleranceRemainingHours    | Decimal   | Hours remaining in tolerance                         |
| ToleranceRemainingLandings | Integer   | Landings remaining in tolerance                      |
| ToleranceRemainingCycles   | Decimal   | Cycles remaining in tolerance                        |
| ToleranceRemainingRIN      | Decimal   | RIN remaining in tolerance                           |
| ToleranceNextDueDate       | Date      | Date due with tolerance included                     |
| ToleranceNextDueHours      | Decimal   | Hours remaining until due with tolerance included    |
| ToleranceNextDueLandings   | Integer   | Landings remaining until due with tolerance included |
| ToleranceNextDueCycles     | Decimal   | Cycles remaining until due with tolerance included   |
| ToleranceNextDueRIN        | Decimal   | RIN remaining until due with tolerance included      |
| AircraftUtilizationHours   | Decimal   | Anticipated utilization of aircraft in hours         |
| AircraftUtilizationCycles  | Decimal   | Anticipated utilization of aircraft in cycles        |
| AircraftUtilizationRIN     | Decimal   | Anticipated utilization of aircraft in RIN           |

Example Request (API Version 2):

```
GET https://api.flightdocs.com/MaintenanceItem/GetDueList/N900EX HTTP/1.1
Host:"api.flightdocs.com"
Accept:"*/*"
User-Agent:{User Agent}
Authorization: {bearer_token_here}
api-version:"2"
Time-Zone: America/New_York
```

Example Response (API Version 2):

```
"HttpStatusCode": 200,
  "Data": [
    {
      "AircraftId": 4603,
      "Aircraft_RegistrationNumber": "N900EX",
      "ATACode": 54,
      "ATADisplay": "54 00-00-220-802",
      "TrackedBy": "Airframe ",
      "DueStatus": 5,
      "Disposition": "",
      "Id": "53deaf80-ae74-4d00-99f8-53d97b478cca",
      "GroupName": "",
      "IsParentItem": false,
      "IsChildItem": false,
      "ItemDescription": "DETL INSP/CHK #2 ENG NAC&PYL",
      "ItemNumber": 7811,
```

```
"ItemType": "PART",
"ManufacturingMaintenanceCode": "00-00-220-802",
"PartNumberOn": "6969",
"Position": "",
"Reference": "",
"RegularNotes": "",
"SerialNumberOn": "553",
"IntervalFormatted": "L: 3750\r\n",
"IntervalAdjustmentFormatted": "",
"LastCWFormatted": "20-NOV-2012\r\nH: 8112.26 \r\nL: 7117 \r\n",
"NextDueFormatted": "L: 10867\r\n",
"RemainingFormatted": "L: -5692\r\n",
"ToleranceFormatted": "",
"CalculatedDueDate": "2011-05-12",
"CalculatedMaxDueDate": "2011-05-12",
"CyclesOnPart": null,
"LandingsOnPart": null,
"HoursOnPart": null,
"RINOnPart": null,
"IntervalDays": 0,
"IntervalMonths": 0,
"IntervalHours": null,
"IntervalLandings": 3750,
"IntervalCycles": 0,
"IntervalRIN": 0,
"IntervalAdjustmentHours": 0,
"IntervalAdjustmentDays": 0,
"IntervalAdjustmentMonths": 0,
"IntervalAdjustmentLandings": 0,
"IntervalAdjustmentCycles": 0,
"IntervalAdjustmentRIN": 0,
"LogDate": "2012-11-20",
"LogHours": 8112.26,
"LogLandings": 7117,
"LogCycles": 0,
"LogRIN": null,
"NextDueDate": null,
"NextDueHours": null,
"NextDueLandings": 10867,
"NextDueCycles": null,
"NextDueRIN": null,
"RemainingHours": null,
"RemainingLandings": -5692,
```

```
"RemainingRIN": null,  
"RemainingCycles": null,  
"RemainingMonths": null,  
"RemainingDays": null,  
"ToleranceMonths": null,  
"ToleranceDays": null,  
"ToleranceHours": null,  
"ToleranceLandings": null,  
"ToleranceCycles": null,  
"ToleranceRIN": null,  
"ToleranceRemainingMonths": null,  
"ToleranceRemainingDays": null,  
"ToleranceRemainingHours": null,  
"ToleranceRemainingLandings": null,  
"ToleranceRemainingCycles": null,  
"ToleranceRemainingRIN": null,  
"ToleranceNextDueDate": null,  
"ToleranceNextDueHours": null,  
"ToleranceNextDueLandings": null,  
"ToleranceNextDueCycles": null,  
"ToleranceNextDueRIN": null,  
"AircraftUtilizationHours": 1,  
"AircraftUtilizationLandings": 2,  
"AircraftUtilizationCycles": -1,  
"AircraftUtilizationRIN": -1  
  }  
"Warnings": [],  
"Errors": [],  
"Messages": [],  
"IsSuccess": true,  
"AvailableRecordCount": 0,  
"ErrorMessage": "No Errors Reported"  
}
```



## GET /MaintenanceItem/GetDueList

**Version 1** – Get a minimal version of the current due list for an aircraft. This is appropriate for a listing of all items coming due. The criteria are used for projecting how far out items are retrieved based on daily aircraft utilization preferences.

Note - See version 2 for a more detailed response useful for display on a calendar or off-line calculations.

### Parameters

| Name               | Data Type  | Description   |
|--------------------|------------|---|
| RegistrationNumber | String(12) | Required/Optional. This is the current registration or tail number of the aircraft as registered with Flightdocs. This or the AircraftId is required on each request. |
| AircraftId         | Integer    | Required/Optional. This is the Flightdocs Id used to identify a customer's aircraft. This or the RegistrationNumber is required on each request.                      |
| ProjectedDays      | Integer    | Optional. Sets the number of days out to get a due list for based on utilization for non-date-based intervals (cycles, landings, hours). The default is 90 days.      |
| ProjectedDueDate   | Date       | Optional. Overrides the due date that would be projected above. (Formatted as YYYY-MM-DD)   |
| ProjectedHours     | Decimal    | Optional. Overrides calculated hours projection based on daily utilization preferences for hours-based intervals.   |
| ProjectedLandings  | Decimal    | Optional. Overrides calculated landings projection based on daily utilization preferences for landings-based intervals.   |
| ProjectedCycles    | Decimal    | Optional. Overrides calculated cycles projection based on daily utilization preferences for cycles-based intervals.   |
| ProjectedRIN       | Decimal    | Optional. Overrides calculated RIN projection based on daily utilization preferences for RIN based intervals.   |
| ProjectedApuHours  | Decimal    | Optional. Overrides calculated APU hours projection based on daily utilization preferences for APU hours-based intervals.   |
| ProjectedApuCycles | Decimal    | Optional. Overrides calculated APU cycles projection based on daily utilization preferences for APU cycles-based intervals.   |
| IncludePaging      | Boolean    | Optional. Indicator if the results should be paged.   |
| PageSize           | Integer    | Optional. Number for records to return in each page of the results.   |
| PageIndex          | Integer    | Optional. Indicator of which page of the results to return.   |

