Portfolio Assignment: API Spec

CS493: Cloud Application Development

Colin Francis

Fall 2022

Oregon State University

# Table of Contents

[Table of Contents 1](#_Toc120552095)

[Data Model 2](#_Toc120552096)

[Create an Aircraft 3](#_Toc120552097)

[Get All Aircrafts 7](#_Toc120552098)

[Get an Aircraft 11](#_Toc120552099)

[Partially Update an Aircraft 14](#_Toc120552100)

[Fully Update an Aircraft 19](#_Toc120552101)

[Delete an Aircraft 24](#_Toc120552102)

[Create a Hangar 26](#_Toc120552103)

[Get All Hangars 30](#_Toc120552104)

[Get a Hangar 34](#_Toc120552105)

[Add an Aircraft to a Hangar 36](#_Toc120552106)

[Remove an Aircraft from a Hangar 39](#_Toc120552107)

[Partially Update a Hangar 42](#_Toc120552108)

[Fully Update a Hangar 47](#_Toc120552109)

[Delete a Hangar 52](#_Toc120552110)

# 

# Data Model

The app stores three kinds of entities in Datastore – Users, Aircrafts, Hangars.

## Users

|  |  |  |
| --- | --- | --- |
| **Property** | **Data Type** | **Notes** |
| id | String | The unique id of the user. This is automatically generated for the user when they first sign-in with Google. |
| givenName | String | The given name of the user. |
| familyName | String | The family name of the user. |
| aircrafts | Integer | The number of aircrafts owned by the user. |

## Aircrafts

|  |  |  |
| --- | --- | --- |
| **Property** | **Data Type** | **Notes** |
| id | Integer | The unique id of the aircraft. Datastore automatically generates this when the aircraft is created. |
| make | String | The make of the aircraft, e.g., Boeing, Cessna, Embraer, etc. |
| model | String | The model of the aircraft, e.g., 747, 150, EMB 110, etc. |
| wingspan | Double | The wingspan of the aircraft in meters. |
| hangar | Integer | The id of the hangar that the aircraft is parked in. If the hangar is not parked in a hangar, this value will be null. |
| ownerId | String | The id of the aircraft’s owner. This will be equal to the id of the User entity that owns the aircraft. |

## Hangars

|  |  |  |
| --- | --- | --- |
| **Property** | **Data Type** | **Notes** |
| id | Integer | The unique id of the hangar. Datastore automatically generates this when the hangar is created. |
| name | String | The name of the hangar. |
| city | String | The United States city that the hangar is located in. |
| state | String | The Unites States state that the hangar is located in. |
| capacity | Integer | The maximum number of aircrafts that the hangar can store. |
| aircrafts | Array | An array containing the IDs of the aircrafts currently stored in the hangar. |

# Create an Aircraft

Allows you to create a new aircraft.

|  |
| --- |
| POST /aircrafts |

## Request

### Media Type(s)

application/json

### Authorization

Required

### Authorization Type

Bearer Token

### Path Parameters

None

### Request Body

Required

### Request Body Format

JSON

### Request JSON Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Description** | **Notes** | **Required?** |
| make | The make of the aircraft, e.g., Boeing, Cessna, Embraer, etc. | The minimum permissible string length of an aircraft make is 1 character. The maximum permissible string length for an aircraft make is 255 characters.  All aircraft makes must use characters that match the regular expression: ^[A-Za-z ]+$ | Yes |
| model | The model of the aircraft, e.g., 747, 150, EMB 110, etc. | The minimum permissible string length of an aircraft model is 1 character. The maximum permissible string length for an aircraft model is 255 characters.  All aircraft models must use characters that match the regular expression: ^[A-Za-z0-9 -/]+$ | Yes |
| wingspan | The wingspan of the aircraft in meters. | The minimum permissible length of an aircraft wingspan is 2m. The maximum permissible length of an aircraft wingspan is 150m. | Yes |

### Request Body Example

|  |
| --- |
| Content-Type: application/json  {  “make”: “Boeing”,  “model”: “747”,  “wingspan”: 59.6  } |

## Response

### Media Type(s)

application/json

### Response Body Format

JSON

### Response Statuses

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status Code** | **Notes** |
| Success | 201 Created |  |
| Failure | 401 Unauthorized | If the request is missing a JSON web token or does not contain a valid JSON web token as a Bearer Token the aircraft will not be created and a 401 status code will be returned. |
| Failure | 415 Unsupported Media Type | If the media type of the content provided in the request body is not JSON and the associated Content-Type request header is not application/json, the aircraft will not be created and a 415 status code will be returned. |
| Failure | 406 Not Acceptable | If the Accept header in the request is not application/json, the aircraft will not be created and a 406 status code will be returned.  Note: If the client has an Accept header set to \*/\*, i.e., any MIME type, the server will serve application/json in the response. The server assumes the client is indifferent to a specific media type as one was not explicitly set in the Accept header. |
| Failure | 400 Bad Request | If the request is missing any of the 3 required attributes, the aircraft will not be created and a 400 status code will be returned.  If any of the 3 required attributes in the request are misconfigured, the aircraft will not be created and a 400 status code will be returned. |
| Failure | 500 Internal Server Error | An internal server error has occurred. |

### Response Examples

#### Success

|  |
| --- |
| Status: 201 Created  Content-Type: application/json  {  “id”: 1234,  “make”: “Boeing”,  “model”: “747”,  “wingspan”: 59.6,  “hangar”: null,  “ownerId”: “5678”,  “self”: “https://appspot.com/aircrafts/1234”  } |

#### Failure

|  |
| --- |
| Status Code: 401 Unauthorized  Content-Type: application/json  {  “Error”: “Bearer token is missing or invalid”  } |

#### Failure

|  |
| --- |
| Status: 415 Unsupported Media Type  Content-Type: application/json  {  “Error”: “This endpoint only accepts application/json”  } |

