

LiteBulb Labs

RunLog Application

Rest API

API Documentation v0.1

**Abstract**

This document outlines the RunLog application REST API functionality and features and the proper formation of requests and their associated response data structures.

John Bianchi

jmbianchi@gmail.com

Table of Contents

[Sample Input 3](#_Toc46938313)

[Sample Output 4](#_Toc46938314)

[Assumptions 6](#_Toc46938315)

[Methods – Get Activities 7](#_Toc46938316)

[Methods – Get Activity (by Id) 10](#_Toc46938317)

[Methods – Create Activity 12](#_Toc46938318)

[Methods – Update Activity 15](#_Toc46938319)

[Methods – Delete Activity 17](#_Toc46938320)

[Methods – Get Positions 19](#_Toc46938321)

[Methods – Get Position (by Id) 21](#_Toc46938322)

[Methods – Create Position 23](#_Toc46938323)

[Methods – Update Position 25](#_Toc46938324)

[Methods – Delete Position 27](#_Toc46938325)

[Glossary 29](#_Toc46938326)

# Sample Input

The input is a JSON string representing an Activity object and its associated collection of Position child objects.

See below for an example of the input.

**Activity**

|  |
| --- |
| {  "runnerName": "string",  "type": 0,  "status": 0,  "createdAt": "2020-07-29T21:30:27.190Z",  "startedAt": "2020-07-29T21:30:27.190Z",  "completedAt": "2020-07-29T21:30:27.190Z",  "description": "string",  "units": 0,  "positions": [  {  "activityId": 0,  "timeStamp": "2020-07-29T21:30:27.190Z",  "latitude": 0,  "longitude": 0,  "altitude": 0,  "accuracy": 0,  "altitudeAccuracy": 0,  "heading": 0,  "speed": 0,  "satelliteCount": 0,  "id": 0 Note: id field is ignored on create  }  ],  "id": 0 Note: id field is ignored on create  } |

**Position**

|  |
| --- |
| {  "activityId": 0,  "timeStamp": "2020-07-29T21:31:15.123Z",  "latitude": 0,  "longitude": 0,  "altitude": 0,  "accuracy": 0,  "altitudeAccuracy": 0,  "heading": 0,  "speed": 0,  "satelliteCount": 0,  "id": 0 Note: id field is ignored on create  } |

# Sample Output

The output is a JSON string representing the new Activity/Position objects created in the system.

See below for an example of the output.

**Activity**

|  |
| --- |
| [  {  "RunnerName": "string",  "Type": 0,  "Status": 0,  "CreatedAt": "2020-07-29T21:32:02.961Z",  "StartedAt": "2020-07-29T21:32:02.961Z",  "CompletedAt": "2020-07-29T21:32:02.961Z",  "Description": "string",  "Units": 0,  "Positions": [  {  "ActivityId": 1,  "TimeStamp": "2020-07-29T21:32:02.961Z",  "Latitude": 0,  "Longitude": 0,  "Altitude": 0,  "Accuracy": 0,  "AltitudeAccuracy": 0,  "Heading": 0,  "Speed": 0,  "SatelliteCount": 0,  "Id": 1  }  ],  "Id": 1  },  {  "RunnerName": "string",  "Type": 0,  "Status": 0,  "CreatedAt": "2020-07-29T21:32:02.961Z",  "StartedAt": "2020-07-29T21:32:02.961Z",  "CompletedAt": "2020-07-29T21:32:02.961Z",  "Description": "string",  "Units": 0,  "Positions": [  {  "ActivityId": 2,  "TimeStamp": "2020-07-29T21:32:02.961Z",  "Latitude": 0,  "Longitude": 0,  "Altitude": 0,  "Accuracy": 0,  "AltitudeAccuracy": 0,  "Heading": 0,  "Speed": 0,  "SatelliteCount": 0,  "Id": 2  }  ],  "Id": 2  }  ] |