## Return Values (Array)

| Name                         | Data Type   | Description  |
|------------------------------|-------------|--|
| AircraftId                   | Int         | Internal database identifier for this registration (integer)   |
| ATACode                      | Int         | The ATA code for the item. 101 indicates a MEL item.   |
| ATADisplay                   | String(100) | Formatted field combining ATACode, Mfg Maint Code, Amendment, and Version for display on reports.  |
| CyclesOnPart                 | Decimal     | Number of cycles the part has at installation  |
| DueStatus                    | Int         | Status Code to categorize items. (1 - No Categorization, 2 - Greater than 10 days till due, 3 - Within 10 days but not overdue, 4 - Past due but within tolerance, 5 - Past due) |
| Disposition                  | String(50)  | Disposition of the maintenance task.   |
| HasTaskcard                  | Bool        | Indicator if the maintenance item has any task cards.  |
| Id                           | Guid        | Unique identifier for a maintenance item.  |
| GroupName                    | String(200) | Item Grouping Name   |
| IntervalFormatted            | String(max) | A textual formatted version of all intervals   |
| IntervalAdjustmentFormatted  | String      | A textual formatted version of all interval adjustments  |
| ItemDescription              | String(255) | The textual description of the item (name).  |
| ItemNumber                   | Int         | The item number identifier   |
| ItemType                     | Int         | The type of item this is, which could be part, inspection, AD, etc.  |
| LandingsOnPart               | Int         | Number of landings the part has at installation  |
| LastCWFormatted              | String(max) | The last complied with formatted   |
| HoursOnPart                  | Decimal     | The number of hours the part has at installation   |
| ManufacturingMaintenanceCode | String(40)  | The manufacturer maintenance code  |
| NextDueDate                  | Date        | The next due date (Format YYYY-MM-DD)  |
| NextDueHours                 | Decimal     | The next due hours   |
| NextDueLandings              | Int         | The next due landings  |
| NextDueCycles                | Decimal     | The next due cycles  |
| NextDueRIN                   | Int         | The next due RIN   |
| NextDueFormatted             | String(max) | A formatted text-version of the next due date  |
| PartNumberOn                 | String(40)  | The part number currently on the aircraft, if applicable for the item  |
| PartNumberOff                | String(40)  | The part number that came off the aircraft on last maintenance, if applicable for the item   |
| Position                     | String(30)  | Information about the position for the item. Open text field for customers to defined, not currently used in reporting.  |
| Reference                    | String(50)  | Reference field typically used for storing the AMM reference number.   |
| RegularNotes                 | String(max) | Text notes   |

| Name                        | Data Type   | Description   |
|-----------------------------|-------------|---|
| RemainingFormatted          | String(max) | The formatted collection of metrics remaining before the next maintenance is due. For date-based items it is a hard date. For all others it is based on the usage preferences for the airframe which can be set in the application. |
| RINOnPart                   | Int         | The number of RIN the part has at installation  |
| SerialNumberOn              | String(40)  | The serial number of the part currently on the aircraft, if applicable for the item   |
| SerialNumberOff             | String(40)  | The serial number of the part that came off the aircraft on last maintenance, if applicable for the item  |
| Tolerance                   | String(50)  | The tolerance or grace period of a maintenance item   |
| TrackedBy                   | String(8)   | The major component this item is tracked by   |
| TrackedByFormatted          | String(max) | The tracked by formatted for display.   |
| Aircraft_RegistrationNumber | String(12)  | The current registration or tail number of the aircraft as registered with Flightdocs.  |

Example Request (API Version 1):

GET

```
https://api.flightdocs.com/MaintenanceItem/GetDueList?RegistrationNumber=N12345&ProjectedDays=180 HTTP/1.1
Host: api.flightdocs.com
Accept: application/json, text/plain, */*
User-Agent: {user_agent}
Authorization: Bearer {bearer_token_here}
api-version: 1
Time-Zone: America/New_York
```

Example Response (API Version 1):

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Strict-Transport-Security: max-age=2592000
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Date: Wed, 05 Nov 2014 19:54:59 GMT
Content-Length: 1437
{
  "HttpStatusCode":200,
  "Data":[
    {
```

```
"AircraftId":1808,
"ATACode":5,
"ATADisplay":"5 0110",
"CyclesOnPart":0.00,
"Disposition": "",
"DueStatus":1,
"HasTaskcard":false,
"GroupName":"1 MO",
"HoursOnPart":0.00,
"Id":"11842571-8ce3-40b6-9967-a3c8abc569a3",
"IntervalFormatted":"M: 1\r\n",
"IntervalAdjustmentFormatted": "",
"ItemDescription":"1 MONTH CHECK",
"ItemNumber":2534,
"ItemType":1,
"LandingsOnPart":0,
>LastCWFormatted":"23/07/2014\r\nH: 81198\r\nL: 71141\r\n",
"ManufacturingMaintenanceCode":"0110",
"NextDueDate":"2014-08-31",
"NextDueHours":null,
"NextDueLandings":null,
"NextDueCycles":null,
"NextDueRIN":null,
"NextDueFormatted":"31/08/2014\r\n",
"PartNumberOn":null,
"PartNumberOff":null,
"Position":"One",
"Reference": "",
"RegularNotes":null,
"RemainingFormatted":"M: -2\r\n",
"RINOnPart":0,
"SerialNumberOn":null,
"SerialNumberOff":null,
"Tolerance": "",
"TrackedBy":"Airframe",
"TrackedByFormatted":"Airframe",
"Aircraft_RegistrationNumber":"N12345"
},
{
"AircraftId":1808,
"ATACode":5,
"ATADisplay":"5 0135",
"CyclesOnPart":0.00,
"Disposition": "",
"HasTaskcard":false,
"HoursOnPart":0.00,
```