#### Failure

|  |
| --- |
| Status: 406 Not Acceptable  Content-Type: application/json  {  “Error”: “This endpoint only serves application/json”  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  Content-Type: application/json  {  “Error”: “Make attribute is missing or invalid”  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  Content-Type: application/json  {  “Error”: “Model attribute is missing or invalid”  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  Content-Type: application/json  {  “Error”: “Wingspan attribute is missing or invalid”  } |

#### Failure

|  |
| --- |
| Status: 500 Internal Server Error  Content-Type: application/json  {  “Error”: “An internal server error has occurred”  } |

# Get All Aircrafts

Allows you to fetch all aircrafts that you own.

|  |
| --- |
| GET /aircrafts |

## Request

### Media Type(s)

None

### Authorization

Required

### Authorization Type

Bearer Token

### Path Parameters

None

### Request Body

None

## Response

### Media Type(s)

application/json

### Response Body Format

JSON

### Response Statuses

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status Code** | **Notes** |
| Success | 200 OK | The maximum number of aircrafts returned by a single request will be 5.  If there are additional aircrafts that can be retrieved, the JSON object returned in the response body will contain an aircrafts array and a “next” attribute with the attribute value being a cursor allowing you to retrieve more aircrafts via pagination.  A “totalAircrafts” attribute in the response body shows the total number of aircrafts that you own. This attribute also represents the total number of aircrafts that you can retrieve.  When there are no aircrafts left to retrieve, the object returned will only include the aircrafts array and the “totalAircrafts” attribute. |
| Failure | 401 Unauthorized | If the request is missing a JSON web token or does not contain a valid JSON web token as a Bearer Token the request will fail and a 401 status code will be returned. |
| Failure | 406 Not Acceptable | If the Accept header in the request is not application/json, the request will fail and a 406 status code will be returned.  Note: If the client has an Accept header set to \*/\*, i.e., any MIME type, the server will serve application/json in the response. The server assumes the client is indifferent to a specific media type as one was not explicitly set in the Accept header. |
| Failure | 500 Internal Server Error | An internal server error has occurred. |

### Response Examples

#### Success

|  |
| --- |
| Status Code: 200 OK  Content-Type: application/json  {  “totalAircrafts”: 6,  “aircrafts”: [  {  “id”: 1234,  “make”: “Boeing”,  “model”: “747”,  “wingspan”: 59.6,  “hangar”: null,  “ownerId”: “5678”,  “self”: “https://appspot.com/aircrafts/1234”  },  {  “id”: 1357,  “make”: “Tarrant”,  “model”: “Tabor”,  “wingspan”: 40.0,  “hangar”: {  “id”: 8765,  “self”: “https://appspot.com/hangars/8765”  },  “ownerId”: “5678”,  “self”: “https://appspot.com/aircrafts/1357”  },  {  “id”: 2468,  “make”: “Airbus”,  “model”: “Beluga”,  “wingspan”: 44.8,  “hangar”: null,  “ownerId”: “5678”,  “self”: “https://appspot.com/aircrafts/2468”  },  {  “id”: 4690,  “make”: “Convair”,  “model”: “B-36”,  “wingspan”: 70.1,  “hangar”: {  “id”: 5928,  “self”: “https://appspot.com/hangars/5928”  },  “ownerId”: “5678”,  “self”: “https://appspot.com/aircrafts/4690”  },  {  “id”: 9876,  “make”: “Antonov”,  “model”: “An-225”,  “wingspan”: 88.4,  “hangar”: {  “id”: 0629,  “self”: “https://appspot.com/hangars/0629”  },  “ownerId”: “5678”,  “self”: “https://appspot.com/aircrafts/9876”  },  ],  “next”: “https://appspot.com/aircrafts?cursor=CjYSMGoWZH5jczQ5My1maW5hbC1mcmFuY2ljb3IWCxIJQWlyY3JhZnRzGICAgJiH2JAKDBgAIAA%3D”  } |

#### Success

|  |
| --- |
| Status: 200 OK  Content-Type: application/json  {  “totalAircrafts”: 1,  “aircrafts”: [  {  “id”: 1234,  “make”: “Boeing”,  “model”: “747”,  “wingspan”: 59.6,  “hangar”: null,  “ownerId”: “5678”,  “self”: “https://appspot.com/aircrafts/1234”  },  ]  } |

#### Failure

|  |
| --- |
| Status Code: 401 Unauthorized  Content-Type: application/json  {  “Error”: “Bearer token is missing or invalid”  } |

#### Failure

|  |
| --- |
| Status: 406 Not Acceptable  Content-Type: application/json  {  “Error”: “This endpoint only serves application/json”  } |

#### Failure

|  |
| --- |
| Status: 500 Internal Server Error  Content-Type: application/json  {  “Error”: “An internal server error has occurred”  } |

# Get an Aircraft

Allows you to fetch a specific aircraft that you own.

|  |
| --- |
| GET /aircrafts/:aircraftId |

## Request

### Media Type(s)

None

### Authorization

Required

### Authorization Type

Bearer Token

### Path Parameters

|  |  |
| --- | --- |
| **Name** | **Description** |
| aircraftId | The ID of the aircraft |

### Request Body

None

## Response

Media Type(s)

application/json

### Response Body Format

JSON

### Response Statuses

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status Code** | **Notes** |
| Success | 200 OK |  |
| Failure | 401 Unauthorized | If the request is missing a JSON web token or does not contain a valid JSON web token as a Bearer Token the request will fail and a 401 status code will be returned. |
| Failure | 406 Not Acceptable | If the Accept header in the request is not application/json, the request will fail and a 406 status code will be returned.  Note: If the client has an Accept header set to \*/\*, i.e., any MIME type, the server will serve application/json in the response. The server assumes the client is indifferent to a specific media type as one was not explicitly set in the Accept header. |
| Failure | 404 Not Found | No aircraft with this aircraftId exists. |
| Failure | 403 Forbidden | If the user making the request does not own the aircraft they are making the request for, the request will fail and a 403 status code will be returned. |
| Failur | 500 Internal Server Error | An internal server error has occurred. |

