Atitit java解析js ast 作为流程控制ast pcl

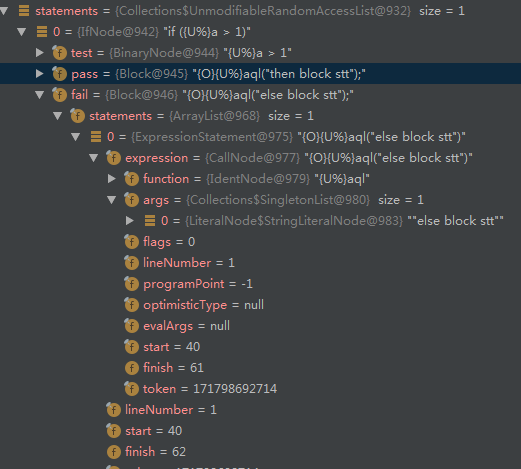
目录

[1.1. Parser 解析 1](#_Toc20083)

[1.2. 第二只方法 3](#_Toc17014)

## Parser 解析

package attilax.js;  
  
import jdk.nashorn.internal.ir.Block;  
import jdk.nashorn.internal.ir.FunctionNode;  
import jdk.nashorn.internal.ir.Statement;  
import jdk.nashorn.internal.parser.Parser;  
import jdk.nashorn.internal.runtime.Context;  
import jdk.nashorn.internal.runtime.ErrorManager;  
import jdk.nashorn.internal.runtime.Source;  
import jdk.nashorn.internal.runtime.options.Options;  
  
import java.util.List;  
  
public class jsAST {  
  
 public static void main(String[] args) {  
 Options options = new Options("nashorn");  
 options.set("anon.functions", true);  
 options.set("parse.only", true);  
 options.set("scripting", true);  
  
 ErrorManager errors = new ErrorManager();  
 Context context = new Context(options, errors, Thread.*currentThread*().getContextClassLoader());  
// Source source = new Source("test", "var a = 10; var b = a + 1;" +  
// "function someFunction() { return b + 1; } ");  
 String js\_content = "var a = 10; var b = a + 1;" +  
 "function someFunction() { return b + 1; } ";  
 js\_content="if(a>1){ aql('aaa');} else{ aql('bbb'); }";  
 Source source = Source.*sourceFor*("test", js\_content);  
 Parser parser = new Parser(context.getEnv(), source, errors);  
 FunctionNode functionNode = parser.parse();  
 Block block = functionNode.getBody();  
 List<Statement> statements = block.getStatements();  
 System.*out*.println(statements);  
 }  
}



package jdk.nashorn.internal.ir;  
  
import jdk.nashorn.internal.ir.annotations.Immutable;  
import jdk.nashorn.internal.ir.visitor.NodeVisitor;  
  
@Immutable  
public final class IfNode extends Statement implements JoinPredecessor {  
 private static final long serialVersionUID = 1L;  
 private final Expression test;  
 private final Block pass;  
 private final Block fail;  
 private final LocalVariableConversion conversion;

## 第二只方法

|  |
| --- |
| load("nashorn:parser.js"); |
|  | var ast = parse("function x(n) {var a=[];for(var i=0;i< n;i++) { a.push(i); }} x(1000);"); |
|  | print(JSON.stringify(ast)); |

Atitit 流程控制语言 sql 自定义apcl