```
"GroupName":"1m",
"Id":"a683962b-fb74-4f4e-8edd-839d94d3d327",
"IntervalFormatted":"D: 30\r\n",
"IntervalAdjustmentFormatted":"",
"ItemDescription":"FAR 135 AIRWORTHINESS RELEASE",
"ItemNumber":5439,
"ItemType":1,
"LandingsOnPart":0,
"LastCWFormatted":"23/07/2014\r\nH: 82000\r\nL: 71141\r\n",
"ManufacturingMaintenanceCode":"0135",
"NextDueDate":"2014-08-22",
"NextDueHours":null,
"NextDueLandings":null,
"NextDueCycles":null,
"NextDueRIN":null,
"NextDueFormatted":"22/08/2014\r\n",
"PartNumberOn":null,
"PartNumberOff":null,
"Position":"Two",
"Reference":"",
"RegularNotes":null,
"RemainingFormatted":"D: -15\r\n",
"RINOnPart":0,
"SerialNumberOn":null,
"SerialNumberOff":null,
"Tolerance":"D: 20",
"TrackedBy":"Airframe",
"TrackedByFormatted":"Airframe",
"Aircraft_RegistrationNumber":null
}
],
"Errors":[
],
"Messages":[
],
"IsSuccess":true,
"AvailableRecordCount":2,
"ErrorMessage":"No Errors Reported"
}
```

## GET /MaintenanceItem/GetDueList

**Version 2** – Get a complete version of the current due list for an aircraft. This is appropriate for a calendar view of the when items come due, off-line calculations of due date, or simply a more detailed listing of all items coming due. The criteria are used for projecting how far out items are retrieved based on daily aircraft utilization preferences.

To use this API add a header value of api-version and set this to a value of “2”. This is shown in an example request below.

### Parameters

| Name               | Data Type  | Description   |
|--------------------|------------|---|
| RegistrationNumber | String(12) | Required/Optional. This is the current registration or tail number of the aircraft as registered with Flightdocs. This or the AircraftId is required on each request. |
| AircraftId         | Integer    | Required/Optional. This is the Flightdocs Id used to identify a customer’s aircraft. This or the RegistrationNumber is required on each request.                      |
| ProjectedDays      | Integer    | Optional. Sets the number of days out to get a due list for based on utilization for non-date-based intervals (cycles, landings, hours). The default is 90 days.      |
| ProjectedDueDate   | Date       | Optional. Overrides the due date that would be projected above. (Formatted as YYYY-MM-DD)   |
| ProjectedHours     | Decimal    | Optional. Overrides calculated hours projection based on daily utilization preferences for hours-based intervals.   |
| ProjectedLandings  | Decimal    | Optional. Overrides calculated landings projection based on daily utilization preferences for landings-based intervals.   |
| ProjectedCycles    | Decimal    | Optional. Overrides calculated cycles projection based on daily utilization preferences for cycles-based intervals.   |
| ProjectedRIN       | Decimal    | Optional. Overrides calculated RIN projection based on daily utilization preferences for RIN based intervals.   |
| ProjectedApuHours  | Decimal    | Optional. Overrides calculated APU hours projection based on daily utilization preferences for APU hours-based intervals.   |
| ProjectedApuCycles | Decimal    | Optional. Overrides calculated APU cycles projection based on daily utilization preferences for APU cycles-based intervals.   |
| IncludePaging      | Boolean    | Optional. Indicator if the results should be paged.   |
| PageSize           | Integer    | Optional. Number for records to return in each page of the results.   |
| PageIndex          | Integer    | Optional. Indicator of which page of the results to return.   |

## Return Values (Array)

| Name                         | Data Type   | Description  |
|------------------------------|-------------|--|
| AircraftId                   | Integer     | Internal database identifier for this registration (integer)   |
| ATACode                      | Integer     | The ATA code for the item.   |
| ATADisplay                   | String(100) | Formatted field combining ATACode, Mfg Maint Code, Amendment, and Version for display on reports.  |
| CyclesOnPart                 | Decimal     | Number of cycles the part has at installation  |
| DueStatus                    | Integer     | Status Code to categorize items. (1 - No Categorization, 2 - Greater than 10 days till due, 3 - Within 10 days but not overdue, 4 - Past due but within tolerance, 5 - Past due, 6 - Aircraft On Ground) |
| Disposition                  | String(50)  | Disposition of the maintenance task.   |
| HasTaskcard                  | Boolean     | Indicator if the maintenance item has any task cards.  |
| Id                           | Guid        | Unique identifier for a maintenance item.  |
| GroupName                    | String(200) | Item Grouping Name   |
| IntervalFormatted            | String(max) | A textual formatted version of all intervals   |
| IntervalAdjustmentFormatted  | String      | A textual formatted version of all interval adjustments  |
| ItemDescription              | String(255) | The textual description of the item (name).  |
| ItemNumber                   | Integer     | The item number identifier   |
| ItemType                     | Integer     | The type of item this is, which could be part, inspection, AD, etc.  |
| LandingsOnPart               | Integer     | Number of landings the part has at installation  |
| LastCWFormatted              | String(max) | The last complied with formatted   |
| HoursOnPart                  | Decimal     | The number of hours the part has at installation   |
| ManufacturingMaintenanceCode | String(40)  | The manufacturer maintenance code  |
| NextDueDate                  | Date        | The next due date (Format YYYY-MM-DD)  |
| NextDueHours                 | Decimal     | The next due hours   |
| NextDueLandings              | Integer     | The next due landings  |
| NextDueCycles                | Decimal     | The next due cycles  |
| NextDueRIN                   | Integer     | The next due RIN   |
| NextDueFormatted             | String(max) | A formatted text-version of the next due date  |
| PartNumberOn                 | String(40)  | The part number currently on the aircraft, if applicable for the item  |
| Position                     | String(30)  | Information about the position for the item. Open text field for customers to defined, not currently used in reporting.  |
| Reference                    | String(50)  | Reference field typically used for storing the AMM reference number.   |
| RegularNotes                 | String(max) | Text notes   |

| Name                        | Data Type   | Description   |
|-----------------------------|-------------|---|
| RemainingFormatted          | String(max) | The formatted collection of metrics remaining before the next maintenance is due. For date-based items it is a hard date. For all others it is based on the usage preferences for the airframe which can be set in the application. |
| RINOnPart                   | Integer     | The number of RIN the part has at installation  |
| SerialNumberOn              | String(40)  | The serial number of the part currently on the aircraft, if applicable for the item   |
| Tolerance                   | String(50)  | The tolerance or grace period of a maintenance item   |
| TrackedBy                   | String(8)   | The major component this item is tracked by   |
| TrackedByFormatted          | String(max) | The tracked by formatted for display  |
| Aircraft_RegistrationNumber | String(12)  | The current registration or tail number of the aircraft as registered with Flightdocs.  |
| ToleranceFormatted          | String(max) | Tolerance formatted for display   |
| CalculatedDate              | Date        | Calculated due date accounting for anticipated utilization  |
| CalculatedMaxDueDate        | Date        | Calculated due date accounting for anticipated utilization and tolerance  |
| IntervalDays                | Integer     | Required maintenance interval in days   |
| IntervalMonths              | Integer     | Required maintenance interval in months   |
| IntervalHours               | Decimal     | Required maintenance interval in hours  |
| IntervalLandings            | Integer     | Required maintenance interval in landings   |
| IntervalCycles              | Decimal     | Required maintenance interval in cycles   |
| IntervalRIN                 | Decimal     | Required maintenance interval in RIN  |
| IntervalAdjustmentHours     | Decimal     | Adjust the interval hours   |
| IntervalAdjustmentDays      | Integer     | Adjust the interval days  |
| IntervalAdjustmentMonths    | Integer     | Adjust the interval months  |
| IntervalAdjustmentLandings  | Integer     | Adjust the interval landings  |
| IntervalAdjustmentCycles    | Decimal     | Adjust the interval cycles  |
| IntervalAdjustmentRIN       | Decimal     | Adjust the interval RIN   |
| LogDate                     | Date        | Current log date  |
| LogHours                    | Decimal     | Total logged hours  |
| LogLandings                 | Integer     | Total logged landings   |
| LogCycles                   | Decimal     | Total logged cycles   |
| LogRIN                      | Decimal     | Total logged RIN  |
| RemainingHours              | Decimal     | Aircraft hours remaining before due   |
| RemainingLandings           | Integer     | Aircraft landings remaining before due  |
| RemainingRIN                | Decimal     | Aircraft RIN remaining before due   |
| RemainingCycles             | Decimal     | Aircraft cycles remaining before due  |
| RemainingMonths             | Integer     | Months remaining before due   |
| RemainingDays               | Integer     | Days remaining before due   |



| Name                       | Data Type | Description  |
|----------------------------|-----------|--|
| ToleranceMonths            | Integer   | Acceptable variance in months from due date          |
| ToleranceDays              | Integer   | Acceptable variance in days from due date            |
| ToleranceHours             | Decimal   | Acceptable variance in hours                         |
| ToleranceLandings          | Integer   | Acceptable variance in landings                      |
| ToleranceCycles            | Decimal   | Acceptable variance in cycles                        |
| ToleranceRIN               | Decimal   | Acceptable variance in RIN                           |
| ToleranceRemainingMonths   | Integer   | Months remaining in tolerance                        |
| ToleranceRemainingDays     | Integer   | Days remaining in tolerance                          |
| ToleranceRemainingHours    | Decimal   | Hours remaining in tolerance                         |
| ToleranceRemainingLandings | Integer   | Landings remaining in tolerance                      |
| ToleranceRemainingCycles   | Decimal   | Cycles remaining in tolerance                        |
| ToleranceRemainingRIN      | Decimal   | RIN remaining in tolerance                           |
| ToleranceNextDueDate       | Date      | Date due with tolerance included                     |
| ToleranceNextDueHours      | Decimal   | Hours remaining until due with tolerance included    |
| ToleranceNextDueLandings   | Integer   | Landings remaining until due with tolerance included |
| ToleranceNextDueCycles     | Decimal   | Cycles remaining until due with tolerance included   |
| ToleranceNextDueRIN        | Decimal   | RIN remaining until due with tolerance included      |
| AircraftUtilizationHours   | Decimal   | Anticipated utilization of aircraft in hours         |
| AircraftUtilizationCycles  | Decimal   | Anticipated utilization of aircraft in cycles        |
| AircraftUtilizationRIN     | Decimal   | Anticipated utilization of aircraft in RIN           |

#### Example Request (API Version 2):