### Response Examples

#### Success

|  |
| --- |
| Status: 200 OK  Content-Type: application/json  {  “id”: 1234,  “make”: “Boeing”,  “model”: “747”,  “wingspan”: 59.6,  “hangar”: null,  “ownerId”: “5678”,  “self”: “https://appspot.com/aircrafts/1234”  } |

#### Failure

|  |
| --- |
| Status: 401 Unauthorized  Content-Type: application/json  {  “Error”: “Bearer token is missing or invalid”  } |

#### Failure

|  |
| --- |
| Status: 406 Not Acceptable  Content-Type: application/json  {  “Error”: “This endpoint only serves application/json”  } |

#### Failure

|  |
| --- |
| Status: 404 Not Found  Content-Type: application/json  {  “Error”: “No aircraft with this aircraftId exists”  } |

#### Failure

|  |
| --- |
| Status: 403 Forbidden  Content-Type: application/json  {  “Error”: “You are not authorized to perform this action”  } |

#### Failure

|  |
| --- |
| Status: 500 Internal Server Error  Content-Type: application/json  {  “Error”: “An internal server error has occurred”  } |

# Partially Update an Aircraft

Allows you to partially update the attributes of an already created aircraft.

|  |
| --- |
| PATCH /aircrafts/:aircraftId |

## Request

### Media Type(s)

application/json

### Authorization

Required

### Authorization Type

Bearer Token

### Path Parameters

|  |  |
| --- | --- |
| **Name** | **Description** |
| aircraftId | The ID of the aircraft |

### Request Body

Required

### Request Body Format

JSON

### Request JSON Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Description** | **Notes** | **Required?** |
| make | The make of the aircraft, e.g., Boeing, Cessna, Embraer, etc. | The minimum permissible string length of an aircraft make is 1 character. The maximum permissible string length for an aircraft make is 255 characters.  All aircraft makes must use characters that match the regular expression: ^[A-Za-z ]+$ | No |
| model | The model of the aircraft, e.g., 747, 150, EMB 110, etc. | The minimum permissible string length of an aircraft model is 1 character. The maximum permissible string length for an aircraft model is 255 characters.  All aircraft models must use characters that match the regular expression: ^[A-Za-z0-9 -/]+$ | No |
| wingspan | The wingspan of the aircraft in meters. | The minimum permissible length of an aircraft wingspan is 2m. The maximum permissible length of an aircraft wingspan is 150m. | No |

\*Note: Any additional attributes provided in the request body that aren’t make, model, or wingspan will be ignored.

### Request Body Example

|  |
| --- |
| Content-Type: application/json  {  “model”: “Super Hornet”  } |

## Response

### Media Type(s)

application/json

### Response Body Format

JSON

### Response Statuses

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status Code** | **Notes** |
| Success | 200 OK |  |
| Failure | 401 Unauthorized | If the request is missing a JSON web token or does not contain a valid JSON web token as a Bearer Token the aircraft will not be updated and a 401 status code will be returned. |
| Failure | 415 Unsupported Media Type | If the media type of the content provided in the request body is not JSON and the associated Content-Type request header is not application/json, the aircraft will not be updated and a 415 status code will be returned. |
| Failure | 406 Not Acceptable | If the Accept header in the request is not application/json, the aircraft will not be updated and a 406 status code will be returned.  Note: If the client has an Accept header set to \*/\*, i.e., any MIME type, the server will serve application/json in the response. The server assumes the client is indifferent to a specific media type as one was not explicitly set in the Accept header. |
| Failure | 400 Bad Request | If any of the 3 possible attributes in the request are provided but misconfigured, the aircraft will not be updated and a 400 status code will be returned. |
| Failure | 404 Not Found | No aircraft with this aircraftId exists. |
| Failure | 403 Forbidden | If the user making the request does not own the aircraft they are trying to update, the request will fail and a 403 status code will be returned. |
| Failure | 500 Internal Server Error | An internal server error has occurred. |

### Response Examples

#### Success

|  |
| --- |
| Status: 200 OK  Content-Type: application/json  {  “id”: 1234,  “make”: “Boeing”,  “model”: “Super Hornet”,  “wingspan”: 59.6,  “hangar”: null,  “ownerId”: “5678”,  “self”: “https://appspot.com/aircrafts/1234”  } |

#### Failure

|  |
| --- |
| Status Code: 401 Unauthorized  Content-Type: application/json  {  “Error”: “Bearer token is missing or invalid”  } |

#### Failure

|  |
| --- |
| Status: 415 Unsupported Media Type  Content-Type: application/json  {  “Error”: “This endpoint only accepts application/json”  } |

#### Failure

|  |
| --- |
| Status: 406 Not Acceptable  Content-Type: application/json  {  “Error”: “This endpoint only serves application/json”  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  Content-Type: application/json  {  “Error”: “Make attribute is invalid”  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  Content-Type: application/json  {  “Error”: “Model attribute is invalid”  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  Content-Type: application/json  {  “Error”: “Wingspan attribute is invalid”  } |

#### Failure

|  |
| --- |
| Status: 404 Not Found  Content-Type: application/json  {  “Error”: “No aircraft with this aircraftId exists”  } |

#### Failure

|  |
| --- |
| Status: 403 Forbidden  Content-Type: application/json  {  “Error”: “You are not authorized to perform this action”  } |

#### Failure

|  |
| --- |
| Status: 500 Internal Server Error  Content-Type: application/json  {  “Error”: “An internal server error has occurred”  } |

# 

# Fully Update an Aircraft

Allows you to update all attributes of an aircraft that you own.

|  |
| --- |
| PUT /aircrafts/:aircraftId |

## Request

### Media Type(s)

application/json

### Authorization

Required

### Authorization Type

Bearer Token

### Path Parameters

|  |  |
| --- | --- |
| **Name** | **Description** |
| aircraftId | The ID of the aircraft |

