

# Integrazione MSAL in Node.js con Azure AD (Entra ID)

Obiettivo:

Autenticare un'applicazione Node.js (server-side) usando Microsoft Entra ID (Azure AD) con MSAL, evitando la gestione manuale di login/password/token JWT.

---

## FASE 1: Registrare l'app in Azure AD (Entra ID)

1. Vai su <https://portal.azure.com>
2. Cerca e apri "Microsoft Entra ID"
3. Seleziona "App registrations" > "New registration"
  - Name: cincotti-server
  - Supported account types: Single tenant (o Multitenant se necessario)
  - Redirect URI: lascia vuoto
4. Clic su "Register"

---

## FASE 2: Configura i segreti (Client Secret)

1. Vai su "Certificates & secrets"
2. Clicca "New client secret"
  - Descrizione: cincotti-secret
  - Scadenza: scegli durata
3. Clic "Add" e copia il valore subito

---

### FASE 3: Ottieni i dati per l'integrazione

Recupera da "Overview" dell'app registrata:

- Client ID
- Tenant ID
- Client Secret (dal passaggio precedente)

---

### FASE 4: Assegna i permessi API

1. Vai su "API permissions"
2. Clicca "Add a permission" > Microsoft Graph
  - Application permissions
  - Aggiungi es. User.Read.All, Directory.Read.All
3. Clicca "Grant admin consent"

---

### FASE 5: Codice Node.js con MSAL

Installa libreria:

```
$ npm install @azure/msal-node
```

auth.js:

-----

```
const { ConfidentialClientApplication } = require("@azure/msal-node");
```

```
const config = {
```

```
auth: {  
  clientId: process.env.CLIENT_ID,  
  authority: `https://login.microsoftonline.com/${process.env.TENANT_ID}`,  
  clientSecret: process.env.CLIENT_SECRET,  
},  
};
```

```
const cca = new ConfidentialClientApplication(config);
```

```
async function getToken() {  
  const result = await cca.acquireTokenByClientCredential({  
    scopes: ["https://graph.microsoft.com/.default"],  
  });  
  return result.accessToken;  
}
```

```
module.exports = { getToken };
```

-----

```
server.js:
```

-----

```
const express = require("express");  
  
const { getToken } = require("./auth");  
  
const app = express();
```

```
app.get("/token", async (req, res) => {
```

```

try {
  const token = await getToken();
  res.send({ token });
} catch (e) {
  console.error(e);
  res.status(500).send("Errore autenticazione MSAL");
}
});

app.listen(3000, () => {
  console.log("Server in ascolto su http://localhost:3000");
});

```

-----

---

.env (variabili da impostare):

```

CLIENT_ID=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx
TENANT_ID=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx
CLIENT_SECRET=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

```

---

Test:

Chiama l'endpoint /token per ricevere un access token valido per Microsoft Graph o altre API protette da Azure AD.