

Erklärung des Codes

Das Ziel dieses Codes ist es, einen hierarchischen Baum aus Daten zu erstellen, die in einem Pandas DataFrame gespeichert sind. Jede Zeile im DataFrame stellt einen Knoten auf einer Ebene der Hierarchie dar, und der Baum wird gemäß den Spalten Root, Branch, Sub-Branch, Leaf, Subleaf und URL aufgebaut. Hier ist eine detaillierte Erklärung des Codes:

Klasse `Node`

Die Klasse `Node` repräsentiert einen Knoten im Baum.

python

```
class Node:
    def __init__(self, name, url=None):
        self.name = name # Name des Knotens
        self.url = url # Dem Knoten zugeordnete URL, falls vorhanden
        self.children = [] # Liste der Kinderknoten

    def add_child(self, child_node):
        self.children.append(child_node) # Fügt einen Kinderknoten zur Liste (
```

- `__init__(self, name, url=None)`: Initialisiert einen neuen Knoten mit einem Namen und einer optionalen URL. Initialisiert auch eine leere Liste zur Speicherung der Kinderknoten.
- `add_child(self, child_node)`: Fügt einen Kinderknoten zur Liste der Kinder des aktuellen Knotens hinzu.

Funktion `build_tree_from_df`

Diese Funktion baut den Baum aus einem Pandas DataFrame auf.

python

```
def build_tree_from_df(df):
    roots = {} # Dictionary zur Speicherung der Wurzelknoten
    for _, row in df.iterrows():
        root_name = row['Root'] # Name der Wurzel
        branch_name = row['Branch'] # Name des Zweiges
        sub_branch_name = row['Sub-Branch'] # Name des Unterzweiges
        leaf_name = row['Leaf'] # Name des Blattes
        subleaf_name = row['Subleaf'] # Name des Unterblattes
        url = row['URL'] # Dem Knoten zugeordnete URL
```