### Request Body

Required

### Request Body Format

JSON

### Request JSON Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Description** | **Notes** | **Required?** |
| make | The make of the aircraft, e.g., Boeing, Cessna, Embraer, etc. | The minimum permissible string length of an aircraft make is 1 character. The maximum permissible string length for an aircraft make is 255 characters.  All aircraft makes must use characters that match the regular expression: ^[A-Za-z ]+$ | Yes |
| model | The model of the aircraft, e.g., 747, 150, EMB 110, etc. | The minimum permissible string length of an aircraft model is 1 character. The maximum permissible string length for an aircraft model is 255 characters.  All aircraft models must use characters that match the regular expression: ^[A-Za-z0-9 -/]+$ | Yes |
| wingspan | The wingspan of the aircraft in meters. | The minimum permissible length of an aircraft wingspan is 2m. The maximum permissible length of an aircraft wingspan is 150m. | Yes |

\*Note: Any additional attributes provided in the request body that aren’t make, model, or wingspan will be ignored.

### Request Body Example

|  |
| --- |
| Content-Type: application/json  {  “make”: “McDonnell”,  “model”: “F-4 Phantom II”,  “wingspan”: 11.7  } |

## Response

### Media Type(s)

application/json

### Response Body Format

JSON

### Response Statuses

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status Code** | **Notes** |
| Success | 200 OK |  |
| Failure | 401 Unauthorized | If the request is missing a JSON web token or does not contain a valid JSON web token as a Bearer Token the aircraft will not be updated and a 401 status code will be returned. |
| Failure | 415 Unsupported Media Type | If the media type of the content provided in the request body is not JSON and the associated Content-Type request header is not application/json, the aircraft will not be updated and a 415 status code will be returned. |
| Failure | 406 Not Acceptable | If the Accept header in the request is not application/json, the aircraft will not be updated and a 406 status code will be returned.  Note: If the client has an Accept header set to \*/\*, i.e., any MIME type, the server will serve application/json in the response. The server assumes the client is indifferent to a specific media type as one was not explicitly set in the Accept header. |
| Failure | 400 Bad Request | If the request is missing any of the 3 required attributes, the aircraft will not be updated and a 400 status code will be returned.  If any of the 3 required attributes in the request are misconfigured, the aircraft will not be updated and a 400 status code will be returned. |
| Failure | 404 Not Found | No aircraft with this aircraftId exists. |
| Failure | 403 Forbidden | If the user making the request does not own the aircraft they are trying to update, the request will fail and a 403 status code will be returned. |
| Failure | 500 Internal Server Error | An internal server error has occurred. |

### Response Examples

#### Success

|  |
| --- |
| Status: 200 OK  Content-Type: application/json  {  “id”: 1234,  “make”: “McDonnell”,  “model”: “F-4 Phantom II”,  “wingspan”: 11.7,  “hangar”: null,  “ownerId”: “5678”,  “self”: “https://appspot.com/aircrafts/1234”  } |

#### Failure

|  |
| --- |
| Status Code: 401 Unauthorized  Content-Type: application/json  {  “Error”: “Bearer token is missing or invalid”  } |

#### Failure

|  |
| --- |
| Status: 415 Unsupported Media Type  Content-Type: application/json  {  “Error”: “This endpoint only accepts application/json”  } |

#### Failure

|  |
| --- |
| Status: 406 Not Acceptable  Content-Type: application/json  {  “Error”: “This endpoint only serves application/json”  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  Content-Type: application/json  {  “Error”: “Make attribute is missing or invalid”  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  Content-Type: application/json  {  “Error”: “Model attribute is missing or invalid”  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  Content-Type: application/json  {  “Error”: “Wingspan attribute is missing or invalid”  } |

#### Failure

|  |
| --- |
| Status: 404 Not Found  Content-Type: application/json  {  “Error”: “No aircraft with this aircraftId exists”  } |

#### Failure

|  |
| --- |
| Status: 403 Forbidden  Content-Type: application/json  {  “Error”: “You are not authorized to perform this action”  } |

#### Failure

|  |
| --- |
| Status: 500 Internal Server Error  Content-Type: application/json  {  “Error”: “An internal server error has occurred”  } |

# Delete an Aircraft

Allows you to delete an aircraft that you own. If the aircraft was parked in a hangar, deleting the aircraft will automatically remove it from the hangar.

|  |
| --- |
| DELETE /aircrafts/:aircraftId |

## Request

### Media Type(s)

None

### Authorization

Required

### Authorization Type

Bearer Token

### Path Parameters

|  |  |
| --- | --- |
| **Name** | **Description** |
| aircraftId | The ID of the aircraft |

### Request Body

None

## Response

### Media Type(s)

Success: None

Failure: application/json

### Response Body Format

Success: No Body

Failure: JSON

### Response Statuses

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status Code** | **Notes** |
| Success | 204 No Content |  |
| Failure | 401 Unauthorized | If the request is missing a JSON web token or does not contain a valid JSON web token as a Bearer Token the request will fail and a 401 status code will be returned. |
| Failure | 404 Not Found | No aircraft with this aircraftId exists. |
| Failure | 403 Forbidden | If the user making the request does not own the aircraft they are trying to delete, the request will fail and a 403 status code will be returned. |
| Failure | 500 Internal Server Error | An internal server error has occurred. |

### Response Examples

#### Success

|  |
| --- |
| Status: 204 No Content |

#### Failure

|  |
| --- |
| Status: 401 Unauthorized  Content-Type: application/json  {  “Error”: “Bearer token is missing or invalid”  } |

#### Failure

|  |
| --- |
| Status: 404 Not Found  Content-Type: application/json  {  “Error”: “No aircraft with this aircraftId exists”  } |

