

# AVS Node API

The AVS Node HTTP RESTful API provides a common interface for Nodes designed for the EigenLayer protocol.

## Base URL

The base URL for the API is:

/eigen

## API Versioning

This document describes the AVS Node API version `v0.0.1`. The API version follows the same EigenLayer AVS Node specification version.

The API version can be included in the query string as follows:

`/eigen/endpoint?version=0.0.2` In case of handling a request without the version query parameter, the API will respond with the latest version of the API.

If the requested version is not supported, the API will respond with a 404 Not Found error and the following message:

```
{ "message": "API version not found" }
```

## Node

Endpoints to query Node related information.

### GET /eigen/node

Provides AVS Node and API identity information in a format similar to [HTTP User-Agent](#).

#### Parameters

No parameters

#### Responses

200 - Request successful

Content-Type: application/json

- Example Value
- Schema

```
{ "node_name": "EigenLayer-AVS", "spec_version": "v0.0.1", "node_version": "v1.0.0" } { "type": "object", "properties": {  
  "node_name": { "type": "string", "description": "Name of the node" }, "spec_version": { "type": "string", "description": "AVS  
Specification version" }, "node_version": { "type": "string", "description": "Version of the node" } }, "required": [ "node_name",  
"spec_version", "node_version" ] } 500 - Internal server error
```

Content-Type: application/json

- Example Value
- Schema

```
{ "message": "Internal server error" } { "type": "object", "properties": { "message": { "type": "string", "description": "Error  
message" } }, "required": [ "message" ] }
```

### GET /eigen/node/health

Returns AVS Node health status in HTTP status codes.

#### Parameters

No parameters

#### Responses

200 - AVS Node is healthy

206 - AVS Node is partially healthy. It is either initializing or some backing services are not healthy.

503 - AVS Node is unhealthy or having issues.

## GET /eigen/node/services

Returns a list of all backing services that are currently registered with the AVS Node setup. Useful to determine if a service is currently registered with the AVS Node and to get the service ID for health endpoint.

### Parameters

No parameters

### Responses

200 - Request successful

Content-Type: application/json

- Example Value
- Schema

```
{ "services": [ { "id": "db-1", "name": "Database", "description": "Database description", "status": "Up", }, { "id": "idx-2", "name": "Indexer", "description": "Indexer description", "status": "Down", } ] } { "type": "object", "properties": { "services": { "type": "array", "description": "List of backing services", "items": { "type": "object", "properties": { "id": { "type": "string", "description": "Service ID" }, "name": { "type": "string", "description": "Service name" }, "description": { "type": "string", "description": "Service description" }, "status": { "type": "string", "description": "Service status. Possible values are Up, Down, or Initializing." }
```

```
}, "required": [ "id", "name", "description" ] } } }, "required": [ "services" ] } 500 - Internal server error
```

Content-Type: application/json

- Example Value
- Schema

```
{ "message": "Internal server error" } { "type": "object", "properties": { "message": { "type": "string", "description": "Error message" } }, "required": [ "message" ] }
```

## GET/eigen/node/services/{service\_ID}/health

Returns the health status of the given service in HTTP status codes. The service ID is the service ID returned by the/eigen/services endpoint. The service is considered healthy if the service is registered with the AVS Node and the service is currently online.

### Parameters

service\_ID string (path) - Service ID

### Responses

200 - Backing service is healthy

206 - Backing service is partially healthy. It is either initializing or not ready yet.

503 - Backing service is unhealthy or having issues. [Previous AVS Node API Next Implementation reference](#)