

# Routing Table

Task	Method	Pathname	Name	Description	Example
1	GET	/	Home Page	Main page. Respond with index.html	http://localhost:8080
GROUP 1 - http://localhost:8080/29678854					
T1	GET	/add-category	Add category page	Respond with category-add.html	http://localhost:8080/29678854/add-category
T1	POST	/add-category		Perform the add category operation and respond by redirecting the client to category-list.html	
T2	GET	/event-categories	List all categories	Return the list of categories and respond with category-list.html	http://localhost:8080/29678854/event-categories
T3	GET	/search-category	Instruction on how to search category	Respond with instruction-search-category.html	http://localhost:8080/29678854/search-category
T3	GET	/search-category?keyword=	List filtered categories (URL query string)	Respond with category-list (filter).html	http://localhost:8080/29678854/search-category?keyword=Melbourne
T4	GET	/event/EAB-1234	View EAB-1234 event page	Respond with event-page.html	http://localhost:8080/29678854/event/EAB-1234
T4	GET	/event/:id	View selected event page (URL Params)	View selected event on List all events page. Respond with event-page.html	http://localhost:8080/29678854/event/ECB-875
T5	GET	/delete-category	Delete a category by ID	Respond with category-delete.html	http://localhost:8080/29678854/delete-category
T5	POST	/delete-category		Return to list of updated categories. Respond with category event-categories.html	
GROUP 2					
T1	GET	/add-event	Add Event Form	Respond with event-add.html	http://localhost:8080/ChunLing/add-event
T1	POST	/add-event		Create a new event object, store event	

				into database and respond with events.html	
T2	GET	/events	List all events	Return the list of events in the event database and respond with events.html	http://localhost:8080/ChunLing/events
T3	GET	/sold-out-events	List sold out events	Return the list of sold out events in the event database and respond with event-delete.html	localhost:8080/studentName/sold-out-events
T4	GET	/category/:categoryId	Category detail page	Respond with category.html	localhost:8080/ChunLing/category/CME-4500
T5	GET	/delete-event	Delete event by Id (query string)	Return the updated list of events in the event database and respond with events.html	http://localhost:8080/ChunLing/delete-event?eventId=EMH-2107
Extra	GET	/delete	Delete event by body	Return the updated list of events in the event database and respond with events.html	http://localhost:8080/ChunLing/delete
Extra	GET	/instruction	Instruction of using EMA	Respond with instruction page for group II	http://localhost:8080/ChunLing/instruction

# Application Run Instructions

1. Open project in Visual Studio Code
2. Open Terminal
3. Run "npm install" - to install all the dependencies
4. Run "node server.js" OR "nodemon server.js"
5. Open web application (eg. Edge, Chrome) and access <http://localhost:8080/> for our application home page

## [Optional] unsolved bugs and issues

As of now, there are no known bugs

## Group I - Instruction

Name: Uyen Vu

Student ID: 29678854

Student Email: [uvuu0001@student.monash.edu](mailto:uvuu0001@student.monash.edu)

### Task 1 - Add category

<http://localhost:8080/29678854/add-category>

### Task 2 - List all categories

<http://localhost:8080/29678854/event-categories>

### Task 3 - List categories by keyword

<http://localhost:8080/29678854/search-category>

### Task 4 - Show Events Details

<http://localhost:8080/29678854/event/EAB-1234>

### Task 5 - Delete category

<http://localhost:8080/29678854/delete-category>

# Group II - Instruction

Name: Tan Chun Ling

ID: 33402973

Email: [ctan0169@student.monash.edu](mailto:ctan0169@student.monash.edu)

\*\* Look at instruction page as well - <http://localhost:8080/ChunLing/instruction> \*\*

## Model

Related File: models/event.js

Field	Type	Required	Example
Id	Auto generated alpha-numeric	Auto generated	EAB-1234
Name	Alphanumeric	Yes	Fantastic Beasts: The Wonder of Nature
Description	String	No	Visit Melbourne Museum to wonder at the magic of the creatures from the Wizarding World alongside the remarkable animals from London's Natural History Museum collection.
Start Date Time	Date Time	Yes	19 May 2023, 17:00:00
Duration	Number	Yes	60
End Date Time	Date Time	Auto generated	19 May 2023, 18:00:00
IsActive	Boolean	No (default true)	True
Image	String	No (if not provided use default image)	"fanstanstic.png"
Capacity	Number	No (default 1000)	20000
Ticket Available	Number	No (default to capacity)	10000
Category Id	String	Yes	**Note: only accept existing category Id CME-1234

## Router

Related File: server.js

You could access task group II related page with router name 'ChunLing'

Name	ChunLing
Example	<a href="http://localhost/ChunLing/instruction">http://localhost/ChunLing/instruction</a>

## End-points

### Task 1- Add Event:

Users use this page to add a new Event to the app. A page with a form containing all the Event's attributes (see Model) and a button to submit the form data to the backend server are required (Note: use existing category Id while creating the event to categorize it).

### Task 2 - List All Events

In this page, the events are listed in a tabular format. For each row, there is a view button that redirects the client to show event details.

### Task 3 - List Sold-out events

This page lists all sold-out events in a tabular format similar to the "List all events" page above. For each row, a button will redirect the client to show event details.

### Task 4 - Category Detail Page:

Display category attributes. Underneath category attributes, we have listed all the events of the category in a tabular format.

The id of the category is to pass as a URL parameter.

For example you could access the page: <http://localhost:8080/ChunLing/category/CME-1234>

### Task 5 - Delete Event By Id

This endpoint deletes an event by its ID. The ID is sent to the backend server as a variable in the URL's query string. The endpoint will redirect the client to the "list all events" page.

Example: <http://localhost:8080/ChunLing/event/delete-event?id=EAB-1234>