```

https://api.flightdocs.com/MaintenanceItem/GetDueList?RegistrationNumber=N900EX&IncludePaging=
true&PageSize=1&PageIndex=0 HTTP/1.1
Host:"api.flightdocs.com"
Accept:"*/*"
User-Agent:{User Agent}
Authorization: {bearer_token_here}
api-version:"2"
Time-Zone: America/New_York

```

#### Example Response (API Version 2):

```

{
  "HttpStatusCode": 200,
  "Data": [
    {
      "AircraftId": 4603,
      "Aircraft_RegistrationNumber": "N900EX",
      "ATACode": 54,
      "ATADisplay": "54 00-00-220-802",
      "TrackedBy": "Airframe "
    }
  ]
}

```

```
"DueStatus": 5,  
"Disposition": "",  
"Id": "53deaf80-ae74-4d00-99f8-53d97b478cca",  
"GroupName": "",  
"IsParentItem": false,  
"IsChildItem": false,  
"ItemDescription": "DETL INSP/CHK #2 ENG NAC&PYL",  
"ItemNumber": 7811,  
"ItemType": "PART",  
"ManufacturingMaintenanceCode": "00-00-220-802",  
"PartNumberOn": "6969",  
"Position": "",  
"Reference": "",  
"RegularNotes": "",  
"SerialNumberOn": "553",  
"IntervalFormatted": "L: 3750\r\n",  
"IntervalAdjustmentFormatted": "",  
"LastCWFormatted": "20-NOV-2012\r\nH: 8112.26 \r\nL: 7117 \r\n",  
"NextDueFormatted": "L: 10867\r\n",  
"RemainingFormatted": "L: -5692\r\n",  
"ToleranceFormatted": "",  
"CalculatedDueDate": "2011-05-13",  
"CalculatedMaxDueDate": "2011-05-13",  
"CyclesOnPart": null,  
"LandingsOnPart": null,  
"HoursOnPart": null,  
"RINOnPart": null,  
"IntervalDays": 0,  
"IntervalMonths": 0,  
"IntervalHours": null,  
"IntervalLandings": 3750,  
"IntervalCycles": 0,  
"IntervalRIN": 0,  
"IntervalAdjustmentHours": 0,  
"IntervalAdjustmentDays": 0,  
"IntervalAdjustmentMonths": 0,  
"IntervalAdjustmentLandings": 0,  
"IntervalAdjustmentCycles": 0,  
"IntervalAdjustmentRIN": 0,  
"LogDate": "2012-11-20",  
"LogHours": 8112.26,  
"LogLandings": 7117,  
"LogCycles": 0,  
"LogRIN": null,  
"NextDueDate": null,  
"NextDueHours": null,
```