#### Failure

|  |
| --- |
| Status: 403 Forbidden  Content-Type: application/json  {  “Error”: “You are not authorized to perform this action”  } |

#### Failure

|  |
| --- |
| Status: 500 Internal Server Error  Content-Type: application/json  {  “Error”: “An internal server error has occurred”  } |

# Create a Hangar

Allows you to create a new hangar.

|  |
| --- |
| POST /hangars |

## Request

### Media Type(s)

application/json

### Authorization

None

### Path Parameters

None

### Request Body

Required

### Request Body Format

JSON

### Request JSON Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Description** | **Notes** | **Required?** |
| name | The name of the hangar. | The minimum permissible string length of a hangar name is 1 character. The maximum permissible string length for a hangar name is 255 characters.  All hangar name must use characters that match the regular expression: ^[A-Za-z0-9 ./,-]+$ | Yes |
| city | The United States city that the hangar is located in. | The minimum permissible string length of a hangar city is 1 character. The maximum permissible string length for a hangar city is 255 characters.  All hangar cities must use characters that match the regular expression: ^[A-Za-z .,]+$ | Yes |
| state | The United States state that the hangar is located in. | The minimum permissible string length of a hangar state is 1 character. The maximum permissible string length for a hangar state is 255 characters.  All hangar states must use characters that match the regular expression: ^[A-Za-z ]+$ | Yes |
| capacity | The maximum number of aircrafts that the hangar can hold. | The smallest permissible capacity value is 1. The largest permissible capacity value is 5. | Yes |

### Request Body Example

|  |
| --- |
| Content-Type: application/json  {  “name”: “Arch Hangar”,  “city”: “Limestone”,  “state”: “Maine”,  “capacity”: 2  } |

## Response

### Media Type(s)

application/json

### Response Body Format

JSON

### Response Statuses

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status Code** | **Notes** |
| Success | 201 Created |  |
| Failure | 415 Unsupported Media Type | If the media type of the content provided in the request body is not JSON and the associated Content-Type request header is not application/json, the hangar will not be created and a 415 status code will be returned. |
| Failure | 406 Not Acceptable | If the Accept header in the request is not application/json, the hangar will not be created and a 406 status code will be returned.  Note: If the client has an Accept header set to \*/\*, i.e., any MIME type, the server will serve application/json in the response. The server assumes the client is indifferent to a specific media type as one was not explicitly set in the Accept header. |
| Failure | 400 Bad Request | If the request is missing any of the 4 required attributes, the hangar will not be created and a 400 status code will be returned.  If any of the 4 required attributes in the request are misconfigured, the hangar will not be created and a 400 status code will be returned. |
| Failure | 500 Internal Server Error | An internal server error has occurred. |

### Response Examples

#### Success

|  |
| --- |
| Status: 201 Created  Content-Type: application/json  {  “id”: 1234,  “name”: “Arch Hangar”,  “city”: “Limestone”,  “state”: “Maine”,  “capacity”: 2,  “aircrafts”: [],  “self”: “https://appspot.com/hangars/1234”  } |

#### Failure

|  |
| --- |
| Status: 415 Unsupported Media Type  Content-Type: application/json  {  “Error”: “This endpoint only accepts application/json”  } |

#### Failure

|  |
| --- |
| Status: 406 Not Acceptable  Content-Type: application/json  {  “Error”: “This endpoint only serves application/json”  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  Content-Type: application/json  {  “Error”: “Name attribute is missing or invalid”  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  Content-Type: application/json  {  “Error”: “City attribute is missing or invalid”  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  Content-Type: application/json  {  “Error”: “State attribute is missing or invalid”  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  Content-Type: application/json  {  “Error”: “Capacity attribute is missing or invalid”  } |

#### Failure

|  |
| --- |
| Status: 500 Internal Server Error  Content-Type: application/json  {  “Error”: “An internal server error has occurred”  } |

# Get All Hangars

Allows you to fetch all hangars.

|  |
| --- |
| GET /hangars |

## Request

### Media Type(s)

None

### Authorization

None

### Path Parameters

None

### Request Body

None

## Response

### Media Type(s)

application/json

### Response Body Format

JSON

### Response Statuses

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status Code** | **Notes** |
| Success | 200 OK | The maximum number of hangars returned will be 5.  If there are additional hangars that can be retrieved, the JSON object returned in the response body will contain a hangars array and a “next” attribute allowing you to retrieve more hangars via pagination. Additionally, a “totalHangars” attribute in the response body shows the total number of hangars.  When there are no hangars left to retrieve, the object returned will only include the hangars array and the “totalHangars” attribute. |
| Failure | 406 Not Acceptable | If the Accept header in the request is not application/json, the request will fail and a 406 status code will be returned.  Note: If the client has an Accept header set to \*/\*, i.e., any MIME type, the server will serve application/json in the response. The server assumes the client is indifferent to a specific media type as one was not explicitly set in the Accept header. |
| Failure | 500 Internal Server Error | An internal server error has occurred. |

