

# PASSION PROJECT - FASHION VOTING MANAGEMENT SYSTEM

API Endpoint Testing and CRUD Operations Using cURL: Designers, Participants, Shows, and Votes

## DESIGNERS API

This cURL test verifies the CRUD functionalities for the Designer API, including retrieving all designers, fetching specific designer details, creating a new designer with associated shows, and updating an existing designer's details, ensuring that each operation reflects correctly in the database.

```
(base) Genevieve@MacBook-Pro-2 winter-semester-2025 % curl -X GET http://localhost:5157/api/designers -H "Accept: application/json"  
[{"designerId":1,"name":"Kelvin Klein","category":"Casual","shows":["Spring SS2"]},{ "designerId":2,"name":"Ella MIA","category":"Traditional","shows":["Spring SS2"]},{ "designerId":3,"name":"Belle Barbie","category":"Formal","shows":["Summer S1"]},{ "designerId":4,"name":"Ella MIA","category":"Winter Jackets","shows":["NFW"]},{ "designerId":5,"name":"Belle Barbie","category":"Street Wear","shows":["Young Famous"]},{ "designerId":7,"name":"Obum Ife","category":"Bridal Wear","shows":["NFW"]}]%  
(base) Genevieve@MacBook-Pro-2 winter-semester-2025 %  
(base) Genevieve@MacBook-Pro-2 winter-semester-2025 % curl -X GET http://localhost:5157/api/designers/1 -H "Accept: application/json"  
  
{"designerId":1,"name":"Kelvin Klein","category":"Casual","shows":["Spring SS2"]}%  
(base) Genevieve@MacBook-Pro-2 winter-semester-2025 % curl -X POST http://localhost:5157/api/designers \  
-H "Content-Type: application/json" \  
-d '{  
    "name": "Versace",  
    "category": "Streetwear",  
    "selectedShowIds": [1, 2]  
}'%  
{"message":"Designer created successfully!","designerId":10}%  
(base) Genevieve@MacBook-Pro-2 winter-semester-2025 % curl -X GET http://localhost:5157/api/designers/10 -H "Accept: application/json"  
  
{"designerId":10,"name":"Versace","category":"Streetwear","shows":["Spring SS2","Summer S1"]}%  
(base) Genevieve@MacBook-Pro-2 winter-semester-2025 % curl -X PUT http://localhost:5157/api/designers/10 \  
-H "Content-Type: application/json" \  
-d '{  
    "designerId": 10,  
    "name": "Donatella Versace",  
    "category": "High Fashion",  
    "selectedShowIds": [2, 3]  
}'%  
{"message":"Designer updated successfully!"}%  
(base) Genevieve@MacBook-Pro-2 winter-semester-2025 % curl -X GET http://localhost:5157/api/designers/10 -H "Accept: application/json"  
  
{"designerId":10,"name":"Donatella Versace","category":"High Fashion","shows":["Summer S1","Winter"]}%  
(base) Genevieve@MacBook-Pro-2 winter-semester-2025 %
```

This shows various tests involving designers. Specific actions include fetching designer details, updating a designer that doesn't exist, and attempting to create a designer without required fields, resulting in validation errors. There's also a successful deletion of a designer.

```
(base) Genevieve@MacBook-Pro-2 winter-semester-2025 % curl -X GET http://localhost:5157/api/designers/15 -H "Accept: application/json"
{"message":"Designer not found."}%
(base) Genevieve@MacBook-Pro-2 winter-semester-2025 % curl -X PUT http://localhost:5157/api/designers/15 \
-H "Content-Type: application/json" \
-d '{
    "designerId": 15,
    "name": "Donatella Versace",
    "category": "High Fashion",
    "selectedShowIds": [2, 3]
}'
{"message":"Designer not found."}%
(base) Genevieve@MacBook-Pro-2 winter-semester-2025 % curl -X POST http://localhost:5157/api/designers \
-H "Content-Type: application/json" \
-d '{
    "name": "",
    "category": "",
    "selectedShowIds": []
}'
{"type":"https://tools.ietf.org/html/rfc9110#section-15.5.1","title":"One or more validation errors occurred.","status":400,"errors":[{"Name":"Designer name is required."],"Category":["Category is required."]}, {"traceId":"00-01d18ad28bbc9c017d2e081959bb0bee-5bc08fac6bf1260c-00"}]%
(base) Genevieve@MacBook-Pro-2 winter-semester-2025 % curl -X DELETE http://localhost:5157/api/designers/10
{"message":"Designer deleted successfully."}%
(base) Genevieve@MacBook-Pro-2 winter-semester-2025 % curl -X GET http://localhost:5157/api/designers -H "Accept: application/json"
[{"designerId":1,"name":"Kelvin Klein","category":"Casual","shows":["Spring SS2"]},{ "designerId":2,"name":"Ella MIA","category":"Traditional","shows":["Spring SS2"]},{ "designerId":3,"name":"Belle Barbie","category":"Formal","shows":["Summer S1"]},{ "designerId":4,"name":"Ella MIA","category":"Winter Jackets","shows":["NFW"]},{ "designerId":5,"name":"Belle Barbie","category":"Street Wear","shows":["Young Famous"]},{ "designerId":7,"name":"Obum Ife","category":"Bridal Wear","shows":["NFW"]}]%
(base) Genevieve@MacBook-Pro-2 winter-semester-2025 % curl -X GET http://localhost:5157/api/designers/10 -H "Accept: application/json"
{"message":"Designer not found."}%
(base) Genevieve@MacBook-Pro-2 winter-semester-2025 %
```

