Path finders;

**public** **class** ParamPathExpressionParser {

**public** **static** **final** String KEY\_PREFIX = "<!";

**public** **static** **final** String KEY\_SUFFIX = "!>";

**public** **static** Map<Integer, String> parse(String in) {

Map<Integer, String> entries = **new** LinkedHashMap<>();

parse(in, 0, entries);

**return** entries;

}

**private** **static** **void** parse(String in, **int** startPos, Map<Integer, String> entries) {

**int** iStart = StringUtils.indexOf(in, KEY\_PREFIX, startPos);

**if**(iStart==StringUtils.INDEX\_NOT\_FOUND)

**return**;

**int** iEnd = findEndRecursive(in, iStart);

String key = StringUtils.substring(in, iStart, iEnd+KEY\_SUFFIX.length());

entries.put(iStart, key);

parse(in, iEnd, entries);

}

**private** **static** **int** findEndRecursive(String in, **int** iStart) {

**int** iEnd = StringUtils.indexOf(in, KEY\_SUFFIX, iStart);

**if**(iEnd==StringUtils.INDEX\_NOT\_FOUND)

**throw** **new** InvalidConfigException("Found config url entry with starting '"+KEY\_PREFIX+"' but no closing '"+KEY\_SUFFIX+"' in "+ in);

// check recursive

**return** countRecursive(in, iStart, iEnd);

}

**private** **static** **int** countRecursive(String in, **int** iStart, **int** iFirstEnd) {

// check if there are any starts

String subStrBetweenFirstStartAndFirstEnd = in.substring(iStart+1, iFirstEnd);

**int** countStartsInBetween = StringUtils.countMatches(subStrBetweenFirstStartAndFirstEnd, KEY\_PREFIX);

**if**(countStartsInBetween==0) {

**return** iFirstEnd;

}

// find next end after current end

**int** iNextStart = StringUtils.indexOf(in, KEY\_PREFIX, iStart+1);

**int** nextEnd = StringUtils.indexOf(in, KEY\_SUFFIX, iFirstEnd+1);

**int** recursiveNextEnd = countRecursive(in, iNextStart, nextEnd);

**if**(nextEnd == recursiveNextEnd)

**return** nextEnd;

**return** countRecursive(in, nextEnd, recursiveNextEnd);

}

**public** **static** String stripPrefixSuffix(String in) {

**return** StringUtils.removeEnd(

StringUtils.removeStart(in, KEY\_PREFIX), KEY\_SUFFIX);

}

**public** **static** **boolean** containsPrefixSuffix(String in) {

**int** start = StringUtils.indexOf(in, KEY\_PREFIX);

**int** end = StringUtils.indexOf(in, KEY\_SUFFIX, start);

**return** (start!=StringUtils.INDEX\_NOT\_FOUND && end!=StringUtils.INDEX\_NOT\_FOUND);

}

}

--

**protected** String mapColElem(Param<?> commandParam, String pathToResolve) {

// check if command param is colElem

**if**(commandParam.isCollectionElem())

**return** commandParam.findIfCollectionElem().getElemId();

// otherwise, if mapped, check if mapsTo param is colElem

**if**(commandParam.isMapped())

**return** mapColElem(commandParam.findIfMapped().getMapsTo(), pathToResolve);

// throw ex ..or.. blank??

**return** "";

}