Node.js: generare una CSR e una chiave privata

In Node.js, è possibile generare una CSR (Certificate Signing Request) e una chiave privata RSA utilizzando il modulo integrato crypto. La CSR viene utilizzata per richiedere un certificato SSL (Secure Sockets Layer) da un'autorità di certificazione, mentre la chiave privata RSA viene utilizzata per crittografare i dati trasmessi attraverso la connessione SSL.

La soluzione è la seguente:

```
'use strict';

const crypto = require('crypto');

const { privateKey } =
crypto.generateKeyPairSync('rsa', {
    modulusLength: 2048,
    publicKeyEncoding: {
        type: 'pkcs1',
        format: 'pem'
    },
    privateKeyEncoding: {
        type: 'pkcs1',
        format: 'pem'
    }
});

const csr = crypto.createSign('RSA-SHA256');
const csrInfo = [
    '-----BEGIN CERTIFICATE REQUEST-----\n',
```

```
    'CN=example.com\n',
    'O=Example Organization\n',
    'L=San Francisco\n',
    'ST=California\n',
    'C=US\n',
    '-----END CERTIFICATE REQUEST-----\n'
];
csr.update(csrInfo.join(''));
csr.end();
const csrData = csr.sign(privateKey, 'base64');
```

Viene prima generata una chiave privata RSA con una lunghezza di 2048 bit e viene codificata in formato PEM. Quindi viene generata una CSR utilizzando l'algoritmo di firma RSA-SHA256. È importante notare che la CSR deve contenere informazioni sul dominio per il quale si richiede il certificato SSL.

È molto importante anche notare che a titolo di esempio stiamo utilizzando la modalità sincrona di generazione della CSR e della chiave privata. Al fine di non bloccare l'Event Loop nel contesto di un'applicazione, si dovrebbero sempre usare i metodi asincroni del modulo `crypto`.