# SHOWS API

Displays tests related to fetching a list of shows. Specific commands fetch the admin list, all shows, and registered shows for a participant.

```
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X GET "http://localhost:5157/api/ShowsApi/20" -H "Accept: application/json"
>Show not found.%#
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X GET "http://localhost:5157/api/ShowsApi/myshows" -H "Accept: application/json"
[{"showId":3,"showName":"Winter","location":"Humber Lakeshore","startTime":"2025-03-01T09:00:00","endTime":"2025-03-01T15:00:00"}]%
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X GET "http://localhost:5157/api/ShowsApi/list" -H "Accept: application/json"
[{"showId":1,"showName":"Spring SS2","location":"Humber North","startTime":"2025-02-15T09:00:00","endTime":"2025-02-15T15:00:00","participantShows":[],"participants":[],"designerShows":[],"votes":[]},{ "showId":2,"showName":"Summer S1","location":"Humber IGS","startTime":"2025-02-22T10:00:00","endTime":"2025-02-22T15:00:00","participantShows":[],"participants":[],"designerShows":[],"votes":[]},{ "showId":3,"showName":"Winter","location":"Humber Lakeshore","startTime":"2025-03-01T09:00:00","endTime":"2025-03-01T15:00:00","participantShows":[],"participants":[],"designerShows":[],"votes":[]}]%
(base) Genevieve@MacBook-Pro-2 xamppfiles %
```

Displays tests for fetching different lists of shows, including upcoming shows and all shows visible to admins.

```
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X GET "http://localhost:5157/api/ShowsApi/list" -H "Accept: application/json"
[{"showId":1,"showName":"Spring SS2","location":"Humber North","startTime":"2025-02-15T09:00:00","endTime":"2025-02-15T15:00:00","participantShows":[],"participants":[],"designerShows":[],"votes":[]},{ "showId":2,"showName":"Summer S1","location":"Humber IGS","startTime":"2025-02-22T10:00:00","endTime":"2025-02-22T15:00:00"}, {"showId":3,"showName":"Winter","location":"Humber Lakeshore","startTime":"2025-03-01T09:00:00","endTime":"2025-03-01T15:00:00"}, {"showId":6,"showName":"Upcoming Designers","location":"Square One","startTime":"2025-02-03T23:10:00","endTime":"2025-02-03T05:30:00"}, {"showId":7,"showName":"Young Famous","location":"Square One","startTime":"2025-02-03T00:40:00","endTime":"2025-02-03T05:40:00"}, {"showId":13,"showName":"NFW","location":"New York","startTime":"2025-02-04T17:20:00","endTime":"2025-02-04T22:00:00"}]%
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X GET "http://localhost:5157/api/ShowsApi/admin" -H "Accept: application/json"
[{"showId":1,"showName":"Spring SS2","location":"Humber North","startTime":"2025-02-15T09:00:00","endTime":"2025-02-15T15:00:00"}, {"showId":2,"showName":"Summer S1","location":"Humber IGS","startTime":"2025-02-22T10:00:00","endTime":"2025-02-22T15:00:00"}, {"showId":3,"showName":"Winter","location":"Humber Lakeshore","startTime":"2025-03-01T09:00:00","endTime":"2025-03-01T15:00:00"}, {"showId":13,"showName":"NFW","location":"New York","startTime":"2025-02-04T17:20:00","endTime":"2025-02-04T22:00:00"}]%
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X GET "http://localhost:5157/api/ShowsApi/1" -H "Accept: application/json"
{"showId":1,"showName":"Spring SS2","location":"Humber North","startTime":"2025-02-15T09:00:00","endTime":"2025-02-15T15:00:00"}%
(base) Genevieve@MacBook-Pro-2 xamppfiles %
```

