

OGNL

Bài này nghiên cứu cách thức OGNL parse một expression.

OGNL ver. 2.7.3

Trước khi OGNL tham gia

Trong struts 2, parameters truyền vào được check valid bởi `ParametersInterceptor` bằng regex.

- Struts 2.0.11 's pattern:

Flow cơ bản như sau:

1. `ParamterInterceptor`: Nhận param, check name nếu valid thì tiếp.
2. `OGNLValueStack.setValue(String expr, Object value, boolean throwExceptionOnFailure)``
3. `OGNLUtil.setValue(String name, Map<String, Object> context, Object root, Object value).`

Tiền xử lý - trả về một object kiểu Node từ expression.

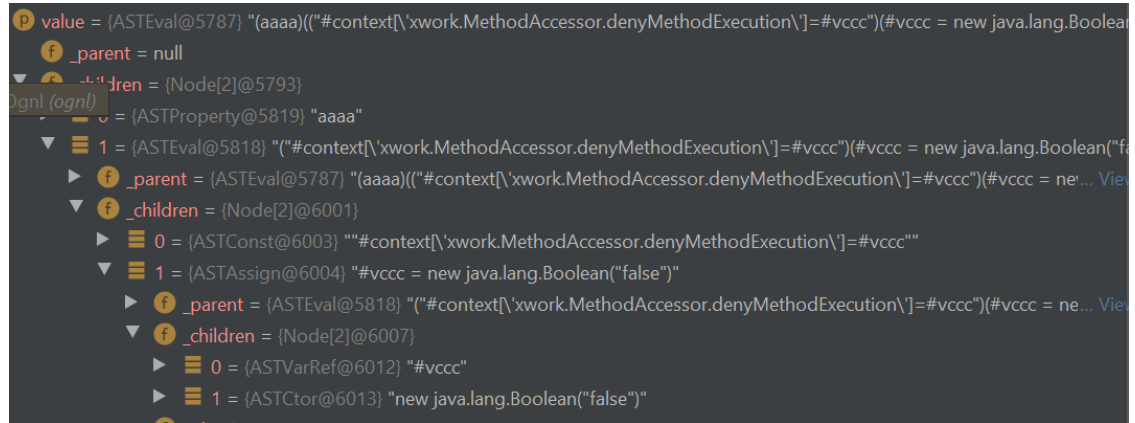
Hàm `OGNLUtil.setValue()` ở trên sẽ gọi `Ognl.setValue(this.compile(name), context, root, value);`

`OGNLUtil.compile(String name)` gọi `OGNL.parseExpression(String expression)` để trả về một object kiểu `Node`, một dạng cấu trúc dữ liệu `Tree`. Gán cho mỗi node một object thuộc `AST<something>`.

1. Ví dụ xử lý một expression sau:

```
1 (aaaa)
  (( '%5Cu0023context%5B%5C'xwork.MethodAccessor.denyMethodExecution%5C'%5D
  %5Cu003d%5Cu0023vccc' )
  (%5Cu0023vccc%5Cu003dnew%20java.lang.Boolean(%22false%22)))
```

Kết quả:



Sau đó, Node tạo ra ở trên được đưa vào hàm `OGNL.setValue(Object tree, Map context, Object root, Object value)` để thực hiện expression.

```
1 | ognl.setValue(this.compile(name), context, root, value);
```

```
1 | public static void setValue(Object tree, Map context, Object root,
  | Object value) throws OgnlException {
2 |     OgnlContext ognlContext = (OgnlContext)addDefaultContext(root,
  | context);
3 |     Node n = (Node)tree;
4 |     if (n.getAccessor() != null) {
5 |         n.getAccessor().set(ognlContext, root, value);
6 |     } else {
7 |         n.setValue(ognlContext, root, value);
8 |     }
9 | }
```

Thực hiện expression

AST<something>

Mỗi AST<something> extends SimpleNode class.

Có 2 loại:

1. ASTProperty, ASTMethod: method `setValueBody()` gọi `OGNLRuntime.<methods>`
2. ASTEval, ASTChain: recursively gọi các method dưới

Các methods:

- `evaluateSetBody()`: chuẩn bị context để gọi `setValueBody()`
- `AST[...].setValueBody()`
- `setValue(OGNLContext, root, value)`
- `getValue()`

Đối với AST loại 2, những hàm trên được thực hiện một cách recursively để xác định:

- target (Dành cho ASTProperty): object để set property
- value:
- source (Dành cho ASTMethod): Class chứa static method gọi bởi `OGNLRuntime.callMethod()`
- args

Sau đó xác định một AST loại 1 để gọi static method của `OGNLRuntime`. Ví dụ:

-
- ASTProperty.setValueBody():

```
1  protected void setValueBody(OgnlContext context, Object target, Object
   value) throws ognlException {
2      ognlRuntime.setProperty(context, target,
   this.getProperty(context, target), value);
3  }
```

Để set property của `target = value`

Sau đó, `AST<something>` gọi một static method của `OGNLRuntime` class.

OGNLRuntime

- setProperty() -call-> [Something]Accessor.setProperty()

[Something]Accessor class: để set property cho một object.