

# Package ‘cidtree’

May 7, 2024

**Type** Package

**Title** cidtree: A Package for mapping concepts using a tree-based data dictionary

**Version** 0.1.0

**Author** Jake Peters

**Maintainer** Jake Peters <jcbptrs@gmail.com>

## Description

This package uses `data.tree` to construct a tree object from JSON representing a data dictionary. This package was originally developed for the Connect for Cancer Prevention Cohort Study.

**License** MIT License

**Encoding** UTF-8

**LazyData** true

**Suggests** knitr,  
rmarkdown

**VignetteBuilder** knitr

**RoxygenNote** 7.3.1

**Roxygen** list(markdown = TRUE)

**URL** <https://github.com/Analyticsphere/cidtree/>, <https://analyticsphere.github.io/cidtree/>

## R topics documented:

construct_dictionary_tree . . . . .	2
extract_cids . . . . .	2
get_cid . . . . .	3
get_key . . . . .	3
get_meta . . . . .	4
get_path_to_example_dictionary . . . . .	5
get_responses . . . . .	5
get_var_name . . . . .	6
is_valid_cid . . . . .	6
<b>Index</b>	<b>8</b>

---

```
construct_dictionary_tree
```

*Construct a Dictionary Tree from JSON Files*

---

### Description

This function reads JSON files from a specified directory, creates nodes for each file, and builds a hierarchical tree based on the relationships defined within the files.

### Usage

```
construct_dictionary_tree(path = "../data/test_dictionary")
```

### Arguments

`path`                      The path to the directory containing JSON files.

### Value

A Node object representing the root of the dictionary tree.

### Examples

```
path <- get_path_to_example_dictionary()
print(path)
dd <- construct_dictionary_tree(path)
# Print output so that it does not wrap on the screen
output <- capture.output(print(dd, 'concept_str', 'concept_type', 'pathString'))
cat(output, sep="\n")
```

---

```
extract_cids
```

*Extract Nine-Digit Concept IDs from a String*

---

### Description

This function extracts all nine-digit sequences from a given input string and returns them as a concatenated single string separated by slashes.

### Usage

```
extract_cids(input_string)
```

### Arguments

`input_string`      A character string from which to extract nine-digit numbers.

### Value

A character string containing all found nine-digit numbers concatenated with a slash ("/") separator. Returns NA if no nine-digit numbers are found.

**Examples**

```
# Extract nine-digit numbers from a sample string
extract_cids("123456789 d_987654321 other text")
# Output: "123456789/987654321"

extract_cids("no nine-digit numbers")
# Output: NA
```

---

**get\_cid***Retrieve the Concept ID Associated with a Key*

---

**Description**

This function returns the concept ID (cid) corresponding to a given key within the data.tree object representing the data dictionary.

**Usage**

```
get_cid(dd, key)
```

**Arguments**

dd	A data.tree object representing the data dictionary.
key	A key to locate the corresponding concept ID.

**Value**

The concept ID associated with the given key. If no match is found, returns NULL.

**Examples**

```
# Assuming `dd` is a properly structured `data.tree` object
# and 'my_key' is a valid key in the tree:
path <- get_path_to_example_dictionary() # Replace with path to your dictionary
dd <- construct_dictionary_tree(path)
concept_id <- get_cid(dd, 'University of Chicago Medicine')
concept_id
```

---

**get\_key***Retrieve the Key Associated with a Concept ID from a Data Tree*

---

**Description**

This function takes a data.tree object representing a data dictionary and a concept ID (cid), returning the "key" associated with that specific concept ID within the data tree.

**Usage**

```
get_key(dd, cid)
```

**Arguments**

dd	A <code>data.tree</code> object representing the data dictionary.
cid	A concept ID as a string or numeric value, used to locate the specific node within the data tree whose key is desired.

**Value**

The key associated with the given concept ID. If the concept ID does not exist, returns `NULL`.

