

How to compress and decompress files in Node.js

Take advantage of the Archiver and Unzipper packages to compress and decompress files in Node.js very easily. Here are detailed instructions.

Take advantage of the **Archiver and Unzipper** packages to compress and decompress files in Node.js very easily. Here are detailed instructions.



File archiving is such an everyday activity in the modern world that everyone can see the benefits of compressing a file or folder into a smaller, portable format.

ZIP is one of the most popular archive formats used in the tech world and is recommended by many people who need to convert raw files for better storage, more efficient conversion or other reasons.

So why should you consider compressing files and how to compress files to ZIP & extract them back to their original state programmatically using Node.js?

Why do you need to compress files?

Files and folders can be so large that sharing and converting them becomes difficult due to disk space limitations or taking too long to upload to the cloud.

In those cases, you should compress the file or folder so that they are smaller in size. In addition to easy file conversion, the following are also the reasons why many people want to compress files:

1. Efficient storage

2. Better file structure & arrangement
3. Security (file encryption and password protection)
4. Ensure file integrity
5. File version management

Node.js . Archiver & Unzipper Package

The Archiver package provides a library of functions that leverage Node.js streams to create compressed archives.

By default, the Archiver package supports many compression formats, including ZIP, GZIP, and TAR. This package also allows you to create archives from files and folders, and splits large compressed files into smaller parts. It also allows you to remove or filter files during compression.

Unzipper is an extremely efficient package for extracting ZIP archives in Node.js. This package provides an easy-to-use API that allows programmers to extract ZIP files with just a few lines of code.

The Archiver & Unzipper packages are the two options in this tutorial because they integrate seamlessly with Node.js' fs module, ensuring smooth and simple compatibility.

How to compress files to ZIP format in Node.js

Compressing files to ZIP format in Node.js is as easy as any other language thanks to the Archiver package. To create a ZIP archive in Node.js in this tutorial, you need to set up a Node.js development environment on your computer.

You will create a simple Node.js script to compress files and folders to ZIP format. Create a new Node project on your computer by running the following commands:

```
mkdir node-zip-archiver cd node-zip-archiver npm init -y
```

Next, you need to install the Archiver package in the project. Run **npm install archiver --save** in **terminal** to install it. When the package installation is complete, create a new file in the project directory and name it as you wish. For example, **app.js** or **archiver.js**.

The fs module handles file operations, and the Archiver package handles compressing files & folders into ZIP format, so this script needs both modules.

Create ZIP file from file

The following code implements a function that accepts a file as an argument and creates a compressed ZIP version of the file.

```
const archiver = require('archiver') const fs = require('fs') // T?o ZIP t?  
file const createZipFromFile = (file) => { const filePath = __dirname + '/' + f
```

This function takes the filename of the file to compress and creates a resulting file with the same name. The function then creates a new archive at level 9 (the highest) and uses the pipe function to convert the resulting compressed file streams into the input of the output file.

The file function adds a file to the archive. It accepts the file path as parameter and an optional options parameter where you can specify the properties of the file in the repository.

Create ZIP file from folder

The process is the same as above, except that you will use the **directory** function of the Archiver package as opposed to the file in the previous function.

Here is the function code to compress the folder into a ZIP file:

```
// tạo ZIP từ thư mục
const createZipFromFolder = (folder) => { const folderPath = __dirname + '/' +
```

How to unzip files in Node.js

Run the following command in the terminal to install the Unzipper package in the project:

```
npm install unzipper --save
```

After installing the package, enter it in the code and use the ZIP extract function as shown below:

```
const unzipper = require("unzipper") //hàm gì?
if (file ZIP) const extractZip = async (file) => { const filePath = __dirname +
```

It's done! Hope the article is useful to you!

You finished reading the article "**How to compress and decompress files in Node.js**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.