

# Node.js: come creare un middleware di autenticazione in ExpressJS

In ExpressJS possiamo creare un middleware per verificare se un utente è autenticato.

Presupponendo l'utilizzo di express-session per la gestione delle sessioni utente, quello che dobbiamo fare è verificare se nella sessione corrente è presente una proprietà che indichi l'avvenuta autenticazione da parte dell'utente.

Se questa proprietà non è presente o non possiede il valore atteso, possiamo reindirizzare l'utente alla pagina di login.

```
'use strict';

module.exports = (req, res, next) => {
  if (!req.session.isLoggedIn) {
    return res.redirect('/login');
  }
  next();
};
```

Il middleware così creato può essere usato come argomento nella definizione delle nostre route. Per applicarlo ad una route specifica possiamo scrivere:

```
'use strict';
```

```
const app = require('express')();
const auth = require('./middleware/auth');
const PORT = process.env.PORT || 3000;

app.get('/backend', auth, (req, res, next) => {
  //...
});

app.listen(PORT);
```

Per applicarlo ad un gruppo di route possiamo utilizzarlo come middleware di un oggetto Router.

```
'use strict';

const express = require('express');
const router = express.Router();
const auth = require('../middleware/auth');

router.use(auth);

router.get('/', (req, res, next) => {
  //...
});
router.get('/files', (req, res, next) => {
  //...
});
router.get('/orders', (req, res, next) => {
  //...
});

module.exports = router;
```

A questo punto nel file principale definiamo il punto di mount del nostro router all'interno dell'app.

```
'use strict';

const app = require('express')();
const auth = require('./middleware/auth');
const PORT = process.env.PORT || 3000;
const adminRoutes = require('./routes/admin');

app.use('/backend', adminRoutes);

app.listen(PORT);
```

In questo modo la route `/backend` e tutte le sue sottoroute verranno protette dal nostro middleware di autenticazione.