**Examples**

```
# Assuming `dd` is a properly structured `data.tree` object
# and '123' is a valid concept ID in the tree:
path <- get_path_to_example_dictionary() # Replace with path to your dictionary
dd <- construct_dictionary_tree(path)
key <- get_key(dd, '151488193')
key
```

---

get\_meta

---

*Retrieve Metadata for a Concept*


---

**Description**

This function retrieves metadata for a concept based on whether it's a valid concept ID (cid) or key.

**Usage**

```
get_meta(dd, concept)
```

**Arguments**

dd	A <code>data.tree</code> object representing the data dictionary.
concept	The concept ID or key to find metadata for.

**Value**

A dataframe containing metadata for the given concept, or `NULL` if not found.

**Examples**

```
# Assuming `dd` is a properly structured `data.tree` object:
path <- get_path_to_example_dictionary() # Replace with path to your dictionary
dd <- construct_dictionary_tree(path)
metadata <- get_meta(dd, '317567178') # for a valid cid
metadata
metadata <- get_meta(dd, 'In the past month') # for a valid key
metadata
```

---

get_path_to_example_dictionary	<i>Get Path to an example dictionary</i>
--------------------------------	--

---

**Description**

Get Path to an example dictionary

**Usage**

```
get_path_to_example_dictionary()
```

**Value**

string Path to a folder containing JSON files for a example dictionary

**Examples**

```
path <- get_path_to_example_dictionary()
print(path)
```

---

get_responses	<i>Retrieve Responses for a Concept</i>
---------------	---

---

**Description**

This function retrieves responses associated with a given concept.

**Usage**

```
get_responses(dd, concept)
```

**Arguments**

dd	A data.tree object representing the data dictionary.
concept	The concept ID or key whose responses are to be retrieved.

**Value**

Responses associated with the concept if it is a question; otherwise, NULL.

**Examples**

```
path <- get_path_to_example_dictionary() # Replace with path to your dictionary
dd <- construct_dictionary_tree(path)
responses <- get_responses(dd, '763164658')
responses
responses <- get_responses(dd, 'How many cigarettes have you smoked in your entire life?')
responses
```

---

get_var_name	<i>Retrieve Variable Name for a Question Concept</i>
--------------	--

---

### Description

This function retrieves the variable name for a question concept, given a string containing it's Concept ID. This is particularly useful for labeling data from Connect's BigQuery tables.

### Usage

```
get_var_name(dd, cid_str)
```

### Arguments

dd	The data dictionary tree
cid_str	A string containing the concept id for a question concept.

### Value

The variable name for the question concept.

### Examples

```
path <- get_path_to_example_dictionary() # Replace with path to your dictionary
dd <- construct_dictionary_tree(path)
var_name <- get_var_name(dd, "d_142654897_d_461488577") # Should return "RcrtES_Aware_v1r0_Email"
var_name
```

---

is_valid_cid	<i>Validate if the Input is a 9-digit Concept ID (cid)</i>
--------------	--

---

### Description

This function checks if a given input is a valid 9-digit concept ID. Returns TRUE if valid, FALSE otherwise.

### Usage

```
is_valid_cid(input)
```

### Arguments

input	The input to check, expected to be a string or numeric type.
-------	--

### Value

A boolean indicating if the input is a valid 9-digit concept ID.

**Examples**

```
is_valid_cid(123456789)    # Should return TRUE
is_valid_cid("987654321")  # Should return TRUE
is_valid_cid(12345)        # Should return FALSE
is_valid_cid(9876543210)   # Should return FALSE
```

# Index

`construct_dictionary_tree`, [2](#)  
`extract_cids`, [2](#)  
`get_cid`, [3](#)  
`get_key`, [3](#)  
`get_meta`, [4](#)  
`get_path_to_example_dictionary`, [5](#)  
`get_responses`, [5](#)  
`get_var_name`, [6](#)  
`is_valid_cid`, [6](#)