**Position**

|  |
| --- |
| **[**  **{**  **"ActivityId": 1,**  **"TimeStamp": "2020-07-29T21:32:02.961Z",**  **"Latitude": 0,**  **"Longitude": 0,**  **"Altitude": 0,**  **"Accuracy": 0,**  **"AltitudeAccuracy": 0,**  **"Heading": 0,**  **"Speed": 0,**  **"SatelliteCount": 0,**  **"Id": 1**  **},**  **{**  **"ActivityId": 2,**  **"TimeStamp": "2020-07-29T21:32:02.961Z",**  **"Latitude": 0,**  **"Longitude": 0,**  **"Altitude": 0,**  **"Accuracy": 0,**  **"AltitudeAccuracy": 0,**  **"Heading": 0,**  **"Speed": 0,**  **"SatelliteCount": 0,**  **"Id": 2**  **}**  **]** |

# Assumptions

1. API will not implement authentication and authorization in the first version.

2. Authentication and authorization features will be addressed later.

3. DTO (data transfer object) data input validation features will be added later.

4. List results paging functionality will be added later.

5. API key functionary will be added later.

6. PATCH update functionality has not been implemented yet.

7. Correlation ID in header has not been implemented yet.

# Methods – Get Activities

**Description**

A list of all RunLog Activity objects created in the system (that have not been deleted yet). The response is paged (default is 0 offset and 50 limit).

**Request URI:**

/api/v1/activities/<offset>,<limit>

*(Example: /api/v1/activities/0,50)*

**Http Verb:**

GET

**Http Status Codes:**

200 – “OK”

404 – “Not Found”

Example error messages:

Not Found: “Error in GetActivitiesAsync() method. Activity list cannot be displayed. Error message returned from ActivityService: {ErrorMessage}”

**Supported Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Param** | **Value** | **Comments** |
| HEAD | api\_key | string | Must be sent with all client requests. Helps the server to validate the request source. |
| URL\_PARAM | offset | int (nullable) | Number of Activity records to skip before retrieving list. Default is 0.  (optional: omit if default values are acceptable) |
| URL\_PARAM | limit | int (nullable) | Number of Activity records to take when retrieving list. Default is 50 and max is 50.  (optional: omit if default values are acceptable) |

**Sample Request Body**

GET {{endpoint}}//api/v1/activities/0,50 HTTP/1.1

Host: api.runlog.com

Authorization: Basic AUTH\_TOKEN

Content-Length: 835

**Sample Response Body**

Response will be an object containing an array of Activity(s). The response is paged.

[

{

"RunnerName": "string",

"Type": 0,

"Status": 0,

"CreatedAt": "2020-07-29T21:32:02.961Z",

"StartedAt": "2020-07-29T21:32:02.961Z",

"CompletedAt": "2020-07-29T21:32:02.961Z",

"Description": "string",

"Units": 0,

"Positions": [

{

"ActivityId": 1,

"TimeStamp": "2020-07-29T21:32:02.961Z",

"Latitude": 0,

"Longitude": 0,

"Altitude": 0,

"Accuracy": 0,

"AltitudeAccuracy": 0,

"Heading": 0,

"Speed": 0,

"SatelliteCount": 0,

"Id": 1

}

],

"Id": 1

},

{

"RunnerName": "string",

"Type": 0,

"Status": 0,

"CreatedAt": "2020-07-29T21:32:02.961Z",

"StartedAt": "2020-07-29T21:32:02.961Z",

"CompletedAt": "2020-07-29T21:32:02.961Z",

"Description": "string",

"Units": 0,

"Positions": [

{

"ActivityId": 2,

"TimeStamp": "2020-07-29T21:32:02.961Z",

"Latitude": 0,

"Longitude": 0,

"Altitude": 0,

"Accuracy": 0,

"AltitudeAccuracy": 0,

"Heading": 0,

"Speed": 0,

"SatelliteCount": 0,

"Id": 2

}

],

"Id": 2

}

]

**Sample Response Headers**

content-length: 835

content-type: application/json; charset=utf-8

date: Tue, 28 Jul 2020 21:26:08 GMT

server: Microsoft-IIS/10.0

status: 200

x-correlation-id: 80000010-0006-ff00-b63f-84710c7967bb

x-powered-by: ASP.NET

# Methods – Get Activity (by Id)

**Description**

Retrieve a single RunLog Activity object from the system (if found).

