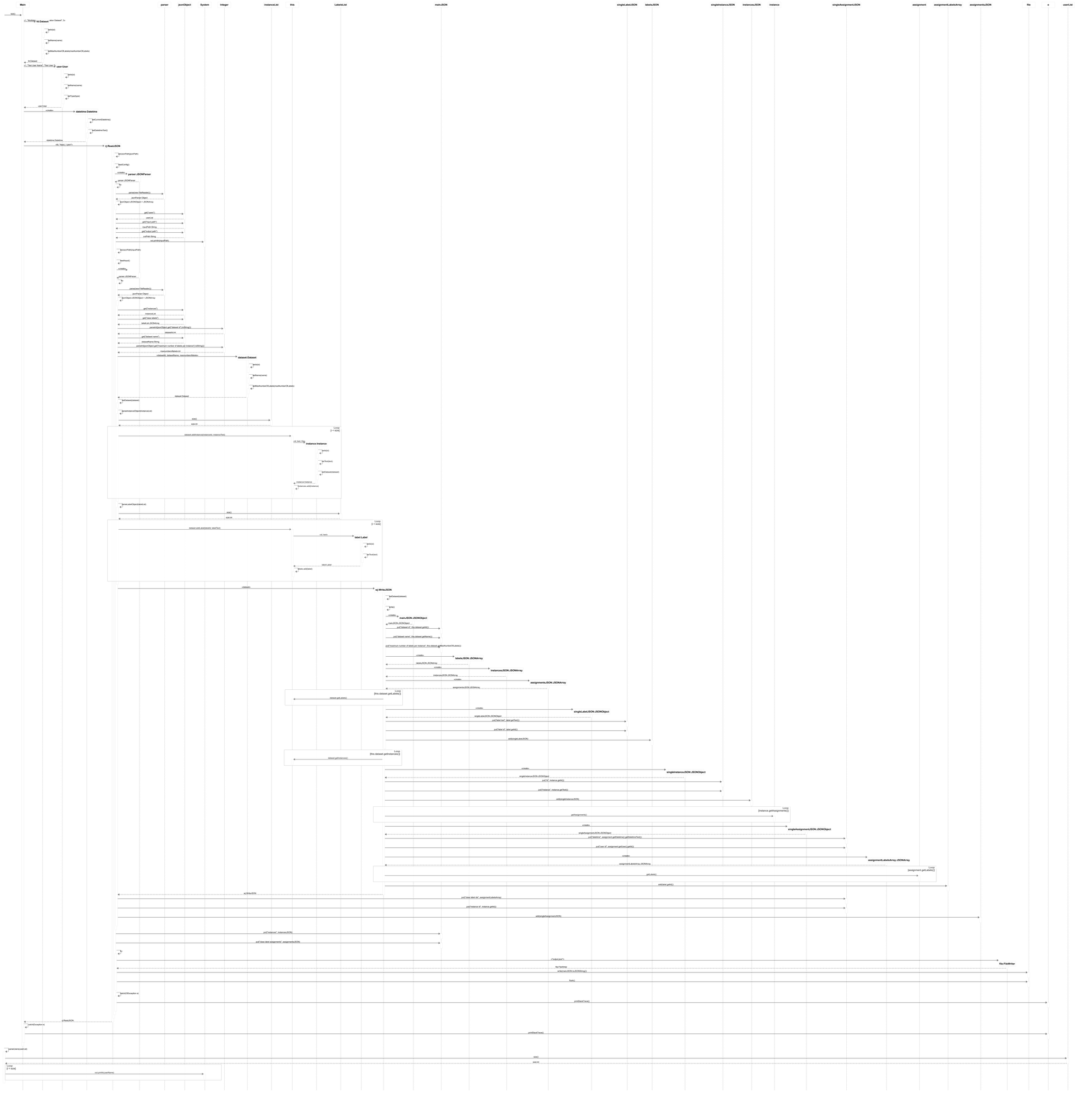
Main
parser
jsonObject
System
Integer
instanceList
this
LabelsList
mainJSON
singleLabelJSON
labelsJSON
singleInstanceJSON
instancesJSON
Instance
singleAssignmentsJSON
assignment
assignmentLabelsArray
assignmentsJSON
file
e
userList
sys.argv[1]
Main()
["-f", "filepath"]
ts: Dataset
"Dataset" in
getId()
getName(name)
getNumberOfLabelsInsteadOfLabels()
ts:Dataset
["-l", "-l", "User Name", "-l", "User Email"]
user: User
getId()
getName(name)
getUserType()
user:User
<create>
datetime:Datetime
getCurrentDatetime()
getDatetimeText()
datetime:Datetime
obj, "logs/_1.jsonl"
c: ReadJSON
getJsonPathJsonPath()
getConfig()
<create>
parser:JSONParser
parser:JSONParser
parse(new FileReader())
jsonParser:Object
[jsonObject:JSONObject = (JSONArray
getUser()
user:int
getInput path()
inputPath:String
getOutput path()
outPath:String
out.println(inputPath)
getJsonPathJsonPath()
getFilesId()
<create>
parser:JSONParser
parser:JSONParser
parse(new FileReader())
jsonParser:Object
[jsonObject:JSONObject = (JSONArray
getInstances()
instances:int
getClass.length()
labels:int:JSONArray
parseint(jsonObject.get("dataset id").toString())
datasetId:int
getDataset name()
datasetName:String
parseint(jsonObject.get("maximum number of labels per instance").toString())
maxInstanceNumber:int
<create>(datasetId, datasetName, maxInstanceNumber)
dataset:Dataset
getId()
getName(name)
getNumberOfLabelsInsteadOfLabels()
dataset:Dataset
getDataset(dataset)
getInstances()
[anotherInstanceCheckInstance:int]
size()
size:int
Loop
[i = size]
dataset.addInstance(instanceS_instanceText)
<create>
instance:Instance
getId()
getText(text)
getDataset(dataset)
instance:Instance
instances.add(instance)
[arrayLabelObjectsLabel:int]
size()
size:int
Loop
[i = size]
