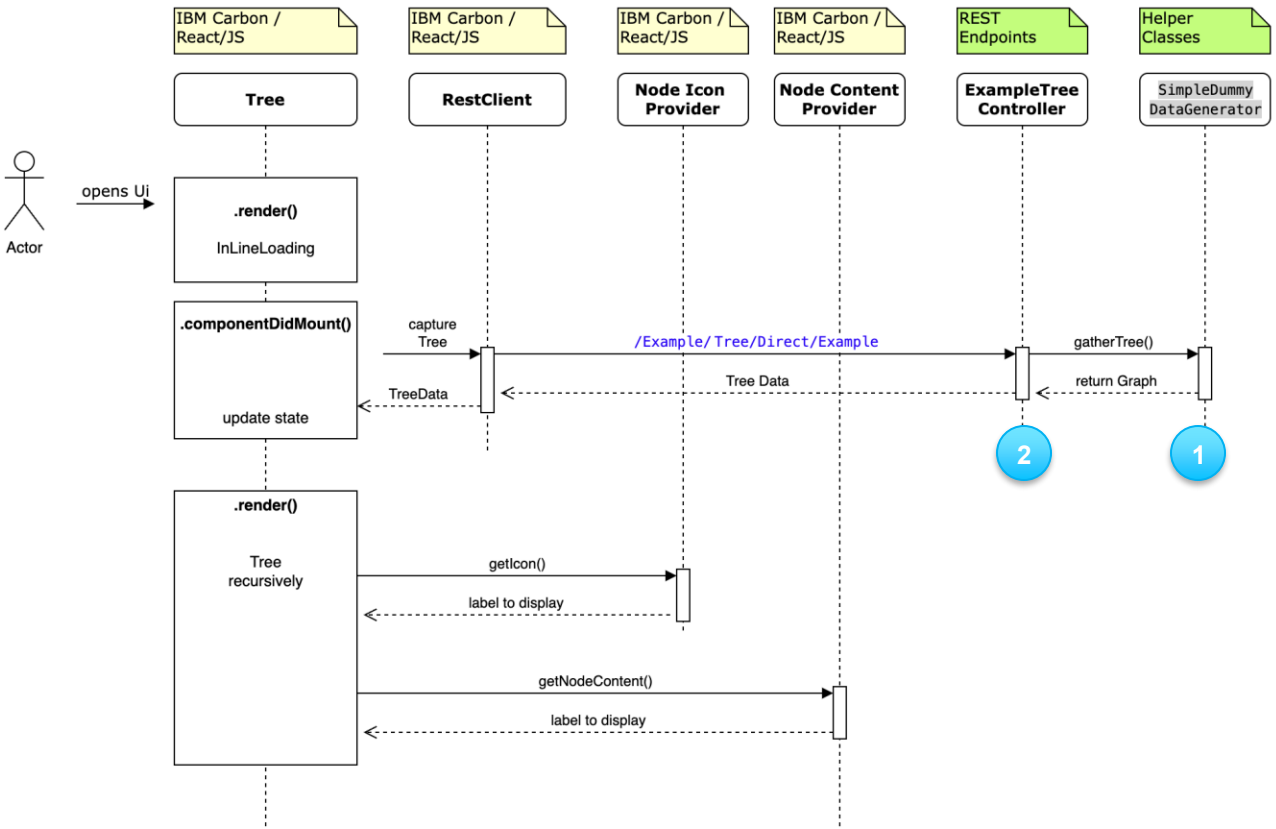
IBM Carbon TreeView

Example 04

Tree (filled by Backend-List)

GET /api/carbonplay/Example/Tree/Direct/Example

Sequence Diagram



Server-sided Sources

1

- restserver.objects.tree.generator
  - DataGeneratorForDirectory.java
  - SimpleDummyDataGenerator.java
  - SimpleDummyDataGeneratorViaList.java

2

- restserver.controller
  - AppConfigController.java
  - AppI18nController.java
  - AppUiParameterController.java
  - ExampleTreeController.java

1

Create the Pojos

```
// ==========================================
// a) create Nodes of Pojos
// ==========================================

PojoJGraphtTreeNode pojoJGraphtTreeNode01 = createNode("A_id",  "A");
PojoJGraphtTreeNode pojoJGraphtTreeNode02 = createNode("B_id",  "B");
PojoJGraphtTreeNode pojoJGraphtTreeNode03 = createNode("A1_id", "A1");
PojoJGraphtTreeNode pojoJGraphtTreeNode04 = createNode("A2_id", "A2");
PojoJGraphtTreeNode pojoJGraphtTreeNode05 = createNode("B1_id", "B1");
PojoJGraphtTreeNode pojoJGraphtTreeNode06 = createNode("B2_id", "B2");
PojoJGraphtTreeNode pojoJGraphtTreeNode07 = createNode("x_id",  "x");

PojoJGraphtTreeNode pojoJGraphtTreeNode08 = createNode("y_id",  "y"); //these 2 nodes:  create the same payload (entityID = "y"),
PojoJGraphtTreeNode pojoJGraphtTreeNode09 = createNode("y_id",  "y"); //but for different TreeNode-iDs (to add "y_label" 1x to "A2" and  1x to "B1")

PojoJGraphtTreeNode pojoJGraphtTreeNode10 = createNode("ya_id", "ya");
PojoJGraphtTreeNode pojoJGraphtTreeNode11 = createNode("yb_id", "yb");
PojoJGraphtTreeNode pojoJGraphtTreeNode12 = createNode("za_id", "za");
PojoJGraphtTreeNode pojoJGraphtTreeNode13 = createNode("zb_id", "zb");
PojoJGraphtTreeNode pojoJGraphtTreeNode14 = createNode("zc_id", "zc");
PojoJGraphtTreeNode pojoJGraphtTreeNode15 = createNode("C_id",  "C");
PojoJGraphtTreeNode pojoJGraphtTreeNode16 = createNode("D_id",  "D");
```

