Portfolio Assignment: API Spec

CS493: Cloud Application Development

Colin Francis

Fall 2022

Oregon State University

# 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

|  |  |  |
| --- | --- | --- |
| **Property** | **Data Type** | **Notes** |
| id | Integer | The unique id of the aircraft. Datastore automatically generated 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. |

# 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\s]+$ | 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 user characters that match the regular expression: ^[A-Za-z\s0-9-/]+$ | 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 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 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 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. |

### 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”  } |

# 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 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 allowing you to retrieve more aircrafts via pagination.  When there are no aircrafts left to retrieve, the object returned will only include the aircrafts array. |
| Failure | 401 Unauthorized | If the request is missing 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 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. |

### Response Examples

#### Success

|  |
| --- |
| Status Code: 200 OK  Content-Type: application/json  {  “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  {  “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”  } |

# 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 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 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 | 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. |

### 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”  } |

# 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 | 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 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. |

### 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”  } |