**Request URI:**

/api/v1/activities/<activityId>

*(Example: /api/v1/activities/1)*

**Http Verb:**

GET

**Http Status Codes:**

200 – “OK”

404 – “Not Found”

Example error messages:

Not Found: “Error in GetActivityByIdAsync() method. Cannot fetch Activity info. Error message returned from ActivityService: {ErrorMessage}”

**Supported Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Param** | **Value** | **Comments** |
| HEAD | api\_key | string | Must be sent with all client requests. Helps the server to validate the request source. |
| URL\_PARAM | activityId | int | Id of the Activity object to retrieve. |

**Sample Request Body**

GET {{endpoint}}//api/v1/activities/1 HTTP/1.1

Host: api.runlog.com

Authorization: Basic AUTH\_TOKEN

Content-Length: 416

**Sample Response Body**

Response will be an Activity object (if id is found).

{

"RunnerName": "string",

"Type": 0,

"Status": 0,

"CreatedAt": "2020-07-29T21:32:02.961Z",

"StartedAt": "2020-07-29T21:32:02.961Z",

"CompletedAt": "2020-07-29T21:32:02.961Z",

"Description": "string",

"Units": 0,

"Positions": [

{

"ActivityId": 1,

"TimeStamp": "2020-07-29T21:32:02.961Z",

"Latitude": 0,

"Longitude": 0,

"Altitude": 0,

"Accuracy": 0,

"AltitudeAccuracy": 0,

"Heading": 0,

"Speed": 0,

"SatelliteCount": 0,

"Id": 1

}

],

"Id": 1

}

**Sample Response Headers**

content-length: 416

content-type: application/json; charset=utf-8

date: Tue, 28 Jul 2020 21:26:08 GMT

server: Microsoft-IIS/10.0

status: 200

x-correlation-id: 80000010-0006-ff00-b63f-84710c7967bb

x-powered-by: ASP.NET

# Methods – Create Activity

**Description**

Create a new Activity object in the system.

**Request URI:**

api/v1/activities

*(Example:* *api/v1/activities)*

**Http Verb:**

POST

**Http Status Codes:**

201 – “Created”

400 – “Bad Request”

404 – “Not Found”

Example error messages:

Bad Request: “Error in CreateActivityAsync() method. Activity object may not have been added to database. Exception was thrown in ActivityService. Exception Type: '{Exception.GetType()}'. Exception Message: {Exception.Message}”

Not Found: “Error in CreateActivityAsync() method. Activity object may not have been added to database. Error message returned from ActivityService: {ErrorMessage}”

**Supported Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Param** | **Value** | **Comments** |
| HEAD | api\_key | string | That was given in response to /api/login. |
| POST | activity | Activity | POCO with all the Activity property values to be used to create a new Activity object in the system. |

**Sample Request Body**

POST {{endpoint}}//api/v1/activities HTTP/1.1

Host: api.runlog.com

Authorization: Basic AUTH\_TOKEN

Content-Length: 416

Content-Type: application/json-patch+json; charset=UTF-8

{

"runnerName": "string",

"type": 0,

"status": 0,

"createdAt": "2020-07-29T21:32:02.961Z",

"startedAt": "2020-07-29T21:32:02.961Z",

"completedAt": "2020-07-29T21:32:02.961Z",

"description": "string",

"units": 0,

"positions": [

{

"activityId": 0,

"timeStamp": "2020-07-29T21:32:02.961Z",

"latitude": 0,

"longitude": 0,

"altitude": 0,

"accuracy": 0,

"altitudeAccuracy": 0,

"heading": 0,

"speed": 0,

"satelliteCount": 0,

"id": 0 Note: id field ignored on create

}

],

"id": 0 Note: id field ignored on create

}

**Sample Response Body**

Response (status code: 201) will be an object containing the id of the new Activity document that was created.