Fill Graph

Vertex & Edges

```
// ==========================================
// b) create tree as graph (Vertices & Edges)
// ==========================================

SimpleDirectedGraph<PojoJGraphtTreeNode, DefaultEdge> graph = new SimpleDirectedGraph<>(DefaultEdge.class);

graph.addVertex(pojoJGraphtTreeNode01);
graph.addVertex(pojoJGraphtTreeNode02);
graph.addVertex(pojoJGraphtTreeNode03);
graph.addVertex(pojoJGraphtTreeNode04);
graph.addVertex(pojoJGraphtTreeNode05);
graph.addVertex(pojoJGraphtTreeNode06);
graph.addVertex(pojoJGraphtTreeNode07);
graph.addVertex(pojoJGraphtTreeNode08);
graph.addVertex(pojoJGraphtTreeNode09);
graph.addVertex(pojoJGraphtTreeNode10);
graph.addVertex(pojoJGraphtTreeNode11);
graph.addVertex(pojoJGraphtTreeNode12);
graph.addVertex(pojoJGraphtTreeNode13);
graph.addVertex(pojoJGraphtTreeNode14);
graph.addVertex(pojoJGraphtTreeNode15);
graph.addVertex(pojoJGraphtTreeNode16);

graph.addEdge(pojoJGraphtTreeNode03, pojoJGraphtTreeNode01);    // A1 ==> A
graph.addEdge(pojoJGraphtTreeNode04, pojoJGraphtTreeNode01);    // A2 ==> A

graph.addEdge(pojoJGraphtTreeNode05, pojoJGraphtTreeNode02);    // B1 ==> B
graph.addEdge(pojoJGraphtTreeNode06, pojoJGraphtTreeNode02);    // B2 ==> B

graph.addEdge(pojoJGraphtTreeNode13, pojoJGraphtTreeNode03);    // zb ==> A1
graph.addEdge(pojoJGraphtTreeNode14, pojoJGraphtTreeNode03);    // zc ==> A1

graph.addEdge(pojoJGraphtTreeNode07, pojoJGraphtTreeNode04);    // x  ==> A2
graph.addEdge(pojoJGraphtTreeNode08, pojoJGraphtTreeNode04);    // y  ==> A2

graph.addEdge(pojoJGraphtTreeNode09, pojoJGraphtTreeNode05);    // y (double) ==> B1

graph.addEdge(pojoJGraphtTreeNode10, pojoJGraphtTreeNode06);    // y   ==> B2
graph.addEdge(pojoJGraphtTreeNode11, pojoJGraphtTreeNode06);    // yb  ==> B2
graph.addEdge(pojoJGraphtTreeNode12, pojoJGraphtTreeNode06);    // za  ==> B2
```

Calculate ParentID and Children[]

```
// ==========================================
// c) calculate parent and children
// ==========================================

JGraphTParentChildrenPopulator.calcParentIDAndChildrens(graph);

return graph;
```

1

Create Payload & UiControl

Create Node

```
/**
 * This method creates a Graph-Vertex ( = PojoJGraphtTreeNode ) which holds a payload with entity - information
 * @param objectId ID of the Object
 * @param objectLabel Label of the Object
 * @return
 */
private static PojoJGraphtTreeNode createNode(String objectId, String objectLabel) {

    //What is an entity ... PojoEntityPayload is the payload, which answers this question
    PojoSimpleObjectPayload pojoObjectPayload = new PojoSimpleObjectPayload();
    pojoObjectPayload.setObjectID(objectId);
    pojoObjectPayload.setObjectLabel(objectLabel);

    PojoPayloadUiControl pojoPayloadUiControl = new PojoPayloadUiControl();

    PojoJGraphtTreeNode pojoTreeNode = new PojoJGraphtTreeNode();
    pojoTreeNode.setLabel(objectLabel);
    pojoTreeNode.setPayload(pojoObjectPayload);
    pojoTreeNode.setUicontrol(pojoPayloadUiControl);

    return pojoTreeNode;
}
```

2

```
//========== >

//get data as Graph without any edges
SimpleDirectedGraph<PojoJGraphtTreeNode, DefaultEdge> graph = SimpleDummyDataGenerator.gatherTree();

//identify root-nodes
PojoJGraphtTreeNode[] array = JGraphTListToTreeTransformer.getTreeRootNodes(graph);

//sort array by label
Arrays.sort(array, Comparator.comparing(node -> node.getLabel()));

//<==========

result =  new ResponseEntity<>(array,HttpStatus.OK);
return result;
```

Client-sided Sources