dataset.addLabel(labelS_labelText)
<create>
label:Label
getId()
getText(text)
label:Label
arrayLabel.add(label)
<create>
w: WriteJSON
getJsonset(dataset)
getJsonset()
<create>
mainJSON:JSONObject
mainJSON:JSONObject
put("dataset id", this.dataset.getId())
put("dataset name", this.dataset.getName())
put("maximum number of labels per instance", this.dataset.getNumberOfLabelsInsteadOfLabels())
<create>
labelsJSON:JSONArray
labelsJSON:JSONArray
<create>
instancesJSON:JSONArray
instancesJSON:JSONArray
<create>
assignmentsJSON:JSONArray
assignmentsJSON:JSONArray
Loop
[this.dataset.getLabels()]
dataset.getLabels()
<create>
singleLabelJSON:JSONObject
singleLabelJSON:JSONObject
put("label text", label.getText())
put("label id", label.getId())
add(singleLabelJSON)
Loop
[this.dataset.getInstances()]
dataset.getInstances()
<create>
singleInstanceJSON:JSONObject
singleInstanceJSON:JSONObject
put("id", instance.getId())
put("instance", instance.getText())
add(singleInstanceJSON)
Loop
[instance.getAssignments()]
getAssignments()
<create>
singleAssignmentJSON:JSONObject
singleAssignmentJSON:JSONObject
put("instance", assignment.getInstance().getId())
put("user id", assignment.getUser().getId())
<create>
assignmentLabelsArray:JSONArray
assignmentLabelsArray:JSONArray
Loop
[assignment.getLabels()]
getLabels()
add(label.getId())
w.Write()
put("class label id", assignment.getLabels())
put("instance id", instance.getId())
add(singleAssignment)
put("instances", instancesJSON)
put("class label assignments", assignmentsJSON)
try
"./Output.json"
fw:FileWriter
fw.FileWrite
write(mainJSON.toJSONString())
flush()
catch(IOException e)
printStackTrace()
c:ReadJSON
catch(Exception e)
printStackTrace()
System.exit(1)
exit()
exit:int
Loop
[i < size]
out.println(userNames)