{

"RunnerName": "string",

"Type": 0,

"Status": 0,

"CreatedAt": "2020-07-29T21:32:02.961Z",

"StartedAt": "2020-07-29T21:32:02.961Z",

"CompletedAt": "2020-07-29T21:32:02.961Z",

"Description": "string",

"Units": 0,

"Positions": [

{

"ActivityId": 2,

"TimeStamp": "2020-07-29T21:32:02.961Z",

"Latitude": 0,

"Longitude": 0,

"Altitude": 0,

"Accuracy": 0,

"AltitudeAccuracy": 0,

"Heading": 0,

"Speed": 0,

"SatelliteCount": 0,

"Id": 2

}

],

"Id": 2

}

**Sample Response Headers**

access-control-allow-origin: \*

content-length: 416

content-type: application/json; charset=utf-8

date: Tue, 28 Jul 2020 21:45:21 GMT

location: https://localhost:44349/api/v1/activties/1

server: Microsoft-IIS/10.0

status: 201

x-correlation-id: 80000006-0004-fd00-b63f-84710c7967bb

x-powered-by: ASP.NET

# Methods – Update Activity

**Description**

Update an Activity object (all properties – no PATCH).

**Request URI:**

/api/v1/activities/<activityId>

*(Example: /api/v1/activities/1)*

**Http Verb:**

PUT

**Http Status Codes:**

200 – “OK”

400 – “Bad Request”

404 – “Not Found”

Example error messages:

BadRequest: “The submitted id value '{id}' does not match the corresponding field contained in the submitted object Activity id value '{activity.Id}'.”

NotFound: “Error in UpdateActvitityAsync() method. Cannot find Activity with id '{id}'. Error message returned from ActivityService: {ErrorMessage}”

Not Found: “Error in UpdateActvitityAsync() method. Cannot update Activity with id '{id}'. Error message returned from ActivityService: {ErrorMessage}”

**Supported Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Param** | **Value** | **Comments** |
| HEAD | api\_key | string | Must be sent with all client requests. Helps the server to validate the request source. |
| URL\_PARAM | activityId | int | Id of the Activity object to update. |
| PUT | activity | Activity | POCO with all the Activity object properties that can be updated. |

**Sample Request Body**

PUT {{endpoint}}/api/v1/activities/1 HTTP/1.1

Host: api.runlog.com

Authorization: Basic AUTH\_TOKEN

Content-Length: 233

Content-Type: application/json-patch+json; charset=UTF-8

{

"runnerName": "UPDATED STRING",

"type": 0,

"status": 0,

"createdAt": "2020-07-29T22:02:28.874Z",

"startedAt": "2020-07-29T22:02:28.874Z",

"completedAt": "2020-07-29T22:02:28.874Z",

"description": "string",

"units": 0,

"positions": [], Note: Position collection is ignored by this method

"id": 1 Note: id field in object must match id field in URL\_PARAM

}

**Sample Response Body**

Response will be an Activity object containing all the fields (except Position objects because those were not affected).

Example:

{

"RunnerName": "UPDATED STRING",

"Type": 0,

"Status": 0,

"CreatedAt": "2020-07-29T22:02:28.874Z",

"StartedAt": "2020-07-29T22:02:28.874Z",

"CompletedAt": "2020-07-29T22:02:28.874Z",

"Description": "string",

"Units": 0,

"Positions": [],

"Id": 1

}

**Sample Response Headers**

access-control-allow-origin: \*

content-length: 233

content-type: application/json; charset=utf-8

date: Tue, 28 Jul 2020 01:48:13 GMT

server: Microsoft-IIS/10.0

status: 200

x-correlation-id: 80000005-0000-fb00-b63f-84710c7967bb

x-powered-by: ASP.NET

# Methods – Delete Activity

**Description**

Delete an Activity object from the system.