Shows the creation of a show. It includes successful creation commands with parameters for the show's name, location, start time, and end time.

```
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X POST "http://localhost:5157/api/ShowsApi/create" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{
    "showName": "Toronto Fashion Show",
    "location": "Humber College",
    "startTime": "2025-02-07T19:15:00Z",
    "endTime": "2025-02-07T21:00:00Z"
}'
{"showId":20,"showName":"Toronto Fashion Show","location":"Humber College","startTime":"2025-02-07T19:15:00Z","endTime":"2025-02-07T21:00:00Z"}%
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X GET "http://localhost:5157/api/ShowsApi/20" -H "Accept: application/json"

{"showId":20,"showName":"Toronto Fashion Show","location":"Humber College","startTime":"2025-02-07T19:15:00Z","endTime":"2025-02-07T21:00:00Z"}%
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X PUT "http://localhost:5157/api/ShowsApi/edit/20" \
-H "Content-Type: application/json" \
-d '{
    "showId": 20,
    "showName": "Toronto Fashion Show",
    "location": "Eaton Center",
    "startTime": "2025-02-07T19:15:00Z",
    "endTime": "2025-02-07T21:00:00Z"
}'
{"message":"Show updated successfully.", "show":{"showId":20,"showName":"Toronto Fashion Show","location":"Eaton Center","startTime":"2025-02-07T19:15:00Z","endTime":"2025-02-07T21:00:00Z","participantShows":[],"participants":[],"designerShows":[],"votes":[]}}%
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X POST "http://localhost:5157/api/ShowsApi/register/20" -H "Accept: application/json"

"Successfully registered."%
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X POST "http://localhost:5157/api/ShowsApi/unregister/20" -H "Accept: application/json"

"Successfully unregistered."%
(base) Genevieve@MacBook-Pro-2 xamppfiles %
```

Includes tests for deleting shows. Successful deletion of past registered shows is shown, along with cases where deletion is attempted for non-existent or inaccessible shows.

```
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X DELETE "http://localhost:5157/api/ShowsApi/deleteregistershow/13" -H "Accept: application/json"
"Past show deleted successfully."%
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X DELETE "http://localhost:5157/api/ShowsApi/deleteregistershow/25" -H "Accept: application/json"
>Show doesn't exist or You are not registered for this show (ID: 25)."%
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X DELETE "http://localhost:5157/api/ShowsApi/delete/20" -H "Accept: application/json"
>Show deleted successfully."%
```

The test validates the registration functionality for a participant, ensuring the system prevents duplicate registrations for the same show and identifies time conflicts with already registered shows. When attempting to register for a show already registered, the system returns "Already registered." Similarly, if a participant tries to register for a show with overlapping timings, the system returns "You are already registered for a show with a time conflict." This ensures accurate registration management.

```
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X POST "http://localhost:5157/api/ShowsApi/register/3" \
-H "Content-Type: application/json" \
-H "Accept: application/json"
"Already registered."%
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X POST "http://localhost:5157/api/ShowsApi/register/21" \
-H "Content-Type: application/json" \
-H "Accept: application/json"
"You are already registered for a show with a time conflict."%
```

# VOTES API

Demonstrates fetching vote details for shows. Includes commands for fetching the total vote count for a show and a breakdown of votes for designers in a particular show.

```
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X GET "http://localhost:5157/api/VotesApi/Vote/20" -H "Accept: application/json"
{"showId":20,"showName": "Toronto Fashion Show", "totalVotes":1,"designers":[{"designerId":2,"name":"Ella MIA","category": "High Fashion", "voteCount":1}]}%
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X GET "http://localhost:5157/api/VotesApi/Vote/13" -H "Accept: application/json"
{"showId":13,"showName": "NFW", "totalVotes":1,"designers":[{"designerId":4,"name":"Ella MIA","category": "Winter Jackets", "voteCount":1},{ "designerId":7,"name": "Obum Ife", "category": "Bridal Wear", "voteCount":0}]}%
```

Focuses on submitting and removing votes. It tests scenarios like submitting votes for a show without designers and successfully submitting and removing votes for a show with valid designers.

```
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X POST "http://localhost:5157/api/VotesApi/SubmitVote" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{
    "showId": 20,
    "designerIds": []
}'

{"message": "You didn't vote for any designer."}%
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X POST "http://localhost:5157/api/VotesApi/SubmitVote" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{
    "showId": 20,
    "designerIds": [2]
}'

 {"message": "Vote submitted successfully!"}%
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X GET "http://localhost:5157/api/VotesApi" -H "Accept: application/json"

[{"voteId":1,"participantId":2,"participantName": "luisdoe", "designerId":5, "designerName": "Belle Barbie", "showId":7, "showName": "Young Famous", "votedAt": "2025-02-03T00:59:40.712923"}, {"voteId":3,"participantId":2,"participantName": "luisdoe", "designerId":4, "designerName": "Ella MIA", "showId":13, "showName": "NFW", "votedAt": "2025-02-04T17:51:04.431818"}, {"voteId":6,"participantId":2,"participantName": "luisdoe", "designerId":2, "designerName": "Ella MIA", "showId":20, "showName": "Toronto Fashion Show", "votedAt": "2025-02-07T19:43:37.442784"}]%
```

