Today, most web applications use an API to access and manage data. It's critical, from a security point of view, that users have limited access to this data and how they manipulate it.

From Wikipedia, an ACL (Access Control List) specifies which users are granted access to objects, as well as what operations are allowed on given objects.

Use JavaScript to implement an Access Control List module that can be used in an api application to limit access of users with different roles to the API endpoints.

Creating roles:

A user role can be defined by calling a `createRole` method on the module, like so:

```
import acl from 'acl';

// create different roles
acl.createRole('admin');
acl.createRole('user');
acl.createRole('guest');
```

Setting permissions:

Permissions should be defined using functions `a` and `an`, like so:

```
import { a, an } from 'acl';

// admin can list all users
an('admin').can('get').from('/users');

// admin can create users
an('admin').can('post').to('/users');

// user can post an article only when it's his data
a('user').can('post').to('/users/:userId/articles').when((params, user) =>
user.id === params.userId);

// guest can get data from articles
a('guest').can('get').from('/articles');
```

Function an is an alias of a. It is simply a function that accepts the role as a string.

Function can accepts an HTTP verb: get, post, delete, patch, put and any other supported verb.

Function to is an alias of from. This function accepts an endpoint as a string.

Optionally, the developer can set a condition using when to restrict access to this endpoint based on certain logic. The when function should return a boolean.

The when function always receives a an object params as a first parameter. The params object holds values for defined URL parameters.

An example would be:

For a URL: /users/:userId/articles when checking against a request with URL: /users/12/articles, the params object would be:

```
{
  userId: 12
}
```

The rest of the parameters will be passed by the developer when checking permissions for a role.

Checking permissions:

Check if a given role can perform a given action on a given endpoint like so:

```
import { check } from 'acl';

check.if('guest').can('post').to('/users'); // false
check.if('admin').can('post').to('/users'); // true

// check if a user can post to articles
check.if('user').can('post').to('/users/10/articles').when({ id: 10 }); // true
```