**Request URI:**

/api/v1/activities/<activityId>

*(Example: /api/v1/activities/1)*

**Http Verb:**

DELETE

**Http Status Codes:**

200 – “OK”

404 – “Resource Not Found”

Example error messages:

“Error in DeleteActivityById() method. Activity object may not have been deleted. Error message returned from ActivityService: {ErrorMessage}”

**Supported Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Param** | **Value** | **Comments** |
| HEAD | api\_key | string | Must be sent with all client requests. Helps the server to validate the request source. |
| URL\_PARAM | activityId | int | Id of the Activity object to be deleted. |

**Sample Request Body**

DELETE {{endpoint}}//api/v1/activities/1 HTTP/1.1

Host: api.runlog.com

Authorization: Basic AUTH\_TOKEN

**Sample Response Body**

Response will be a 200 OK containing a response message.

"Delete successful"

Note: "Delete unsuccessful" if nothing was deleted

**Sample Response Header**

access-control-allow-origin: \*

content-encoding: gzip

content-type: text/plain; charset=utf-8

date: Tue, 28 Jul 2020 23:30:45 GMT

server: Microsoft-IIS/10.0

status: 200

vary: Accept-Encoding

x-correlation-id: 80000012-0006-ff00-b63f-84710c7967bb

x-powered-by: ASP.NET

# Methods – Get Positions

**Description**

A list of all RunLog Position objects created in the system (that have not been deleted yet). The response is paged (default is 0 offset and 50 limit).

**Request URI:**

/api/v1/positions/<offset>,<limit>

*(Example: /api/v1/positions/0,50)*

**Http Verb:**

GET

**Http Status Codes:**

200 – “OK”

404 – “Not Found”

Example error messages:

Not Found: “Error in GetPositionsAsync() method. Position list cannot be displayed. Error message returned from PositionService: {ErrorMessage}”

**Supported Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Param** | **Value** | **Comments** |
| HEAD | api\_key | string | Must be sent with all client requests. Helps the server to validate the request source. |
| URL\_PARAM | offset | int (nullable) | Number of Position records to skip before retrieving list. Default is 0.  (optional: omit if default values are acceptable) |
| URL\_PARAM | limit | int (nullable) | Number of Position records to take when retrieving list. Default is 50 and max is 50.  (optional: omit if default values are acceptable) |

**Sample Request Body**

GET {{endpoint}}//api/v1/positions/0,50 HTTP/1.1

Host: api.runlog.com

Authorization: Basic AUTH\_TOKEN

Content-Length: 385

**Sample Response Body**

Response will be an object containing an array of Position(s). The response is paged.

[

{

"ActivityId": 1,

"TimeStamp": "2020-07-29T22:54:42.747Z",

"Latitude": 0,

"Longitude": 0,

"Altitude": 0,

"Accuracy": 0,

"AltitudeAccuracy": 0,

"Heading": 0,

"Speed": 0,

"SatelliteCount": 0,

"Id": 1

},

{

"ActivityId": 2,

"TimeStamp": "2020-07-29T22:54:42.747Z",

"Latitude": 0,

"Longitude": 0,

"Altitude": 0,

"Accuracy": 0,

"AltitudeAccuracy": 0,

"Heading": 0,

"Speed": 0,

"SatelliteCount": 0,

"Id": 2

}

]

**Sample Response Headers**

content-length: 385

content-type: application/json; charset=utf-8

date: Tue, 28 Jul 2020 21:26:08 GMT

server: Microsoft-IIS/10.0

status: 200

x-correlation-id: 80000010-0006-ff00-b63f-84710c7967bb

x-powered-by: ASP.NET

# Methods – Get Position (by Id)

**Description**

Retrieve a single RunLog Position object from the system (if found).

**Request URI:**

/api/v1/positions/<positionId>

*(Example: /api/v1/positions/1)*

**Http Verb:**

GET

**Http Status Codes:**

200 – “OK”

404 – “Not Found”

Example error messages:

Not Found: “Error in GetPositionByIdAsync() method. Cannot fetch Position info. Error message returned from PositionService: {ErrorMessage}”

**Supported Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Param** | **Value** | **Comments** |
| HEAD | api\_key | string | Must be sent with all client requests. Helps the server to validate the request source. |
| URL\_PARAM | positionId | int | Id of the Position to retrieve. |

**Sample Request Body**

GET {{endpoint}}//api/v1/positions/1 HTTP/1.1

Host: api.runlog.com

Authorization: Basic AUTH\_TOKEN

Content-Length: 191

**Sample Response Body**

Response will be a Position object (if id is found).

