**User Auth**

**/auth/register (POST)**

This route will attempt to create a user entry in the database with the given information, checking to ensure that the data in each entry is valid.

RETURN: TIER (INT)

(body)

{

"email": "Fiona@gmail.com",

"password": "Capybaraisthebest",

"name": "Fiona"

}

#### **Responses**

|  |  |  |
| --- | --- | --- |
| Code | Description | Example |
| 200 | OK | {  "userName": "Fiona",  "userID": 1002741,  “creditPoint”: 100,  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImhheWRlbkB1bnN3LmVkdS5hdSIsImlhdCI6MTYwMzk0MzIzMH 0.b37PfwlcH\_cue6yhgvDt2IiNvhRACf79hTNtacYB94Q"  } |
| 400 | Bad Input | {  "error": "Invalid input"  } |

**/auth/login (POST)**

This route will check that the given login information is valid, and if so will return an authentication token, giving access to user exclusive features.

(body)

{

"email": "Fiona@gmail.com",

"password": "Capybaraisthebest"

}

#### **Responses**

|  |  |  |
| --- | --- | --- |
| Code | Description | Example |
| 200 | OK | {  "userName": "Fiona",  "userID": 1002741,  “creditPoint”: 100,  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImhheWRlbkB1bnN3LmVkdS5hdSIsImlhdCI6MTYwMzk0MzIzMH0.b37PfwlcH\_cue6yhgvDt2IiNvhRACf79hTNtacYB94Q",  'cardName': Capy,  'cardNumber': 1235453,  'expiryDate': 12/11,  'cvv': 123,  } |
| 400 | Bad Input | {  "error": "Invalid input"  } |

**/auth/logout (POST)**

This route will log out a user, removing the token from local storage.

(header)

{

"Authorization": "zIzMH0.b37PfwlcH\_cue6yhgvDt2IiNvhRACf79hTNtacYB94Q"

}

#### **Responses**

|  |  |  |
| --- | --- | --- |
| Code | Description | Example |
| 200 | OK | {} |
| 403 | Forbidden | {  "error": "Invalid Token"  } |

Event

**/event/create (POST)**

This route will create a new event entry to be stored in the database, taking in the fields as specified below:

(header)

{

"Authorization": "zIzMH0.b37PfwlcH\_cue6yhgvDt2IiNvhRACf79hTNtacYB94Q"

}

(body)

{

"title": "Capybara 🦦",

"organiserName": "Fiona",

"organiserId": 1000,

"eventType": "Food",

"location": "UNSW",

"startTimeandDate": "2042/09/02 02:30:00",

"endTimeandDate": "2042/10/02 12:30:0",

"banner": "",

"description": "Learn how to train capybara, cause Capy >> llama.",

"totalTicketQuantity": 4,

"ticketPrice": 20  
}

#### **Responses**

|  |  |  |
| --- | --- | --- |
| Code | Description | Example |
| 200 | OK | {  “eventID”: 10345  } |
| 400 | Bad Input | {  "error": "Invalid input"  } |
| 403 | Forbidden | {  "error": "Invalid Token"  } |

**/event/getall (GET)**

This route will return the data of events in the database system.

No param/header/body

#### **Responses**

|  |  |  |
| --- | --- | --- |
| Code | Description | Example |
| 200 | OK | {  "events": [  {  "id": 56513315,  "title": "Capybara 🦦",  "address": "UNSW",  "startTimeandDate": "2042/09/02 02:30:00",  "endTimeandDate": "2042/10/02 12:30:0",  "banner": ""  }  ]  } |

**/event/{eventID} (GET)** getting all details for an event, to display on event page

(path): eventID

#### **Responses**

|  |  |  |
| --- | --- | --- |
| Code | Description | Example |
| 200 | OK | {  ‘id’:1,  "title": "Capybara",  "organiserName": "Fiona",  "organiserId": 1000,  "eventType": "Food",  "location": "UNSW",  "startTimeandDate": "2042/09/02 02:30:00",  "endTimeandDate": "2042 /10/02 12:30:0",  "banner": "",  "description": "Learn how to train capybara, cause Capy >> llama.",  "availableTicketQuantity": 4,  "totalTicketQuantity": 4,  "ticketPrice": 20  “review”: [{‘reviewId’: 12443, ‘reviewerId’: 2, ‘reviewerName’: ‘Fiona’, comment: ‘What a great event!’, ‘rating’: 5, hostReply: ‘I hate shrek!’},{‘reviewId’: 12403, ‘reviewerId’: 5, ‘reviewerName’: ‘Bob’, comment: ‘What a great event!’, ‘rating’: 5, hostReply: None}] } |
| 400 | Bad Input | {  "error": "Invalid input"  } |

**/event/{eventID} (PUT)**

This route will request to change the data associated with the event entry associated with {eventID}, specifying the field to change, and the data to replace it with.

