ManageUserGroupFunction for "Where To?"

This document details the ManageUserGroupFunction Lambda function for the "Where To?" application. This function is used to manage user groups: it creates new groups, adds users to existing groups, and deletes users from groups.

> *Keep in mind, nothing is set in stone, this is just potential at the moment*

Function Summary

The `ManageUserGroupFunction` is responsible for three main actions:

1. Creating a new group in the UserGroups table in DynamoDB.
2. Adding a user to an existing group.
3. Deleting a user from an existing group.

The action to be performed is determined by an action parameter included in the request.

Expected Inputs

The function expects a JSON object with the following structure:

```
{
  "action": "createGroup" | "addUser" | "deleteUser",
  "userId": "<userId>",
  "groupName": "<groupName>",
  "targetUserId": "<targetUserId>"
}
```

- `action`: A string that indicates the action to be performed. It should be one of "createGroup", "addUser", or "deleteUser".
- `userId`: The ID of the user who is performing the action.
- `groupName`: The name of the group. This is needed for all actions.
- `targetUserId`: The ID of the user to be added to or deleted from a group. This is only needed when the action is "addUser" or "deleteUser".

Expected Outputs

The function will return a JSON object with the following structure:

```
{
  "statusCode": 200,
  "body": "<Response Message>"
}
```

- `statusCode`: A status code that indicates the result of the function's execution. A 200 code means the operation was successful.
- `body`: A string containing a message about the result of the function's execution.

Permissions

The `ManageUserGroupFunction` Lambda function needs permissions to read and write to the UserGroups table in DynamoDB. The required permissions can be included in an IAM role attached to the Lambda function. This role should include the `dynamodb:GetItem`, `dynamodb:PutItem`, `dynamodb:UpdateItem`, and `dynamodb:DeleteItem` permissions for the UserGroups table.

Example

An example input to the `ManageUserGroupFunction` might look like this:

```json
{
  "action": "addUser",
  "userId": "user1",
  "groupName": "group1",
  "targetUserId": "user2"
}
```

This request would add user2 to the group named group1.

An example output might look like this:

```json
{
  "statusCode": 200,
  "body": "User 'user2' successfully added to 'group1'."
}
```

This response indicates that the operation was successful.