```
"NextDueLandings": 10867,
"NextDueCycles": null,
"NextDueRIN": null,
"RemainingHours": null,
"RemainingLandings": -5692,
"RemainingRIN": null,
"RemainingCycles": null,
"RemainingMonths": null,
"RemainingDays": null,
"ToleranceMonths": null,
"ToleranceDays": null,
"ToleranceHours": null,
"ToleranceLandings": null,
"ToleranceCycles": null,
"ToleranceRIN": null,
"ToleranceRemainingMonths": null,
"ToleranceRemainingDays": null,
"ToleranceRemainingHours": null,
"ToleranceRemainingLandings": null,
"ToleranceRemainingCycles": null,
"ToleranceRemainingRIN": null,
"ToleranceNextDueDate": null,
"ToleranceNextDueHours": null,
"ToleranceNextDueLandings": null,
"ToleranceNextDueCycles": null,
"ToleranceNextDueRIN": null,
"AircraftUtilizationHours": 1,
"AircraftUtilizationLandings": 2,
"AircraftUtilizationCycles": -1,
"AircraftUtilizationRIN": -1
}
],
"Warnings": [],
"Errors": [],
"Messages": [],
"IsSuccess": true,
"AvailableRecordCount": 0,
"ErrorMessage": "No Errors Reported"
}
```

## GET /NonRoutineMaintenanceItem/Search/

Allow the searching of non-routine maintenance items (like discrepancies) for a particular aircraft.

### Parameters

| Name                   | Data Type  | Required | Description  |
|------------------------|------------|----------|--|
| RegistrationNumber     | String(12) | X        | This is the current registration or tail number of the aircraft as registered with Flightdocs.   |
| StatusId               | Integer    |          | Status of the non-routine item. Valid values are: <ul style="list-style-type: none"> <li>• 1 – Open</li> <li>• 2 – In-Progress</li> <li>• 3 – Deferred</li> <li>• 4 - Closed</li> </ul>  |
| ItemTypeId             | Integer    |          | The type of non-routine item this is. Valid values are: <ul style="list-style-type: none"> <li>• 5 – Discrepancy</li> <li>• 6 – MEL</li> <li>• 7 – NEF</li> <li>• 8 – CDL</li> <li>• 9 – Watch List</li> </ul>   |
| ATACode                | Integer    |          | The ATA code for the item.   |
| DateReported           | Date       |          | Filters the non-routine items by the reported date. Usage is based on the DateReportedConstraint filter. (Formatted as YYYY-MM-DD)   |
| DateReportedSecondary  | Date       |          | Secondary date field used when the DateReportedConstraint is set to “2” which looks for values between the DateReported field and this field. (Formatted as YYYY-MM-DD)  |
| DateReportedConstraint | Integer    |          | Determines how the DateReported and DateReportedSecondary are used. Valid values are 1 – Equals (only DateReported is used and an exact date match); 2 – Between (both DateReported and DateReportedSecondary are used to find items between the dates); 3 – Before (only DateReported is used which find items reported before the provided date); 4 – After (only DateReported is used which finds items reported after the provided date) |
| LogDate                | Date       |          | Filters the non-routine items by the log date. Usage is based on the LogDateConstraint filter. (Formatted as YYYY-MM-DD)   |
| LogDateSecondary       | Date       |          | Secondary date field used when the LogDateConstraint is set to “2” which looks for values between the LogDate field and this field. (Formatted as YYYY-MM-DD)  |

|                   |             |   |
|-------------------|-------------|---|
| LogDateConstraint | Integer     | Determines how the LogDate and LogDateSecondary are used. Valid values are 1 – Equals (only LogDate is used and an exact date match); 2 – Between (both LogDate and LogDateSecondary are used to find items between the dates); 3 – Before (only LogDate is used which find items reported before the provided date); 4 – After (only LogDate is used which finds items reported after the provided date) |
| Reference         | String(50)  | Reference field typically used for storing the AMM reference number.  |
| DiscoveryPoint    | String(250) | Discovery Point of the non-routine item.  |
| IsOptional        | Boolean     | If supplied would return non-routine item that are flagged as optional or not optional.   |
| DiscrepancyTypeId | Integer     | Valid values: <ul style="list-style-type: none"> <li>• 1 – Pirep</li> <li>• 2 - Non-Routine</li> </ul>  |
| PageSize          | Integer     | Number for records to return in each page of the results.   |
| PageIndex         | Integer     | Indicator of which page of the results to return.   |
| IncludePaging     | Boolean     | Indicator if the results should be paged.   |

Return Values (Array)

| Name               | Data Type  | Nullable | Description   |
|--------------------|------------|----------|---|
| Id                 | Guid       |          | Internal database identifier each non-routine item  |
| EquipmentId        | Integer    |          | Id of the equipment that this item is found on. Id backing the TrackedBy column   |
| ItemNumber         | Integer    |          | Unique number for the maintenance item within the equipment.  |
| RegistrationNumber | String(12) |          | Tail number of the aircraft   |
| TrackedBy          | String(50) |          | Description of the equipment and the position of the equipment on the aircraft. i.e. Engine 1   |
| Number             | String(50) |          | Unique system generated number for the non-routine item.  |
| StatusId           | Integer    |          | Id indicating the Status: <ul style="list-style-type: none"> <li>• 1 – Open</li> <li>• 2 – In-Progress</li> <li>• 3 – Deferred</li> <li>• 4 – Closed</li> </ul> |
| StatusName         | String(20) |          | Text name of the status of the non-routine item.  |
| IsOptional         | Boolean    |          | Indicator if the non-routine is an optional item  |

| Name                         | Data Type   | Nullable | Description   |
|------------------------------|-------------|----------|---|
| ControlNumber                | String(20)  |          | Control Number of the non-routine item.   |
| Type                         | String(50)  |          | Text name of the non-routine item type: Discrepancy, MEL, NEF, CDL, Watch List.   |
| ATACode                      | Integer     | X        | ATA Chapter of the non-routine item   |
| ManufacturingMaintenanceCode | String(40)  |          | Manufacture code for the non-routine item.  |
| Description                  | String(700) |          | Text to describe the non-routine maintenance item.  |
| LastUpdated                  | DateTime    | X        | Date and Time of when the non-routine item was last updated.  |
| DateReported                 | Date        | X        | Date of when the non-routine item was reported. (format YYYY-MM-DD)   |
| ReportedHours                | Decimal     | X        | The number of hours on the equipment when the item was reported   |
| ReportedLandings             | Integer     | X        | The number of landings on the equipment when the item was reported.   |
| ReportedCycles               | Decimal     | X        | The number of cycles on the equipment when the item was reported.   |
| ReportedRIN                  | Decimal     | X        | The number of RIN on the equipment when the item was reported.  |
| Category                     | String(10)  |          | Category of the non-routine item: <ul style="list-style-type: none"> <li>• A – Multiple Metrics</li> <li>• B – Date (In 3 Days)</li> <li>• C – Date (In 10 Days)</li> <li>• D – Date (In 120 Days)</li> </ul> |
| HardNextDueDate              | Date        | X        | The date of when the non-routine item is due. (format YYYY-MM-DD)   |
| HardNextDueHours             | Decimal     | X        | The hours on the equipment when the item is due.  |
| HardNextDueLandings          | Integer     | X        | The landings on the equipment when the item is due.   |
| HardNextDueCycles            | Decimal     | X        | The cycles on the equipment when the item is due.   |
| HardNextDueRIN               | Decimal     | X        | The RIN on the equipment when the item is due.  |
| DiscrepancyTypeid            | Integer     | X        | The id of the discrepancy type: <ul style="list-style-type: none"> <li>• 1 – Pirep</li> <li>• 2 - Non-Routine</li> </ul>  |
| DiscrepancyTypeName          | String(20)  |          | Text display of the discrepancy type.   |
| DelayStatus                  | String(20)  |          | Delay status of the non-routine item: <ul style="list-style-type: none"> <li>• Delay</li> <li>• No Delay</li> <li>• Deviation</li> </ul>  |

| Name                           | Data Type   | Nullable | Description   |
|--------------------------------|-------------|----------|---|
| ActionCode                     | String(20)  |          | <ul style="list-style-type: none"> <li>• Cancellation</li> </ul> Non-routines item's action code: <ul style="list-style-type: none"> <li>• Unspecified</li> <li>• MISR</li> <li>• SDR</li> <li>• ETOPS</li> </ul> |
| PreliminaryInspectionRequired  | Boolean     | X        | Indicator if a preliminary inspection is required.  |
| IsAircraftGrounded             | Boolean     | X        | Indicator if the aircraft was grounded because of the non-routine item.   |
| Reference                      | String(50)  |          | Reference field of the non-routine item.  |
| HiddenDamageInspectionRequired | Boolean     | X        | Indicator if a hidden damage inspection is required.  |
| DiscoveryPoint                 | String(250) |          | Description of the discovery point.   |
| DiscoveredBy                   | String(65)  |          | Indicator of who discovered the non-routine item.   |
| ReportedInformationNotes       | String(max) |          | Notes that went along with the reported information.  |
| InterruptionHours              | Decimal     | X        | Interruption Hours  |
| WorkHours                      | Decimal     | X        | Hours spent addressing the non-routine item.  |
| ResolvedAt                     | String(150) |          | Description of when/where the item was resolved.  |
| CorrectiveActionNotes          | String(max) |          | Notes to go along with the corrective action information.   |
| ResolvedBy                     | String(150) |          | Who resolved the non-routine item.  |
| LogDate                        | Date        |          | Date of when the item as completed.   |
| LogCycles                      | Decimal     | X        | Cycles on the equipment when the item was completed.  |
| LogHours                       | Decimal     | X        | Hours on the equipment when the item was completed.   |
| LogLandings                    | Integer     | X        | Landings on the equipment when the item was completed.  |
| LogRIN                         | Decimal     | X        | RIN on the equipment when the item was completed.   |
| Created                        | DateTime    |          | Date and time of when the non-routine item was created.   |

Example Request:

```
GET https://api.flightdocs.com/NonRoutineMaintenanceItem/Search?RegistrationNumber=NHOOLI1
HTTP/1.1
Host: api.flightdocs.com
Authorization: Bearer {bearer_token_here}
```

Example Response:

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Date: Tue, 09 Apr 2019 22:33:03 GMT
Content-Length: 93479
```

```
{
  "HttpStatusCode": 200,
  "Data": [
    {
      "Id": "a7a5c08e-623d-4380-b3fa-31c62c7f3b3e",
      "EquipmentId": 266,
      "ItemNumber": 5180,
      "RegistrationNumber": "NHOOLI1",
      "TrackedBy": "Airframe ",
      "Number": "201609101442",
      "StatusId": 2,
      "StatusName": "In-Progress",
      "IsOptional": false,
      "ControlNumber": "",
      "Type": "MEL",
      "ATACode": 22,
      "ManufacturingMaintenanceCode": "",
      "Description": "Yaw damp fail cass message",
      "LastUpdated": "2017-02-10T21:44:27.1841209Z",
      "DateReported": "2017-02-03",
      "ReportedHours": null,
      "ReportedLandings": null,
      "ReportedCycles": null,
      "ReportedRIN": null,
      "Category": "C",
      "HardNextDueDate": "2017-02-13",
      "HardNextDueHours": 0,
      "HardNextDueLandings": 0,
      "HardNextDueCycles": 0,
    }
  ]
}
```



```
"HardNextDueRIN": null,
"DiscrepancyTypeId": 1,
"DiscrepancyTypeName": "Pirep",
"DelayStatus": "No Delay",
"ActionCode": "Unspecified",
"PreliminaryInspectionRequired": false,
"IsAircraftGrounded": false,
"Reference": "",
"HiddenDamageInspectionRequired": false,
"DiscoveryPoint": "",
"DiscoveredBy": "",
"ReportedInformationNotes": "",
"InterruptionHours": null,
"WorkHours": 0,
"ResolvedAt": "",
"CorrectiveActionNotes": "Removed GSA 81 Servo P/N 011-00878-10 S/N 123XXX1 and installed
serviceable P/N 011-00878-20 S/N 123XXX2 IAW EMB Phenom 100 AMM 22-11-08-400-801-A. Ops
Check good\r\n",
"ResolvedBy": "",
"LogDate": "2016-09-13",
"LogCycles": null,
"LogHours": 6194.8,
"LogLandings": 5402,
"LogRIN": null,
"Created": "0001-01-01T00:00:00Z"
},
{
  "Id": "c1c6e6da-db4e-4447-8112-f9e32861ffb4",
  "EquipmentId": 266,
  "ItemNumber": 5186,
  "RegistrationNumber": "NHOOLI1",
  "TrackedBy": "Airframe ",
  "Number": "201609161914",
  "StatusId": 1,
  "StatusName": "Open",
  "IsOptional": false,
  "ControlNumber": null,
  "Type": "WATCH LIST",
  "ATACode": 25,
  "ManufacturingMaintenanceCode": "",
  "Description": "#1 tray table will not open",
  "LastUpdated": "2017-02-10T21:45:13.3599808Z",
  "DateReported": "2016-09-16",
  "ReportedHours": null,
  "ReportedLandings": null,
  "ReportedCycles": null,
```

```
"ReportedRIN": null,  
"Category": "A",  
"HardNextDueDate": null,  
"HardNextDueHours": 0,  
"HardNextDueLandings": 0,  
"HardNextDueCycles": 0,  
"HardNextDueRIN": null,  
"DiscrepancyTypeId": 1,  
"DiscrepancyTypeName": "Pirep",  
"DelayStatus": "No Delay",  
"ActionCode": "Unspecified",  
"PreliminaryInspectionRequired": false,  
"IsAircraftGrounded": false,  
"Reference": "",  
"HiddenDamageInspectionRequired": false,  
"DiscoveryPoint": "",  
"DiscoveredBy": "",  
"ReportedInformationNotes": "",  
"InterruptionHours": null,  
"WorkHours": 0,  
"ResolvedAt": "",  
"CorrectiveActionNotes": "Cleaned debris from table tray as required ops check good.\r\n",  
"ResolvedBy": "",  
"LogDate": "2016-09-16",  
"LogCycles": null,  
"LogHours": 6209.5,  
"LogLandings": 5413,  
"LogRIN": null,  
"Created": "0001-01-01T00:00:00Z"  
}  
],  
"Warnings": [],  
"Errors": [],  
"Messages": [],  
"IsSuccess": true,  
"AvailableRecordCount": 0,  
"ErrorMessage": "No Errors Reported"  
}
```