(path): eventID

(header)

{

"Authorization": "zIzMH0.b37PfwlcH\_cue6yhgvDt2IiNvhRACf79hTNtacYB94Q"

}

(body)

{

"title": "Capybara 🦦",

"organiserId": 1000,

"eventType": "Food",

"location": "UNSW",

"startTimeandDate": "2042/09/02 02:30:00",

"endTimeandDate": "2042/10/02 12:30:0",

"banner": "",

"description": "Learn how to train capybara, cause Capy >> llama.",

"totalTicketQuantity": 4,

"ticketPrice": 20

}

#### **Responses**

|  |  |  |
| --- | --- | --- |
| Code | Description | Example |
| 200 | OK | {} |
| 403 | Forbidden | {  "error": "Invalid Token"  } |
| 400 | Bad Input | {  "error": "Invalid input"  } |

**event/getall/{userID} (GET)**

This route will return all events that a user {userID} has created i.e. is hosting

(header)

{

"Authorization": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImhheWRlbkB1bnN3LmVkdS5hdSIsImlhdCI6MTYwMzk0MzIzMH0.b37PfwlcH\_cue6yhgvDt2IiNvhRACf79hTNtacYB94Q"

}

(path): userID

#### **Responses**

|  |  |  |
| --- | --- | --- |
| Code | Description | Example |
| 200 | OK | {  “events”: [  {  “id”: 10034,  "title": "Capybara 🦦",  "eventType": "Food",  "location": "UNSW",  "startTimeandDate": "2042/09/02 02:30:00",  "endTimeandDate": "2042/10/02 12:30:0",  “banner":" ”  "description": "Learn how to train capybara, cause Capy >> llama.",  "totalTicketQuantity": 4,  "ticketPrice": 20,  },  ]  } |
| 403 | Forbidden | {  "error": "Invalid Token"  } |
| 400 | Bad Input | {  "error": "Invalid input"  } |

**/event/recommendations/{userID} (GET)**

This route will get all recommended events for a particular user.

(header)

{

"Authorization": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImhheWRlbkB1bnN3LmVkdS5hdSIsImlhdCI6MTYwMzk0MzIzMH0.b37PfwlcH\_cue6yhgvDt2IiNvhRACf79hTNtacYB94Q"

}

(path): userID

#### **Responses**

|  |  |  |
| --- | --- | --- |
| Code | Description | Example |
| 200 | OK | [  {  "id": 56513315,  "title": "Capybara 🦦",  "address": "UNSW",  "startTimeandDate": "2042/09/02 02:30:00",  "endTimeandDate": "2042/10/02 12:30:0",  "banner": ""  }  ] |

**/event/cancel/{eventID} (DELETE)**

This route will let event organiser delete an event

(header)

{

"Authorization": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImhheWRlbkB1bnN3LmVkdS5hdSIsImlhdCI6MTYwMzk0MzIzMH0.b37PfwlcH\_cue6yhgvDt2IiNvhRACf79hTNtacYB94Q"

}

(path): eventID

#### **Responses**

|  |  |  |
| --- | --- | --- |
| Code | Description | Example |
| 200 | OK | {'eventID': eventID} |

**/event/getallAttendees/{eventID}**

Given an event id, a list of attendees is returned

(header)

{

"Authorization": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImhheWRlbkB1bnN3LmVkdS5hdSIsImlhdCI6MTYwMzk0MzIzMH0.b37PfwlcH\_cue6yhgvDt2IiNvhRACf79hTNtacYB94Q"

}

(path): eventID

#### **Responses**

|  |  |  |
| --- | --- | --- |
| Code | Description | Example |
| 200 | OK | [{‘attendeeID’: 3, ‘attendeeName’ : ‘Jannet’, isFollowed : False},{‘attendeeID’: 4, ‘attendeeName’ : ‘Alice’, isFollowed : True}  ] |

**Booking**

**/booking/new/{eventID} (POST)** create a new booking

This route should create a new event entry and send it to the backend.

* The next time event/getall or similar routes are called, the change should be reflected

(header)

{

"Authorization": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImhheWRlbkB1bnN3LmVkdS5hdSIsImlhdCI6MTYwMzk0MzIzMH0.b37PfwlcH\_cue6yhgvDt2IiNvhRACf79hTNtacYB94Q"

}

(path): eventID

(body)

{

"userID": 56513315,

"quantity": 1,

“cardNumber”: 1233211234321,

“expirationDate”: 10/26,

“securityCode”: 123

}

#### **Responses**

|  |  |  |
| --- | --- | --- |
| Code | Description | Example |
| 200 | OK | {  "bookingID": 78204786  } |
| 403 | Forbidden | {  "error": "Invalid Token"  } |
| 400 | Bad Input | {  "error": "Invalid input"  } |

**/booking/del/{bookingId} (DELETE)** delete a booking

This route should send a request to the backend to find the event as tagged by eventID, and will delete it from the database

* The next time event/getall or similar routes are called, the change should be reflected

(header)

{

"Authorization": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImhheWRlbkB1bnN3LmVkdS5hdSIsImlhdCI6MTYwMzk0MzIzMH0.b37PfwlcH\_cue6yhgvDt2IiNvhRACf79hTNtacYB94Q"