{

"ActivityId": 1,

"TimeStamp": "2020-07-29T22:54:42.747Z",

"Latitude": 0,

"Longitude": 0,

"Altitude": 0,

"Accuracy": 0,

"AltitudeAccuracy": 0,

"Heading": 0,

"Speed": 0,

"SatelliteCount": 0,

"Id": 1

}

**Sample Response Headers**

content-length: 191

content-type: application/json; charset=utf-8

date: Tue, 28 Jul 2020 21:26:08 GMT

server: Microsoft-IIS/10.0

status: 200

x-correlation-id: 80000010-0006-ff00-b63f-84710c7967bb

x-powered-by: ASP.NET

# Methods – Create Position

**Description**

Create a new Position object in the system.

**Request URI:**

api/v1/positions

*(Example:* *api/v1/positions)*

**Http Verb:**

POST

**Http Status Codes:**

201 – “Created”

400 – “Bad Request”

404 – “Not Found”

Example error messages:

Bad Request: “Error in CreatePositionAsync() method. Position object may not have been added to database. Exception was thrown in PositionService. Exception Type: '{Exception.GetType()}'. Exception Message: {Exception.Message}”

Not Found: “Error in CreatePositionAsync() method. Position object may not have been added to database. Error message returned from PositionService: {ErrorMessage}”

**Supported Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Param** | **Value** | **Comments** |
| HEAD | api\_key | string | That was given in response to /api/login. |
| POST | position | Position | POCO with all the Position property values to be used to create a new Position object in the system. |

**Sample Request Body**

POST {{endpoint}}//api/v1/positions HTTP/1.1

Host: api.runlog.com

Authorization: Basic AUTH\_TOKEN

Content-Length: 191

Content-Type: application/json-patch+json; charset=UTF-8

{

"activityId": 1, Note: this id field is important to associate it with an Activity object

"timeStamp": "2020-07-29T22:57:29.811Z",

"latitude": 0,

"longitude": 0,

"altitude": 0,

"accuracy": 0,

"altitudeAccuracy": 0,

"heading": 0,

"speed": 0,

"satelliteCount": 0,

"id": 0 Note: id field ignored on create

}

**Sample Response Body**

Response (status code: 201) will be an object containing the id of the new Position document that was created.

{

"ActivityId": 1,

"TimeStamp": "2020-07-29T22:57:29.811Z",

"Latitude": 0,

"Longitude": 0,

"Altitude": 0,

"Accuracy": 0,

"AltitudeAccuracy": 0,

"Heading": 0,

"Speed": 0,

"SatelliteCount": 0,

"Id": 3

}

**Sample Response Headers**

access-control-allow-origin: \*

content-length: 191

content-type: application/json; charset=utf-8

date: Tue, 28 Jul 2020 21:45:21 GMT

location: https://localhost:44349/api/v1/positions/1

server: Microsoft-IIS/10.0

status: 201

x-correlation-id: 80000006-0004-fd00-b63f-84710c7967bb

x-powered-by: ASP.NET

# Methods – Update Position

**Description**

Update a Position object (all properties – no PATCH).

**Request URI:**

/api/v1/positions/<positionId>

*(Example: /api/v1/positions/1)*

**Http Verb:**

PUT

**Http Status Codes:**

200 – “OK”

400 – “Bad Request”

404 – “Not Found”

Example error messages:

BadRequest: “The submitted id value '{id}' does not match the corresponding field contained in the submitted object Position id value '{position.Id}'.”