### Response Examples

#### Success

|  |
| --- |
| Status Code: 200 OK  Content-Type: application/json  {  “totalHangars”: 7,  “hangars”: [  {  “id”: 1234,  “name”: “Arch Hangar”,  “city”: “Limestone”,  “state”: “Maine”,  “capacity”: 2,  “aircrafts”: [  { “id”: 8765, “self”: “https://appspot.com/aircrafts/8765” },  { “id”: 4321, “self”: “https://appspot.com/aircrafts/4321” }  ],  “self”: “https://appspot.com/hangars/1234”  },  {  “id”: 5678,  “name”: “Riverside Hangar”,  “city”: “St. Paul”,  “state”: “Minnesota”,  “capacity”: 1,  “aircrafts”: [],  “self”: “https://appspot.com/hangars/5678”  },  {  “id”: 9012,  “name”: “Hangar 1”,  “city”: “Los Angeles”,  “state”: “California”,  “capacity”: 5,  “aircrafts”: [  { “id”: 0987, “self”: “https://appspot.com/aircrafts/0987” },  { “id”: 6543, “self”: “https://appspot.com/aircrafts/0987” },  { “id”: 2109, “self”: “https://appspot.com/aircrafts/2109” }  ],  “self”: “https://appspot.com/hangars/9012”  },  {  “id”: 3456,  “name”: “Wingfoot Lake Airship Hangar”,  “city”: “Suffield”,  “state”: “Ohio”,  “capacity”: 3,  “aircrafts”: [],  “self”: “https://appspot.com/hangars/3456”  },  {  ““id”: 7890,  “name”: “Gulfport Army Airfield Hangar”,  “city”: “Gulfport”,  “state”: “Mississippi”,  “capacity”: 4,  “aircrafts”: [],  “self”: “https://appspot.com/hangars/7890”  },  ],  “next”: “https://appspot.com/hangars?cursor=CjYSMGoWZH5jczQ5My1maW5hbC1mcmFuY2ljb3IWCxIJQWlyY3JhZnRzGICAgJiH2JAKDBgAIAA%3D”  } |

#### Success

|  |
| --- |
| Status: 200 OK  Content-Type: application/json  {  “totalHangars”: 2,  “hangars”: [  {  “id”: 1234,  “name”: “Arch Hangar”,  “city”: “Limestone”,  “state”: “Maine”,  “capacity”: 2,  “aircrafts”: [  { “id”: 8765, “self”: “https://appspot.com/aircrafts/8765” },  { “id”: 4321, “self”: “https://appspot.com/aircrafts/4321” }  ],  “self”: “https://appspot.com/hangars/1234”  },  {  “id”: 5678,  “name”: “Riverside Hangar”,  “city”: “St. Paul”,  “state”: “Minnesota”,  “capacity”: 1,  “aircrafts”: [],  “self”: “https://appspot.com/hangars/5678”  },  ]  } |

#### Failure

|  |
| --- |
| Status: 406 Not Acceptable  Content-Type: application/json  {  “Error”: “This endpoint only serves application/json”  } |

#### Failure

|  |
| --- |
| Status: 500 Internal Server Error  Content-Type: application/json  {  “Error”: “An internal server error has occurred”  } |

# Get a Hangar

Allows you to fetch a specific hangar.

|  |
| --- |
| GET /hangars/:hangarId |

## Request

### Media Type(s)

None

### Authorization

None

### Path Parameters

|  |  |
| --- | --- |
| **Name** | **Description** |
| hangarId | The ID of the hangar |

### Request Body

None

## Response

Media Type(s)

application/json

### Response Body Format

JSON

### Response Statuses

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status Code** | **Notes** |
| Success | 200 OK |  |
| Failure | 406 Not Acceptable | If the Accept header in the request is not application/json, the request will fail and a 406 status code will be returned.  Note: If the client has an Accept header set to \*/\*, i.e., any MIME type, the server will serve application/json in the response. The server assumes the client is indifferent to a specific media type as one was not explicitly set in the Accept header. |
| Failure | 404 Not Found | No hangar with this hangarId exists. |
| Failure | 500 Internal Server Error | An internal server error has occurred. |

### Response Examples

#### Success

|  |
| --- |
| Status: 200 OK  Content-Type: application/json  {  “id”: 1234,  “name”: “Arch Hangar”,  “city”: “Limestone”,  “state”: “Maine”,  “capacity”: 2,  “aircrafts”: [],  “self”: “https://appspot.com/hangars/1234”  } |

#### Failure

|  |
| --- |
| Status: 406 Not Acceptable  Content-Type: application/json  {  “Error”: “This endpoint only serves application/json”  } |

#### Failure

|  |
| --- |
| Status: 404 Not Found  Content-Type: application/json  {  “Error”: “No hangar with this hangarId exists”  } |

#### Failure

|  |
| --- |
| Status: 500 Internal Server Error  Content-Type: application/json  {  “Error”: “An internal server error has occurred”  } |

# Add an Aircraft to a Hangar

Allows you to add an aircraft that you own to a hangar.

|  |
| --- |
| PUT /hangars/:hangarId/aircrafts/:aircraftId |

## Request

### Media Type(s)

None

### Authorization

Required

### Authorization Type

Bearer Token

### Path Parameters

|  |  |
| --- | --- |
| **Name** | **Description** |
| hangarId | The ID of the hangar |
| aircraftId | The ID of the aircraft |

### Request Body

None

## Response

### Media Type(s)

application/json

### Response Body Format

Success: No Body

Failure: JSON

### Response Statuses

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status Code** | **Description** |
| Success | 204 No Content |  |
| Failure | 401 Unauthorized | If the request is missing a JSON web token or does not contain a valid JSON web token as a Bearer Token the request will fail and a 401 status code will be returned. |
| Failure | 403 Forbidden | If the user making the request does not own the aircraft that they are trying to add to a hangar, the request will fail and a 403 status will be returned. |
| Failure | 400 Bad Request | If the hangar is at capacity or if the aircraft is already parked in another hangar, then the request will fail and a 400 status will be returned. |
| Failure | 404 Not Found | No hangar with this hangarId exists or no aircraft with this aircraftId exists. |
| Failure | 500 Internal Server Error | An internal server error has occurred. |

### Response Examples

#### Success

|  |
| --- |
| Status: 204 No Content |

#### Failure

|  |
| --- |
| Status: 401 Unauthorized  Content-Type: application/json  {  “Error”: “Bearer token is missing or invalid”  } |

#### Failure

|  |
| --- |
| Status: 404 Not Found  Content-Type: application/json  {  “Error”: “No hangar with this hangarId exists”  } |

#### Failure

|  |
| --- |
| Status: 404 Not found  Content-Type: application/json  {  “Error”: “No aircraft with this aircraftId exists”  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  Content-Type: application/json  {  “Error”: “This hangar is at capacity”  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  Content-Type: application/json  {  “Error”: “This aircraft is already parked in a hangar”  } |