}

(path): bookingid

#### **Responses**

|  |  |  |
| --- | --- | --- |
| Code | Description | Example |
| 200 | OK | {} |
| 403 | Forbidden | {  "error": "Invalid Token"  } |
| 400 | Bad Input | {  "error": "Invalid input"  } |

**booking/getall/{userID} (GET)**

This route should return all events that a user is attending

(header)

{

"Authorization": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImhheWRlbkB1bnN3LmVkdS5hdSIsImlhdCI6MTYwMzk0MzIzMH0.b37PfwlcH\_cue6yhgvDt2IiNvhRACf79hTNtacYB94Q"

}

(path): userID

#### **Responses**

|  |  |  |
| --- | --- | --- |
| Code | Description | Example |
| 200 | OK | {  “bookings”: [  {  “id”: 10034,  “eventID”: 1056,  “title”: Capybara,  "startTimeandDate": "2042/09/02 02:30:00",  "endTimeandDate": "2042/10/02 12:30:0",  “quantity”: 1,  },  ]  } |
| 403 | Forbidden | {  "error": "Invalid Token"  } |
| 400 | Bad Input | {  "error": "Invalid input"  } |

**Review**

**/review/new/{eventID} (POST)** a user create a review for an event

This route should create a new review entry to be stored in the database, with the entry requiring a userID, comment, and a rating from 1 to 5

(header)

{

"Authorization": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImhheWRlbkB1bnN3LmVkdS5hdSIsImlhdCI6MTYwMzk0MzIzMH0.b37PfwlcH\_cue6yhgvDt2IiNvhRACf79hTNtacYB94Q"

}

(path): eventID

(body):

{

"userID": 56513315,

"comment": “nice event”,

“rate”: 3

}

#### **Responses**

|  |  |  |
| --- | --- | --- |
| Code | Description | Example |
| 200 | OK | {  “reviewId”: 1245  } |
| 403 | Forbidden | {  "error": "Invalid Token"  } |
| 400 | Bad Input | {  "error": "Invalid input"  } |

**/review/reply/{reviewID}** (POST) a host can reply to comment on events which they have organised

(header)

{

"Authorization": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImhheWRlbkB1bnN3LmVkdS5hdSIsImlhdCI6MTYwMzk0MzIzMH0.b37PfwlcH\_cue6yhgvDt2IiNvhRACf79hTNtacYB94Q"

}

(path): reviewID

(body):

{

"reply": “thank you for your review!”

}

|  |  |  |
| --- | --- | --- |
| Code | Description | Example |
| 200 | OK | {  “replyID”: 1245  } |
| 403 | Forbidden | {  "error": "Invalid Token"  } |
| 400 | Bad Input | {  "error": "Invalid input"  } |

**Friends**

**/Friend/follow/{userID} (POST)** the user can follow other users using there userID

(header)

{

"Authorization": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImhheWRlbkB1bnN3LmVkdS5hdSIsImlhdCI6MTYwMzk0MzIzMH0.b37PfwlcH\_cue6yhgvDt2IiNvhRACf79hTNtacYB94Q"

}

(path): userID

#### **Responses**

|  |  |  |
| --- | --- | --- |
| Code | Description | Example |
| 200 | OK | {} |
| 403 | Forbidden | {  "error": "Invalid Token"  } |
| 400 | Bad Input | {  "error": "Invalid input"  } |

**/Friend/unfollow/{userID} (POST)** the user can follow other users using there userID

(header)

{

"Authorization": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImhheWRlbkB1bnN3LmVkdS5hdSIsImlhdCI6MTYwMzk0MzIzMH0.b37PfwlcH\_cue6yhgvDt2IiNvhRACf79hTNtacYB94Q"

}

(path): userID

#### **Responses**

|  |  |  |
| --- | --- | --- |
| Code | Description | Example |
| 200 | OK | {} |
| 403 | Forbidden | {  "error": "Invalid Token"  } |
| 400 | Bad Input | {  "error": "Invalid input"  } |

**=> event, user, comment, rate**

**=> reviewID**

**Message**

**/messages/getall/<userid> (GET)**

Given the userID, we receive all messages for that user

(header)

{

"Authorization": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6ImhheWRlbkB1bnN3LmVkdS5hdSIsImlhdCI6MTYwMzk0MzIzMH0.b37PfwlcH\_cue6yhgvDt2IiNvhRACf79hTNtacYB94Q"

}

(path): userID

#### **Responses**

|  |  |  |
| --- | --- | --- |
| Code | Description | Example |
| 200 | OK | {} |
| 403 | Forbidden | {  "error": "Invalid Token"  } |
| 400 | Bad Input | {  "error": "Invalid input"  } |

**Tier System**

**input = User ID**

**output = credit point**

**Frontend:**

**<100 Bronze full price**

**100 - 500 Silver 5% discount**

**500 - 2000 Gold 10% discount**

**2000 - INF Diamond 20% discount**

**$1 / 1 credit**