Highlights successful and unsuccessful vote submissions. Examples include submitting votes when no designer is selected and successfully voting for a designer.

```
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X POST "http://localhost:5157/api/VotesApi/SubmitVote" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{
    "showId": 20,
    "designerIds": [2]
}'

{"message":"No designers are registered for this show. Voting is not possible."}%
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X PUT http://localhost:5157/api/designers/2 \
-H "Content-Type: application/json" \
-d '{
    "designerId": 2,
    "name": "Ella MIA",
    "category": "High Fashion",
    "selectedShowIds": [20]
}'

 {"message":"Designer updated successfully!"}%
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X POST "http://localhost:5157/api/VotesApi/SubmitVote" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{
    "showId": 20,
    "designerIds": [2]
}'

 {"message":"Vote submitted successfully!"}%
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X POST "http://localhost:5157/api/VotesApi/Unvote" \
-H "Content-Type: application/json" \
-H "Accept: application/json" \
-d '{
    "showId": 20,
    "designerId": 2
}'

 {"message":"Vote removed successfully!"}%
(base) Genevieve@MacBook-Pro-2 xamppfiles %
```

# PARTICIPANTS API

Covers CRUD operations for participants. It includes fetching participant details, updating participant data with new shows, and successful deletions of participants.

```
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X GET "http://localhost:5157/api/participants" \
-H "Content-Type: application/json"

[{"participantId":1,"name":"Jane Doe","email":"janedoe@gmail.com","shows":[]}, {"participantId":2,"name":"luisdoe","email":"luisdoe@gmail.com","shows":[{"showId":3,"showName":"Winter","location":"Humber Lakeshore","startTime":"2025-03-01T09:00:00","endTime":"2025-03-01T15:00:00"}, {"showId":13,"showName":"NFW","location":"New York","startTime":"2025-02-04T17:20:00","endTime":"2025-02-04T22:00:00"}]]%
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X GET "http://localhost:5157/api/participants/1" \
-H "Content-Type: application/json"

{"participantId":1,"name":"Jane Doe","email":"janedoe@gmail.com","shows":[]}]
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X PUT "http://localhost:5157/api/participants/edit/1" \
-H "Content-Type: application/json" \
-d '{
    "participantId": 1,
    "name": "Jane Doe",
    "email": "janedoe@gmail.com",
    "selectedShowIds": [1, 3]
}'

>{"message":"Participant updated successfully!"}%
(base) Genevieve@MacBook-Pro-2 xamppfiles %
```

Displays scenarios involving participants. It includes attempts to fetch, update, or delete participants that result in errors like "Participant not found" or validation issues due to missing required fields.

```
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X GET "http://localhost:5157/api/participants/3" \
-H "Content-Type: application/json"

>{"message":"Participant not found."}%
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X PUT "http://localhost:5157/api/participants/edit/1" \
-H "Content-Type: application/json" \
-d '{
    "participantId": 3,
    "name": "Jane Doe",
    "email": "janedoe@gmail.com",
    "selectedShowIds": [1, 3]
}'

>ID mismatch.%%
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X PUT "http://localhost:5157/api/participants/edit/1" \
-H "Content-Type: application/json" \
-d '{
    "participantId": 1,
    "name": "",
    "email": "",
    "selectedShowIds": []
}'

>{"type":"https://tools.ietf.org/html/rfc9110#section-15.5.1","title":"One or more validation errors occurred.","status":400,"errors":{"Name":["Name is required."],"Email":["Email is required.","The Email field is not a valid e-mail address."]}, "traceId":"00-32950c2f949045f1439d9e082b200aca-0aba708b20083fb0d-00"}%
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X DELETE "http://localhost:5157/api/participants/delete/1" \
-H "Content-Type: application/json"

>{"message":"Participant deleted successfully."}%
(base) Genevieve@MacBook-Pro-2 xamppfiles % curl -X GET "http://localhost:5157/api/participants" \
-H "Content-Type: application/json"

[{"participantId":2,"name":"luisdoe","email":"luisdoe@gmail.com","shows":[{"showId":3,"showName":"Winter","location":"Humber Lakeshore","startTime":"2025-03-01T09:00:00","endTime":"2025-03-01T15:00:00"}, {"showId":13,"showName":"NFW","location":"New York","startTime":"2025-02-04T17:20:00","endTime":"2025-02-04T22:00:00"}]]%
(base) Genevieve@MacBook-Pro-2 xamppfiles %
```