API Gateway for "Where To?"

This document details the API Gateway setup for the "Where To?" application. The API Gateway will serve as the main point of interaction between the client application and the AWS backend services. The API Gateway is set up with RESTful endpoints that correspond to the operations provided by the Lambda functions.

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

Summary

The API Gateway will interact with the AWS Lambda functions to perform the operations. This includes creating and managing user groups, locations, and attributes. It is designed to follow RESTful principles, meaning that each endpoint represents a specific resource and the HTTP method represents the action to be performed on that resource.

Endpoints

Here's a more detailed breakdown of the endpoints:

- `POST /usergroup`: Create a new user group. Maps to the `ManageUserGroupFunction`.
- `PUT /usergroup/:groupId/user/:userId`: Add a user to a group. Also maps to the `ManageUserGroupFunction`.
- `GET /usergroup/:groupId`: Get all information about a user group, including locations and attributes. Maps to a function that retrieves this data.
- `POST /location`: Create a new location. Maps to the `ManageLocationFunction`.

- `PUT /location/:locationId`: Edit a location. Maps to the `ManageLocationFunction`.
- `DELETE /location/:locationId`: Delete a location. Maps to the `ManageLocationFunction`.
- `GET /location/:locationId`: Get information about a location. Maps to a function that retrieves this data.
- `POST /attribute`: Create a new attribute. Maps to the `ManageAttributeFunction`.
- `PUT /attribute/:attributeId`: Edit an attribute. Maps to the `ManageAttributeFunction`.
- `DELETE /attribute/:attributeId`: Delete an attribute. Maps to the `ManageAttributeFunction`.
- `GET /attribute/:attributeId`: Get information about an attribute. Maps to a function that retrieves this data.

Permissions

The API Gateway needs the `lambda:InvokeFunction` permission for the respective Lambda functions. This allows the API Gateway to trigger the Lambda functions. This can be provided through an IAM role.

Authorization

Authorization for the API Gateway can be provided through **Amazon Cognito User Pools**. Here are the general steps for setting this up:

1. Set up a user pool in Amazon Cognito.
2. In the API Gateway console, for each method, set the Authorization to the Amazon Cognito user pool.
3. When calling the API Gateway from the client application, include the JWT ID token from Cognito in the `Authorization` header.

Example

Here's an example of how the `POST /usergroup` endpoint might be used:

- **Request**:

```
POST /usergroup HTTP/1.1
Content-Type: application/json
Authorization: Bearer <JWT token>

{
  "userId": "user1",
  "groupName": "group1"
}
```

- **Response**:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "statusCode": 200,
  "body": "User group 'group1' successfully created."
}
```

This creates a new user group named "group1" by the user "user1". The JWT token provided in the `Authorization` header authenticates the user.