Cem Dereli - 30708

Melih Dilbaz – 31169

In the backend of the web application, detailed information regarding destinations can be seen and people can comment on the destinations. Additionally, itineraries are being created by adding destinations and detailed information regarding those itineraries can be seen as well. Get all itineraries (**getAllItineraries**) is a get method and lists all the itineraries by name and id. **findAll** method is being used to retrieve all the itineraries. Get details of an itinerary (**getItineraryById**) is a get method which returns the details of a single itinerary by its id, using **findById** method. Save new itinerary (**createItinerary**) is a post method which accepts the title of the itinerary as a RequestHeader parameter and creates a new itinerary by saving it. Add destination to the itinerary (**addDestinationToItinerary**) method is a post method which accepts the title of the itinerary and the name of the destination as RequestHeader parameters. It uses **findByNameContainsIgnoreCase** method to find the name of the destination from the destination repository and add the destination to the itinerary by using **addDestination** method. Removing destination from the itinerary (**dropDestinationOffTheEnd**) is a post method that returns the updated destination after removing the last destination in the itinerary list.

Get all categories (**getAllCategories**) is a get method which lists all the categories and returns the ids and names of them. Get details of a destination (**getDetailedInfo**) method is a get method which returns the detailed information of the destination by using the **findDestinationById** method. Get all destinations by category name (**getAllDestinationsByCategoryName**) is a get method which returns the list of destinations that are under the given category name. To get the name of the category, **findByName** method is being used and to get the list of destinations that belongs to that category, the **getDestinationByCategory** method is being used. This last method returns the destinations using **findByCategory** method.

Comment logic implemented using two methods. The first one is getting all comments by destination id (**getAllCommentsByDestinationId**) which is a get method and returns the list of comments by destination id. The second one is the add new comment to destination (**postComment**) method which accepts a comment as a JSON object and returns a response entity. By using **postCommentByDestinationId** method, new comment is created and saved to the specific destination.

Endpoints:

1 – Get all Itineraries (getAllItineraries)

Description: Lists all the itineraries. Id, name and list of destinations are returned.

Address: http://localhost:8080/home/itineraries

Method: GET

Returns: JSON Array:

```
[
  {
    "id": "662bb689c36b8855b0c4292b",
    "name": "destList",
    "list": [
      {
        "id": "662bb501c36b8855b0c42923",
        "name": "Suleymaniye Mosque",
        "text": ". . .",
        "link": "https://en.wikipedia.org/wiki/S%C3%BCleymaniye_Mosque",
        "image": ". . .",
        "category": {
          "categoryName": "Mosque"
        },
        "comments": []
      },
      {
        "id": "662bb501c36b8855b0c4292a",
        "name": "Ataturk Arboretum",
        "text": ". . .",
        "link": "https://en.wikipedia.org/wiki/Atat%C3%BCrk_Arboretum",
        "image": ". . .",
        "category": {
          "categoryName": "Park"
        },
        "comments": []
      }
    ]
```

```
},
{
    "id": "662bb6bbc36b8855b0c4292c",
    "name": "where_to_go_in_Istanbul",
    "list": [
        {
            "id": "662bb501c36b8855b0c42924",
            "name": "Hagia Sophia",
            "text": ". . .",
            "link": "https://en.wikipedia.org/wiki/Hagia_Sophia",
            "image": ". . .",
            "category": {
                "categoryName": "Mosque"
            },
            "comments": []
        }
    ]
},
{
    "id": "662bb6f6c36b8855b0c4292d",
    "name": "IstanbulSummer",
    "list": [
        {
            "id": "662bb501c36b8855b0c4291f",
            "name": "Camlıca Mosque",
            "text": ". . .",
            "link": "https://en.wikipedia.org/wiki/%C3%87aml%C4%B1ca_Mosque",
            "image": ". . .",
            "category": {
                "categoryName": "Mosque"
                },
                "comments": [
                    {
                        "id": "662bbde88cc02c6b93255e3c",
                        "name": "melih",
                        "text": "It seems to have become a waste of money, too huge"
                    },
                    {
                        "id": "662bbe868cc02c6b93255e3d",
                        "name": "cem",
                        "text": "What a big mosque!!!"
                    }
                ]
            }
        ]
    }
]
```

2 – Get details of an Itinerary (getItineraryById)

Description: Details of a single itinerary by its ID. Id, name and list of destinations are returned.

Address: http://localhost:8080/home/itineraries/{id}

Method: GET

Returns: JSON Object:

```
{
    "id": "662bb689c36b8855b0c4292b",
    "name": "destList",
    "list": [
        {
            "id": "662bb501c36b8855b0c42923",
            "name": "Suleymaniye Mosque",
            ". . .",
            "link": "https://en.wikipedia.org/wiki/S%C3%BCleymaniye_Mosque",
            "image": ". . .",
            "category": {
                "categoryName": "Mosque"
            },
            "comments": []
        }
    ]
}
```

3 – Save New Itinerary (createItinerary)

Description: Saves a new itinerary by its title. It expects the title of the itinerary to be provided in the request header.