NotFound: “Error in UpdatePositionAsync() method. Cannot find Position with id '{id}'. Error message returned from PositionService: {ErrorMessage}”

Not Found: “Error in UpdatePositionAsync() method. Cannot update Position with id '{id}'. Error message returned from PositionService: {ErrorMessage}”

**Supported Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Param** | **Value** | **Comments** |
| HEAD | api\_key | string | Must be sent with all client requests. Helps the server to validate the request source. |
| URL\_PARAM | positionId | int | Id of the Position object to update. |
| PUT | position | Position | POCO with all the Position object properties that can be updated. |

**Sample Request Body**

PUT {{endpoint}}/api/v1/positions/1 HTTP/1.1

Host: api.runlog.com

Authorization: Basic AUTH\_TOKEN

Content-Length: 194

Content-Type: application/json-patch+json; charset=UTF-8

{

"activityId": 1,

"timeStamp": "2020-07-29T23:00:49.116Z",

"latitude": 99,

"longitude": 101,

"altitude": 5,

"accuracy": 0,

"altitudeAccuracy": 0,

"heading": 0,

"speed": 0,

"satelliteCount": 0,

"id": 1 Note: id field in object must match id field in URL\_PARAM

}

**Sample Response Body**

Response will be a Position object containing all the fields..

Example:

{

"ActivityId": 1,

"TimeStamp": "2020-07-29T23:00:49.116Z",

"Latitude": 99,

"Longitude": 101,

"Altitude": 5,

"Accuracy": 0,

"AltitudeAccuracy": 0,

"Heading": 0,

"Speed": 0,

"SatelliteCount": 0,

"Id": 1

}

**Sample Response Headers**

access-control-allow-origin: \*

content-length: 194

content-type: application/json; charset=utf-8

date: Tue, 28 Jul 2020 01:48:13 GMT

server: Microsoft-IIS/10.0

status: 200

x-correlation-id: 80000005-0000-fb00-b63f-84710c7967bb

x-powered-by: ASP.NET

# Methods – Delete Position

**Description**

Delete a Position object from the system.

**Request URI:**

/api/v1/positions/<positionId>

*(Example: /api/v1/positions/1)*

**Http Verb:**

DELETE

**Http Status Codes:**

200 – “OK”

404 – “Resource Not Found”

Example error messages:

“Error in DeletePositionById() method. Position object may not have been deleted. Error message returned from PositionService: {ErrorMessage}”

**Supported Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Param** | **Value** | **Comments** |
| HEAD | api\_key | string | Must be sent with all client requests. Helps the server to validate the request source. |
| URL\_PARAM | positionId | int | Id of the Position object to be deleted. |

**Sample Request Body**

DELETE {{endpoint}}//api/v1/positions/1 HTTP/1.1

Host: api.runlog.com

Authorization: Basic AUTH\_TOKEN

**Sample Response Body**

Response will be a 200 OK containing a response message.

"Delete successful"

Note: "Delete unsuccessful" if nothing was deleted

**Sample Response Header**

access-control-allow-origin: \*

content-encoding: gzip

content-type: text/plain; charset=utf-8

date: Tue, 28 Jul 2020 23:30:45 GMT

server: Microsoft-IIS/10.0

status: 200

vary: Accept-Encoding

x-correlation-id: 80000012-0006-ff00-b63f-84710c7967bb

x-powered-by: ASP.NET

# Glossary

**Conventions**

* Client - Client application.
* Status - HTTP status code of response.
* All the possible responses are listed under ‘Responses’ for each method. Only one of them is issued per request server.
* All responses are in JSON format.
* All request parameters are mandatory unless explicitly marked as [optional]
* The type of values accepted for a request parameter are shown the values column like this [10|<any number>]. The | symbol means OR. If the parameter is [optional], the default value is shown in blue bold text, as 10 is written in [10|<any number>].

**Status Codes**

All status codes are standard HTTP status codes. The below ones are used in this API.

2XX - Success of some kind

4XX - Error occurred in client’s part

5XX - Error occurred in server’s part

|  |  |
| --- | --- |
| **Status Code** | **Description** |
| 200 | OK |
| 201 | Created |
| 202 | Accepted (Request accepted, and queued for execution) |
| 400 | Bad request |
| 401 | Authentication failure |
| 403 | Forbidden |
| 404 | Resource not found |
| 405 | Method Not Allowed |
| 409 | Conflict |
| 412 | Precondition Failed |
| 413 | Request Entity Too Large |
| 422 | Unprocessable Entity |
| 500 | Internal Server Error |
| 501 | Not Implemented |
| 503 | Service Unavailable |