## POST /NonRoutineMaintenanceItem/Create/

**Version 1** - Allow the creation of non-routine maintenance items (like discrepancies) for a particular aircraft.

**Notes:** While the EquipmentId is not required, if it is not supplied then the non-routine item will be created on the airframe of the aircraft. If the discrepancy or other type of non-routine is found on an engine for instance the /Aircraft/GetEquipment/ method should be called so that a list of the equipment on the aircraft can be presented to the user and the appropriate EquipmentId can be sent in with the create.

### Parameters

| Name                         | Data Type   | Required | Description  |
|------------------------------|-------------|----------|--|
| RegistrationNumber           | String(12)  | X        | This is the current registration or tail number of the aircraft as registered with Flightdocs.   |
| EquipmentId                  | Integer     |          | Id of the equipment that this item is to be created on.  |
| IsOptional                   | Boolean     |          | Indicator if the non-routine is an optional item   |
| ControlNumber                | String(20)  |          | Control Number of the non-routine item.  |
| ItemTypeId                   | Integer     | X        | The type of non-routine item: <ul style="list-style-type: none"> <li>• 5 – Discrepancy</li> <li>• 6 – MEL</li> <li>• 7 – NEF</li> <li>• 8 – CDL</li> <li>• 9 – Watch List</li> </ul> |
| ATACode                      | Integer     |          | ATA Chapter of the non-routine item  |
| ManufacturingMaintenanceCode | String(40)  |          | Manufacture code for the non-routine item.   |
| Description                  | String(700) | X        | Text to describe the non-routine maintenance item.   |
| LastUpdated                  | DateTime    |          | Date and Time of when the non-routine item was last updated.   |
| DateReported                 | Date        |          | Date of when the non-routine item was reported. (format YYYY-MM-DD)  |
| ReportedHours                | Decimal     |          | The number of hours on the equipment when the item was reported  |
| ReportedLandings             | Integer     |          | The number of landings on the equipment when the item was reported.  |
| ReportedCycles               | Decimal     |          | The number of cycles on the equipment when the item was reported.  |

| Name                           | Data Type   | Required | Description   |
|--------------------------------|-------------|----------|---|
| ReportedRIN                    | Decimal     |          | The number of RIN on the equipment when the item was reported.  |
| Category                       | String(1)   |          | Category of the non-routine item. <ul style="list-style-type: none"> <li>• A – Multiple Metrics</li> <li>• B – Date (In 3 Days)</li> <li>• C – Date (In 10 Days)</li> <li>• D – Date (In 120 Days)</li> </ul> |
| HardNextDueDate                | Date        |          | The date of when the non-routine item is due. (format YYYY-MM-DD)   |
| HardNextDueHours               | Decimal     |          | The hours on the equipment when the item is due.  |
| HardNextDueLandings            | Integer     |          | The landings on the equipment when the item is due.   |
| HardNextDueCycles              | Decimal     |          | The cycles on the equipment when the item is due.   |
| HardNextDueRIN                 | Decimal     |          | The RIN on the equipment when the item is due.  |
| DiscrepancyTypeid              | Integer     |          | The id of the discrepancy type: <ul style="list-style-type: none"> <li>• 1 – Pirep</li> <li>• 2 - Non-Routine</li> </ul>  |
| DelayStatus                    | String(20)  |          | Delay status of the non-routine item: <ul style="list-style-type: none"> <li>• Delay</li> <li>• No Delay</li> <li>• Deviation</li> <li>• Cancellation</li> </ul>  |
| ActionCode                     | String(20)  |          | Non-routines item’s action code: <ul style="list-style-type: none"> <li>• Unspecified</li> <li>• MISR</li> <li>• SDR</li> <li>• ETOPS</li> </ul>  |
| PreliminaryInspectionRequired  | Boolean     |          | Indicator if a preliminary inspection is required.  |
| Reference                      | String(50)  |          | Reference field of the non-routine item.  |
| HiddenDamageInspectionRequired | Boolean     |          | Indicator if a hidden damage inspection is required.  |
| DiscoveryPoint                 | String(50)  |          | Description of the discovery point.   |
| DiscoveredBy                   | String(50)  |          | Indicator of who discovered the non-routine item.   |
| ReportedInformationNotes       | String(max) |          | Notes to be included with the reported information.   |

## Return Values

| Name        | Data Type  | Description   |
|-------------|------------|---|
| Id          | Guid       | Internal database identifier of the newly created non-routine item              |
| EquipmentId | Integer    | Id of the equipment that this item is found on. Id backing the TrackedBy column |
| ItemNumber  | Integer    | Unique number for the maintenance item within the equipment.                    |
| Number      | String(50) | Unique system generated number for the non-routine item.                        |

### Example Request:

```
POST https://api.flightdocs.com/NonRoutineMaintenanceItem/Create HTTP/1.1
Host: api.flightdocs.com
Accept: application/json, text/plain, */*
Content-Type: application/json
Authorization: Bearer {bearer_token_here}
```

```
{
  "RegistrationNumber": "NHOOL1",
  "EquipmentId": 267,
  "ItemTypeId": 5,
  "Description": "At FL 390 Rad Alt alerted to 800 ft momentarily"
}
```

### Example Response:

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Date: Wed, 10 Apr 2019 01:25:26 GMT
Content-Length: 256
```

