

# Package ‘dir2json’

May 8, 2026

**Type** Package

**Title** Convert Directory to JSON

**Version** 0.1.0

**Description** Convert a directory structure into a JSON format. This package lets you recursively traverse a directory and convert its contents into a JSON object, making it easier to import code base from file systems into large language models.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.3.1

**Imports** base64enc, fs, jsonlite, rlang

**Suggests** knitr, rmarkdown

**URL** <https://github.com/parmsam/dir2json-r>,  
<https://parmsam.github.io/dir2json-r/>

**BugReports** <https://github.com/parmsam/dir2json-r/issues>

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Sam Parmar [aut, cre, cph]

**Maintainer** Sam Parmar <parmartsam@gmail.com>

**Repository** CRAN

**Date/Publication** 2025-05-27 09:00:05 UTC

## Contents

as_file_list . . . . .	2
json_decode_dir . . . . .	2
json_encode_dir . . . . .	3
text_file_extensions . . . . .	4
validate_dir_json . . . . .	4
write_files . . . . .	5

<b>Index</b>	<b>6</b>
--------------	----------

---

as_file_list	<i>Convert a File to a List</i>
--------------	---------------------------------

---

**Description**

This function converts a file into a structured list for encoding.

**Usage**

```
as_file_list(path, name = fs::path_file(path), type = NULL)
```

**Arguments**

path	A character string specifying the file path.
name	An optional character string specifying the file name.
type	An optional character string specifying the file type ("text" or "binary").

**Value**

A structured list representing the file.

**Examples**

```
## Not run:  
# Convert a text file to a list  
as_file_list("file.txt")  
  
# Convert a binary file to a list  
as_file_list("image.jpg", type = "binary")  
  
## End(Not run)
```

---

json_decode_dir	<i>Decode JSON to a Directory</i>
-----------------	-----------------------------------

---

**Description**

This function decodes a JSON string into a directory structure.

**Usage**

```
json_decode_dir(json_data, dir)
```

**Arguments**

json_data	A JSON string representing the directory's contents.
dir	A character string specifying the target directory.

**Value**

None. Creates files in the specified directory.

**Examples**

```
## Not run:
# Decode JSON back into a directory
json_decode_dir(json_str, "output_dir")

## End(Not run)
```

---

json_encode_dir	<i>Encode a Directory to JSON</i>
-----------------	-----------------------------------

---

**Description**

This function encodes all files in a directory into a JSON format.

**Usage**

```
json_encode_dir(
  dir,
  type = c("text", "binary"),
  metadata = NULL,
  ignore = NULL
)
```

**Arguments**

dir	A character string specifying the directory to encode.
type	A character vector specifying the file types to include ('text', 'binary', or both). One or both of the following options: * 'text': Text files (e.g., '.txt', '.csv', '.json'). * 'binary': Binary files (e.g., '.bin', '.exe', '.jpg'). Defaults to both.
metadata	A character vector specifying additional metadata to include in the JSON ('file_size', 'creation_time', 'last_modified_time'). One or both of the following options: * 'file_size': Size of the file in bytes. * 'creation_time': Creation time of the file. * 'last_modified_time': Last modified time of the file. Defaults to 'NULL.'
ignore	A character vector specifying file names to exclude from encoding. Defaults to 'NULL.'

**Value**

A JSON string representing the directory's contents.

**Examples**

```
## Not run:
# Encode all files in a directory to JSON
json_str <- json_encode_dir("mydir")

# Encode only text files, including file size metadata
json_str <- json_encode_dir("mydir", type = "text", metadata = "file_size")

# Ignore specific files
json_str <- json_encode_dir("mydir", ignore = c("ignore.txt"))

## End(Not run)
```

---

text\_file\_extensions    *Get Text File Extensions*

---

**Description**

This function returns a vector of common text file extensions.

**Usage**

```
text_file_extensions()
```

**Value**

A character vector of text file extensions.

**Examples**

```
text_file_extensions()
```

---

validate\_dir\_json    *Validate Directory JSON Structure*

---

**Description**

Checks if a JSON string is compliant with the expected schema for a directory structure.

**Usage**

```
validate_dir_json(json_data)
```

**Arguments**

json\_data    A JSON string representing the directory's contents.

**Value**

TRUE if valid, otherwise throws an error.

**Examples**

```
## Not run:  
# Validate a JSON string for directory structure  
validate_dir_json(json_str)  
  
## End(Not run)
```

---

write_files	<i>Write Files to a Directory</i>
-------------	-----------------------------------

---

**Description**

This function writes files to a specified directory based on a structured list.

**Usage**

```
write_files(sl_app, dest)
```

**Arguments**

sl_app	A structured list representing files and their contents.
dest	A character string specifying the target directory.

**Value**

None. Writes files to the specified directory.

**Examples**

```
## Not run:  
# Write files from a structured list to a directory  
write_files(sl_app, "output_dir")  
  
## End(Not run)
```

# Index

`as_file_list`, [2](#)

`json_decode_dir`, [2](#)

`json_encode_dir`, [3](#)

`text_file_extensions`, [4](#)

`validate_dir_json`, [4](#)

`write_files`, [5](#)