#### Failure

|  |
| --- |
| Status: 403 Forbidden  Content-Type: application/json  {  “Error”: “You are not authorized to perform this action”  } |

#### Failure

|  |
| --- |
| Status: 500 Internal Server Error  Content-Type: application/json  {  “Error”: “An internal server error has occurred”  } |

# Remove an Aircraft from a Hangar

Allows you to remove an aircraft that you own from a hangar.

|  |
| --- |
| DELETE /hangars/:hangarId/aircrafts/:aircraftId |

## Request

### Media Type(s)

None

### Authorization

Required

### Authorization Type

Bearer Token

### Path Parameters

|  |  |
| --- | --- |
| **Name** | **Description** |
| hangarId | The ID of the hangar |
| aircraftId | The ID of the aircraft |

### Request Body

None

## Response

### Media Type(s)

application/json

### Response Body Format

Success: No Body

Failure: JSON

### Response Statuses

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status Code** | **Description** |
| Success | 204 No Content |  |
| Failure | 401 Unauthorized | If the request is missing a JSON web token or does not contain a valid JSON web token as a Bearer Token the request will fail and a 401 status code will be returned. |
| Failure | 403 Forbidden | If the user making the request does not own the aircraft that they are trying to add to a hangar, then the request will fail and a 403 status will be returned. |
| Failure | 400 Bad Request | If the hangar is empty or if the aircraft is parked in a different hangar, then the request will fail and a 400 status will be returned. |
| Failure | 404 Not Found | No hangar with this hangarId exists or no aircraft with this aircraftId exists. |
| Failure | 500 Internal Server Error | An internal server error has occurred. |

### Response Examples

#### Success

|  |
| --- |
| Status: 204 No Content |

#### Failure

|  |
| --- |
| Status: 401 Unauthorized  Content-Type: application/json  {  “Error”: “Bearer token is missing or invalid”  } |

#### Failure

|  |
| --- |
| Status: 404 Not Found  Content-Type: application/json  {  “Error”: “No hangar with this hangarId exists”  } |

#### Failure

|  |
| --- |
| Status: 404 Not found  Content-Type: application/json  {  “Error”: “No aircraft with this aircraftId exists”  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  Content-Type: application/json  {  “Error”: “This hangar is empty”  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  Content-Type: application/json  {  “Error”: “This aircraft is not parked at this hangar”  } |

#### Failure

|  |
| --- |
| Status: 403 Forbidden  Content-Type: application/json  {  “Error”: “You are not authorized to perform this action”  } |

#### Failure

|  |
| --- |
| Status: 500 Internal Server Error  Content-Type: application/json  {  “Error”: “An internal server error has occurred”  } |

# 

# Partially Update a Hangar

Allows you to partially update the attributes of an already created hangar.

|  |
| --- |
| PATCH /hangars/:hangarId |

## Request

### Media Type(s)

application/json

### Authorization

None

### Path Parameters

|  |  |
| --- | --- |
| **Name** | **Description** |
| hangarId | The ID of the hangar |

### Request Body

Required

### Request Body Format

JSON

### Request JSON Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Description** | **Notes** | **Required?** |
| name | The name of the hangar. | The minimum permissible string length of a hangar name is 1 character. The maximum permissible string length for a hangar name is 255 characters.  All hangar name must use characters that match the regular expression: ^[A-Za-z0-9 ./,-]+$ | No |
| city | The United States city that the hangar is located in. | The minimum permissible string length of a hangar city is 1 character. The maximum permissible string length for a hangar city is 255 characters.  All hangar cities must use characters that match the regular expression: ^[A-Za-z .,]+$ | No |
| state | The United States state that the hangar is located in. | The minimum permissible string length of a hangar state is 1 character. The maximum permissible string length for a hangar state is 255 characters.  All hangar states must use characters that match the regular expression: ^[A-Za-z ]+$ | No |
| capacity | The maximum number of aircrafts that the hangar can hold. | The smallest permissible capacity value is 1. The largest permissible capacity value is 5. | No |

\*Note: Any additional attributes provided in the request body that aren’t name, city, state, or capacity will be ignored.

### Request Body Example

|  |
| --- |
| Content-Type: application/json  {  “model”: “The Arch Hangar”,  “capacity”: 3  } |

## Response

### Media Type(s)

application/json

### Response Body Format

JSON

### Response Statuses

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status Code** | **Notes** |
| Success | 200 OK |  |
| Failure | 415 Unsupported Media Type | If the media type of the content provided in the request body is not JSON and the associated Content-Type request header is not application/json, the hangar will not be updated and a 415 status code will be returned. |
| Failure | 406 Not Acceptable | If the Accept header in the request is not application/json, the hangar will not be updated and a 406 status code will be returned.  Note: If the client has an Accept header set to \*/\*, i.e., any MIME type, the server will serve application/json in the response. The server assumes the client is indifferent to a specific media type as one was not explicitly set in the Accept header. |
| Failure | 400 Bad Request | If any of the 3 possible attributes in the request are misconfigured, the hangar will not be updated and a 400 status code will be returned.  An attempt to update the capacity to a value less than the current number of aircrafts in a hangar will fail with a 400 status code. |
| Failure | 404 Not Found | No hangar with this hangarId exists. |
| Failure | 500 Internal Server Error | An internal server error has occurred. |

### Response Examples

#### Success

|  |
| --- |
| Status: 200 OK  Content-Type: application/json  {  “id”: 1234,  “name”: “The Arch Hangar”,  “city”: “Limestone”,  “state”: “Maine”,  “capacity”: 3,  “aircrafts”: [],  “self”: “https://appspot.com/hangars/1234”  } |

#### Failure

|  |
| --- |
| Status: 415 Unsupported Media Type  Content-Type: application/json  {  “Error”: “This endpoint only accepts application/json”  } |

