However, I can provide a brief overview of the imported classes and their commonly used methods:

1. FlatTreeControl: This class is used to control a flat tree, i.e., a tree with an underlying flat list of nodes. Some of its methods include:
   * expand: Expands a given data node in the tree.
   * collapse: Collapses a given data node in the tree.
   * getLevel: Gets the level of a given tree node.
2. MatTreeFlatDataSource: This class is a data source for a flat list that is used in conjunction with a MatTreeFlattener. Some of its methods include:
   * connect: Connects a collection viewer (such as a MatTree) to this data source.
   * disconnect: Disconnects a collection viewer from this data source.
3. MatTreeFlattener: This class is used to convert a normal tree (i.e., a hierarchical tree structure) into a flat tree. Some of its methods include:
   * flattenNode: Flattens a particular node in the tree.
   * expandLevel: Expands the level of a particular node in the tree.
4. MatPaginator: This class provides a paginator for a MatTable. Some of its methods include:
   * nextPage: Advances to the next page.
   * previousPage: Moves to the previous page.
   * firstPage: Moves to the first page.
   * lastPage: Moves to the last page.
5. MatSort: This class provides sorting functionality for a MatTable. Some of its methods include:
   * sort: Sorts the data based on a sort header id and a direction (asc or desc).

The FoodFlatNode class currently does not have any methods. It only has properties name, count, children, and text.

Here's a brief explanation of the methods in this class:

1. constructor(): It initializes the dataSource with the buildFileTree method and sets the pageSize.
2. ngOnInit(): Here, it sets the length and page size of the paginator, and sorts the columns if sorting is provided.
3. hasChild = (\_: number, node: FoodFlatNode) => node.expandable;: This method checks if a node is expandable.
4. buildFileTree(obj: any[], level: string): FoodFlatNode[]: This method builds the file structure tree. It takes an array of objects and a level as parameters and returns an array of FoodFlatNode.
5. onPageChange(event: MatPaginator | any): This method handles page changes in the paginator. It updates the pageSize and the dataSource.data.
6. sortHeaders(col: string, order?: string): This method sorts the headers based on the column and order provided. It sorts the data in ascending or descending order.
7. getColumns(col: string): This method returns the column with the given id.
8. filterChanged(event: any): This method is called when the filter changes. It filters the data and expands or collapses all nodes based on the filter text.
9. filter(filterText: string): This method filters the data based on the filter text provided. It builds the tree nodes from the filtered data and notifies the change.

Please note that this is a high-level explanation. The actual behavior of these methods depends on the data they are working with and how they are used in the component's template.