Address: http://localhost:8080/home/itineraryheader/create

Method: POST

Accepts: Header:

RequestHeader("title") String title

Returns: JSON Object:

```
{
    "id": "662fe37c3f5fa148d6cda97f",
    "name": "exampleItinerary",
    "list": []
}
```

4 – Add Destination to the Itinerary (addDestinationToItinerary)

Description: Adds a destination to the itinerary. Accepts the name of the itinerary and the destination.

Address: http://localhost:8080/home/itineraryheader/add

Method: POST

Accepts: Header:

@RequestHeader("title") String title, @RequestHeader("name") String destName

Returns: JSON Object:

```
{
    "id": "662fe37c3f5fa148d6cda97f",
    "name": "exampleItinerary",
    "list": [
        {
            "id": "662bb501c36b8855b0c42921",
            "name": "Beylerbeyi Palace",
            "text": ". . .",
            "link": "https://en.wikipedia.org/wiki/Beylerbeyi_Palace",
            "image": ". . .",
            "category": {
                "categoryName": "Palace"
            },
            "comments": []
        }
    ]
}
```

5 – Remove Destination from the Itinerary (dropDestinationOffTheEnd)

Description: Removes a destination from the itinerary. Accepts the name of the destination as a parameter.

Address: http://localhost:8080/home/itineraryheader/remove

Method: POST

Accepts: Param:

@RequestParam("title") String title

Returns: JSON Object:

```
{
   "id": "662fe37c3f5fa148d6cda97f",
   "name": "exampleItinerary",
   "list": []
}
```

6 – Get all Categories (getAllCategories)

Description: Lists all the categories. Id and name are returned.

Address: http://localhost:8080/categories/list

Method: GET

Returns: JSON Array:

```
[
   {
      "categoryName": "Church"
   },
   {
      "categoryName": "Park"
   },
   {
      "categoryName": "Mosque"
   },
   {
      "categoryName": "Palace"
   }
]
```

7 – Get details of a Destination (getDetailedInfo)

Description: Details of a single destination by its ID. Id, name, text, link, image, category, and list of comments are returned.

Address: http://localhost:8080/home/destinations

Method: GET

Returns: JSON Object:

```
{
   "id": "662bb501c36b8855b0c4291f",
   "name": "Camlıca Mosque",
   "text": ". . .",
   "link": "https://en.wikipedia.org/wiki/%C3%87aml%C4%B1ca_Mosque",
   "image": ". . .",
   "category": {
      "categoryName": "Mosque"
   },
   "comments": [
      {
         "id": "662bbde88cc02c6b93255e3c",
         "name": "melih",
         "text": "It seems to have become a waste of money, too huge"
      },
      {
         "id": "662bbe868cc02c6b93255e3d",
         "name": "cem",
         "text": "What a big mosque!!!"
      }
   ]
}
```

8 – Get all Destinations by Category Name (getAllDestinationsByCategoryName)

Description: Lists all the destinations by the category name. Id, name, text, link, image, category, and list of comments are returned.

Address: http://localhost:8080/home/destinations/getall

Method: GET

Returns: JSON Array:

```
[
   {
      "id": "662bb501c36b8855b0c4291f",
      "name": "Camlıca Mosque",
      "text": ". . .",
      "link": "https://en.wikipedia.org/wiki/%C3%87aml%C4%B1ca_Mosque",
      "image": ". . .",
      "category": {
         "categoryName": "Mosque"
      },
      "comments": [
         {
            "id": "662bbde88cc02c6b93255e3c",
            "name": "melih",
            "text": "It seems to have become a waste of money, too huge"
         },
         {
            "id": "662bbe868cc02c6b93255e3d",
            "name": "cem",
            "text": "What a big mosque!!!"
         }
      ]
   },
   {
        "id": "662bb501c36b8855b0c42923",
        "name": "Suleymaniye Mosque",
        "text": ". . .",
        "link": "https://en.wikipedia.org/wiki/S%C3%BCleymaniye_Mosque",
        "image": ". . .",
        "category": {
            "categoryName": "Mosque"
        },
        "comments": []
    },
    {
        "id": "662bb501c36b8855b0c42924",
        "name": "Hagia Sophia",
        "text": ". . .",
        "link": "https://en.wikipedia.org/wiki/Hagia_Sophia",
        "image": ". . .",
        "category": {
            "categoryName": "Mosque"
        },
        "comments": []
    }
]
```

9 – Get all Comments by Destination Id (getAllCommentsByDestinationId)

Description: Lists all the comment by the destination Id. Id, name and text are returned.

Address: http://localhost:8080/home/comments

Method: GET

Returns: JSON Array:

```
[
    {
        "id": "662bbde88cc02c6b93255e3c",
        "name": "melih",
        "text": "It seems to have become a waste of money, too huge"
    },
    {
        "id": "662bbe868cc02c6b93255e3d",
        "name": "cem",
        "text": "What a big mosque!!!"
    }
]
```

10 – Add New Comment to Destination (postComment)

Description: Add a comment to a destination. Saves the new comment by its name and the text.

Address: http://localhost:8080/home/post

Method: POST

Accepts: JSON:

```
{
    "name":"ziynet",
    "text":"I think it's one of a kind"
}
```

Returns: Response Entity:

"Comment added successfully"