

Acl

August 18, 2026

Acl

Kind: Interface

Source: `packages/microservices/external/kafka.interface.ts`

Part of: [Microservices](#)

`Acl` defines a Kafka access control list entry used to describe which principal can perform a specific operation against a host. It combines an authenticated identity, network host, operation type, and permission type for Kafka authorization workflows.

Properties

Property	Type
<code>principal</code>	<code>string</code>
<code>host</code>	<code>string</code>
<code>operation</code>	<code>AclOperationTypes</code>
<code>permissionType</code>	<code>AclPermissionTypes</code>

Diagram

```
graph LR
  A[Acl] --> P[P[principal: string]]
  A --> H[H[host: string]]
  A --> O[O[operation: AclOperationTypes]]
  A --> PT[PT[permissionType: AclPermissionTypes]]

  P --> I[I[Authenticated user or service identity]]
  H --> N[N[Allowed source host]]
  O --> K[K[Kafka operation]]
  PT --> R[R[Allow or deny permission]]
```

Usage

```
import {
  Acl,
  AclOperationTypes,
  AclPermissionTypes,
} from '@nestjs/microservices';

const consumerReadAcl: Acl = {
  principal: 'User:analytics-service',
  host: '*',
  operation: AclOperationTypes.READ,
  permissionType: AclPermissionTypes.ALLOW,
};

// Pass the ACL definition to Kafka administration or authorization setup.
console.log(consumerReadAcl);
```

AI Coding Instructions

- Use the `User:<name>` principal convention expected by Kafka when defining user or service identities.
- Set `host` to `'*'` only when access from all hosts is intended; prefer a specific host where possible.
- Use `AclOperationTypes` and `AclPermissionTypes` enum values instead of hard-coded strings.
- Pair ACL entries with the appropriate Kafka resource definition when creating or deleting broker ACLs.
- Review `ALLOW` and `DENY` rules carefully, as conflicting ACLs can produce unexpected authorization behavior.