#### Failure

|  |
| --- |
| Status: 406 Not Acceptable  Content-Type: application/json  {  “Error”: “This endpoint only serves application/json”  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  Content-Type: application/json  {  “Error”: “Name attribute is invalid”  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  Content-Type: application/json  {  “Error”: “City attribute is invalid”  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  Content-Type: application/json  {  “Error”: “State attribute is invalid”  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  Content-Type: application/json  {  “Error”: “Capacity attribute is invalid”  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  Content-Type: application/json  {  “Error”: “Hangar capacity cannot be less than the current number of aircrafts in the hangar”  } |

#### Failure

|  |
| --- |
| Status: 404 Not Found  Content-Type: application/json  {  “Error”: “No hangar with this hangarId exists”  } |

#### Failure

|  |
| --- |
| Status: 500 Internal Server Error  Content-Type: application/json  {  “Error”: “An internal server error has occurred”  } |

# Fully Update a Hangar

Allows you to fully update the attributes of an already created hangar.

|  |
| --- |
| PUT /hangars/:hangarId |

## Request

### Media Type(s)

application/json

### Authorization

None

### Path Parameters

|  |  |
| --- | --- |
| **Name** | **Description** |
| hangarId | The ID of the hangar |

### Request Body

Required

### Request Body Format

JSON

### Request JSON Attributes

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Description** | **Notes** | **Required?** |
| name | The name of the hangar. | The minimum permissible string length of a hangar name is 1 character. The maximum permissible string length for a hangar name is 255 characters.  All hangar name must use characters that match the regular expression: ^[A-Za-z0-9 ./,-]+$ | Yes |
| city | The United States city that the hangar is located in. | The minimum permissible string length of a hangar city is 1 character. The maximum permissible string length for a hangar city is 255 characters.  All hangar cities must use characters that match the regular expression: ^[A-Za-z .,]+$ | Yes |
| state | The United States state that the hangar is located in. | The minimum permissible string length of a hangar state is 1 character. The maximum permissible string length for a hangar state is 255 characters.  All hangar states must use characters that match the regular expression: ^[A-Za-z ]+$ | Yes |
| capacity | The maximum number of aircrafts that the hangar can hold. | The smallest permissible capacity value is 1. The largest permissible capacity value is 5. | Yes |

\*Note: Any additional attributes provided in the request body that aren’t name, city, state, or capacity will be ignored.

### Request Body Example

|  |
| --- |
| Content-Type: application/json  {  “model”: “The Arch Hangar”,  “city”: “Limestone”,  “state”: “Maine”  “capacity”: 2  } |

## Response

### Media Type(s)

application/json

### Response Body Format

JSON

### Response Statuses

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status Code** | **Notes** |
| Success | 200 OK |  |
| Failure | 415 Unsupported Media Type | If the media type of the content provided in the request body is not JSON and the associated Content-Type request header is not application/json, the hangar will not be updated and a 415 status code will be returned. |
| Failure | 406 Not Acceptable | If the Accept header in the request is not application/json, the hangar will not be updated and a 406 status code will be returned.  Note: If the client has an Accept header set to \*/\*, i.e., any MIME type, the serve will server application/json in the response. The server assumes the client is impartial to a specific media type as one was not explicitly set in the Accept header. |
| Failure | 400 Bad Request | If any of the 3 required attributes in the request are missing or misconfigured, the hangar will not be updated and a 400 status code will be returned.  An attempt to update the capacity to a value less than the current number of aircrafts in a hangar will fail with a 400 status code. |
| Failure | 404 Not Found | No hangar with this hangarId exists. |

### Response Examples

#### Success

|  |
| --- |
| Status: 200 OK  Content-Type: application/json  {  “id”: 1234,  “name”: “The Arch Hangar”,  “city”: “Limestone”,  “state”: “Maine”,  “capacity”: 2,  “aircrafts”: [],  “self”: “https://appspot.com/hangars/1234”  } |

#### Failure

|  |
| --- |
| Status: 415 Unsupported Media Type  Content-Type: application/json  {  “Error”: “This endpoint only accepts application/json”  } |

#### Failure

|  |
| --- |
| Status: 406 Not Acceptable  Content-Type: application/json  {  “Error”: “This endpoint only serves application/json”  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  Content-Type: application/json  {  “Error”: “Name attribute is missing or invalid”  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  Content-Type: application/json  {  “Error”: “City attribute is missing or invalid”  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  Content-Type: application/json  {  “Error”: “State attribute is missing or invalid”  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  Content-Type: application/json  {  “Error”: “Capacity attribute is missing or invalid”  } |

#### Failure

|  |
| --- |
| Status: 400 Bad Request  Content-Type: application/json  {  “Error”: “Hangar capacity cannot be less than the current number of aircrafts in the hangar”  } |

#### Failure

|  |
| --- |
| Status: 404 Not Found  Content-Type: application/json  {  “Error”: “No hangar with this hangarId exists”  } |

# Delete a Hangar

Allows you to delete a Hangar. All aircrafts parked in the hangar will be automatically marked as being in flight when the hangar is deleted.

|  |
| --- |
| DELETE /hangars/:hangarId |

## Request

### Media Type(s)

None

### Path Parameters

|  |  |
| --- | --- |
| **Name** | **Description** |
| hangarId | ID of the hangar |

### Request Body

None

## Response

### Media Type(s)

Success: None

Failure: application/json

### Response Body Format

Success: No Body

Failure: JSON

### Response Statuses

|  |  |  |
| --- | --- | --- |
| **Outcome** | **Status Code** | **Notes** |
| Success | 204 No Content |  |
| Failure | 404 Not Found | No hangar with this hangarId exists. |

### Response Examples

#### Success

|  |
| --- |
| Status: 204 No Content |

#### Failure

|  |
| --- |
| Status: 404 Not Found  Content-Type: application/json  {  “Error”: “No hangar with this hangarId exists”  } |