```
{
  "HttpStatusCode": 200,
  "Data": {
    "Id": "9be6df3a-e3a7-427c-87c3-cd686a498502",
    "EquipmentId": 267,
    "ItemNumber": 5209,
    "Number": "20190410012522"
  },
  "Warnings": [],
}
```

```

"Errors": [],
"Messages": [],
"IsSuccess": true,
"AvailableRecordCount": 0,
"ErrorMessages": "No Errors Reported"
}

```

## POST /NonRoutineMaintenanceItem/Create/

**Version 2** - Allow the creation of non-routine maintenance items (like discrepancies) for a particular aircraft with the status of Aircraft On Ground (AOG) if the discrepancy being created causes the aircraft to be grounded.

**Notes:** While the EquipmentId is not required, if it is not supplied then the non-routine item will be created on the airframe of the aircraft. If the discrepancy or other type of non-routine is found on an engine for instance the /Aircraft/GetEquipment/ method should be called so that a list of the equipment on the aircraft can be presented to the user and the appropriate EquipmentId can be sent in with the create.

### Parameters

| Name                         | Data Type   | Required | Description  |
|------------------------------|-------------|----------|--|
| RegistrationNumber           | String(12)  | X        | This is the current registration or tail number of the aircraft as registered with Flightdocs.   |
| IsAircraftGrounded           | Boolean     |          | Indicator if the non-routine item causes the aircraft to be grounded.  |
| EquipmentId                  | Integer     |          | Id of the equipment that this item is to be created on.  |
| IsOptional                   | Boolean     |          | Indicator if the non-routine is an optional item   |
| ControlNumber                | String(20)  |          | Control Number of the non-routine item.  |
| ItemTypeId                   | Integer     | X        | The type of non-routine item: <ul style="list-style-type: none"> <li>• 5 – Discrepancy</li> <li>• 6 – MEL</li> <li>• 7 – NEF</li> <li>• 8 – CDL</li> <li>• 9 – Watch List</li> </ul> |
| ATACode                      | Integer     |          | ATA Chapter of the non-routine item  |
| ManufacturingMaintenanceCode | String(40)  |          | Manufacture code for the non-routine item.   |
| Description                  | String(700) | X        | Text to describe the non-routine maintenance item.   |

| Name                          | Data Type  | Required | Description   |
|-------------------------------|------------|----------|---|
| LastUpdated                   | DateTime   |          | Date and Time of when the non-routine item was last updated.  |
| DateReported                  | Date       |          | Date of when the non-routine item was reported. (format YYYY-MM-DD)   |
| ReportedHours                 | Decimal    |          | The number of hours on the equipment when the item was reported   |
| ReportedLandings              | Integer    |          | The number of landings on the equipment when the item was reported.   |
| ReportedCycles                | Decimal    |          | The number of cycles on the equipment when the item was reported.   |
| ReportedRIN                   | Decimal    |          | The number of RIN on the equipment when the item was reported.  |
| Category                      | String(1)  |          | Category of the non-routine item. <ul style="list-style-type: none"> <li>• A – Multiple Metrics</li> <li>• B – Date (In 3 Days)</li> <li>• C – Date (In 10 Days)</li> <li>• D – Date (In 120 Days)</li> </ul> |
| HardNextDueDate               | Date       |          | The date of when the non-routine item is due. (format YYYY-MM-DD)   |
| HardNextDueHours              | Decimal    |          | The hours on the equipment when the item is due.  |
| HardNextDueLandings           | Integer    |          | The landings on the equipment when the item is due.   |
| HardNextDueCycles             | Decimal    |          | The cycles on the equipment when the item is due.   |
| HardNextDueRIN                | Decimal    |          | The RIN on the equipment when the item is due.  |
| DiscrepancyTypeid             | Integer    |          | The id of the discrepancy type: <ul style="list-style-type: none"> <li>• 1 – Pirep</li> <li>• 2 - Non-Routine</li> </ul>  |
| DelayStatus                   | String(20) |          | Delay status of the non-routine item: <ul style="list-style-type: none"> <li>• Delay</li> <li>• No Delay</li> <li>• Deviation</li> <li>• Cancellation</li> </ul>  |
| ActionCode                    | String(20) |          | Non-routines item’s action code: <ul style="list-style-type: none"> <li>• Unspecified</li> <li>• MISR</li> <li>• SDR</li> <li>• ETOPS</li> </ul>  |
| PreliminaryInspectionRequired | Boolean    |          | Indicator if a preliminary inspection is required.  |
| Reference                     | String(50) |          | Reference field of the non-routine item.  |

| Name                           | Data Type   | Required | Description  |
|--------------------------------|-------------|----------|--|
| HiddenDamageInspectionRequired | Boolean     |          | Indicator if a hidden damage inspection is required. |
| DiscoveryPoint                 | String(50)  |          | Description of the discovery point.                  |
| DiscoveredBy                   | String(50)  |          | Indicator of who discovered the non-routine item.    |
| ReportedInformationNotes       | String(max) |          | Notes to be included with the reported information.  |

#### Return Values

| Name        | Data Type  | Description   |
|-------------|------------|---|
| Id          | Guid       | Internal database identifier of the newly created non-routine item              |
| EquipmentId | Integer    | Id of the equipment that this item is found on. Id backing the TrackedBy column |
| ItemNumber  | Integer    | Unique number for the maintenance item within the equipment.                    |
| Number      | String(50) | Unique system generated number for the non-routine item.                        |

#### Example Request:

```
POST https://api.flightdocs.com/NonRoutineMaintenanceItem/Create HTTP/1.1
Host: api.flightdocs.com
Accept: application/json, text/plain, */*
Content-Type: application/json
Authorization: Bearer {bearer_token_here}
```

```
{
  "RegistrationNumber": "NHOOL1",
  "IsAircraftGrounded": false,
  "EquipmentId": 267,
  "ItemTypeId": 5,
  "Description": "At FL 390 Rad Alt alerted to 800 ft momentarily"
}
```

#### Example Response:

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
```



Content-Type: application/json; charset=utf-8

Expires: -1

Date: Wed, 10 Apr 2019 01:25:26 GMT

Content-Length: 256

```
{
  "HttpStatusCode": 200,
  "Data": {
    "Id": "9be6df3a-e3a7-427c-87c3-cd686a498502",
    "EquipmentId": 267,
    "ItemNumber": 5209,
    "Number": "20190410012522"
  },
  "Warnings": [],
  "Errors": [],
  "Messages": [],
  "IsSuccess": true,
  "AvailableRecordCount": 0,
  "ErrorMessages": "No Errors Reported"
}
```

Last Updated: 10/7/2021