

Fix error:0308010C:digital envelope routines::unsupported

Notebook: REACT JS 16.12.0

Created: 10/31/2023 4:55 PM

Updated: 10/31/2023 4:56 PM

Author: Arin Suga

URL: <https://www.datainfinities.com/49/error-0308010C-digital-envelope-routines-unsupp...>

Fix error:0308010C:digital envelope routines::unsupported

<https://www.datainfinities.com/49/error-0308010C-digital-envelope-routines-unsupported>

The "error:0308010C:digital envelope routines::unsupported" occurs when you use Node.js of version 17+ and use OpenSSL v3.0 which has some major changes in its algorithm.

The error can be solved in the following ways.

- Set the NODE_OPTIONS environment variable to --openssl-legacy-provider
- Pass the --openssl-legacy-provider flag to Webpack
- For Create React App applications, upgrade react-scripts to v5.0.0 or higher
- Downgrade the installed node.js version to 16 or less

What causes error:0308010C:digital envelope routines::unsupported

The Node.js V17+ with OpenSSL 3.0 moved MD4 algorithms into OpenSSL 3.0's legacy provider. A provider is a collection of algorithms that can be added to our application when it is required. A legacy provider contains algorithms that are unsafe security-wise and outdated. The legacy provider will not load by default.

Webpack uses some cryptographic methods based on the MD4 algorithm which is available in legacy providers for Node.js v17+ with

OpenSSL 3.0.

How to solve the error:0308010C:digital envelope routines::unsupported

There are some different ways to solve the error. Let us discuss them one by one.

Set the **NODE_OPTIONS** environment variable to **--openssl-legacy-provider**

To set the `NODE_OPTIONS` environment variable to **--openssl-legacy-provider**, open your terminal and run the below command.

```
# for macOS, Linux or Windows Git Bash
export NODE_OPTIONS=--openssl-legacy-provider

# for Windows CMD (Command Prompt)
set NODE_OPTIONS=--openssl-legacy-provider

# for Windows PowerShell
$env:NODE_OPTIONS="--openssl-legacy-provider"

# for Docker (in your Dockerfile)
ENV NODE_OPTIONS="--openssl-legacy-provider"
```

Pass the **--openssl-legacy-provider** flag to **Webpack**

If the above solution does not work, you can try to pass the `--openssl-legacy-provider` flag to the webpack. In a React app, open the `package.json` file in the root directory and you can pass `--openssl-legacy-provider` to the start script like:

```
{
  "scripts": {
    "start": "react-scripts start --openssl-legacy-provider",
  }
}
```

If the error is still persisting, then go for the next method of solution.

For Create React App applications, upgrade **react-scripts** to **v5.0.0** or higher

Check the installed version of `react-scripts` on your application in the `package.json` file. If the version is less than `v5.0.0`, then upgrade the

version of react-scripts to v5.0.0 or higher.

First, remove the currently installed react-scripts by running the below command.

```
# For npm
npm uninstall react-scripts

# For yarn
yarn remove react-scripts
```

To install react-scripts V5.0.0 run the below command.

```
# For npm
npm install --save --save-exact react-scripts@5.0.0

# For yarn
yarn add --exact react-scripts@5.0.0
```

If you encounter errors after upgrading, then you may want to reinstall your dependencies. For this, delete your node_modules folder and package-lock.json from your application folder. Then clean your npm cache using the below command.

```
npm cache clean --force
```

Next, reinstall your dependencies by running npm install (or yarn).

```
npm install
```

For more details read the [change log](#).

Downgrade the installed node.js version to 16 or less

Another solution for this error is to downgrade the installed node.js version to 16 or less. You can check the currently installed version of node.js by running the following command.

```
node -v
```

If you are running Node.js via nvm, then you can easily downgrade the node.js version by running the below command.

```
nvm install v16
```

For more about NVM, read it on [github](#).

Conclusion

To solve the error:0308010C:digital envelope routines::unsupported, you have to enable the legacy provider for Node.js of version 17+ and use OpenSSL v3.0. For this set the NODE_OPTIONS environment variable to --openssl-legacy-provider or pass the --openssl-legacy-provider flag to webpack. Other solutions are upgrading react-scripts to a version greater than 5 or downgrading the installed node.js version to 16 or less.

