QueryBuilder and Custom Predicates

AEM
COMMUNITY
BELARUS

Viktor Kadol

AXAMIT

Outline

- What is QueryBuilder & Predicates?
- Predicates from the box.
- Custom predicates.
- Debugging

What is QueryBuilder & Predicates?

AXAMIT

Search In AEM

Common approaches: XPath, SQL2, QueryBuilder

QueryBuilder

```
path=/content
1_property=sling:resourceType
1_property.value=foundation/components/text
1_property.operation=like
```

SQL2

```
select * from [nt:base] as a where [sling:resourceType] like
'foundation/components/text' and isdescendantnode(a, '/content')
```

XPath

```
/jcr:root/content//*[jcr:like(@sling:resourceType, 'foundation/components/text')
```

AXAMIT

What is QueryBuilder?

IS

Syntactic sugar API over XPath that accepts a query description(predicates), create and run an XPath query, optionally filter the result set, and also extract facets, if desired

IS NOT

Query Engine itself

Search optimization tool (index or cache)

Ways to use QueryBuilder

Search Service Default Servlet(/bin/querybuilder.json /bin/querybuilder.feed)

Java API: HashMap

```
Map<String, String> map = new HashMap<String, String>();
map.put("path", "/content");
map.put("type", "cq:Page");
map.put("p.offset", "0"); // same as query.setStart(0) below
map.put("p.limit", "20"); // same as query.setHitsPerPage(20) below
Query query = builder.createQuery(PredicateGroup.create(map), session);
```

HTTP Request Params

```
PredicateGroup pg = PredicateGroup.create(request.getParameterMap())
Query query = builder.createQuery(pg , session);
```

Predicates combination

```
PredicateGroup pg = PredicateGroup.create(request.getParameterMap())
pg.add(new Predicate("path").set("path", "/content"));
pg.add(new Predicate("type").set("type", "cq:Page"));
Query query = builder.createQuery(pg , session);
```

Query Builder Debugger (/libs/cq/search/content/querydebug.html)

What predicate is?

Description for single logical search term or condition.

Exists

```
1_property.operation=exists
1_property=submitted
```

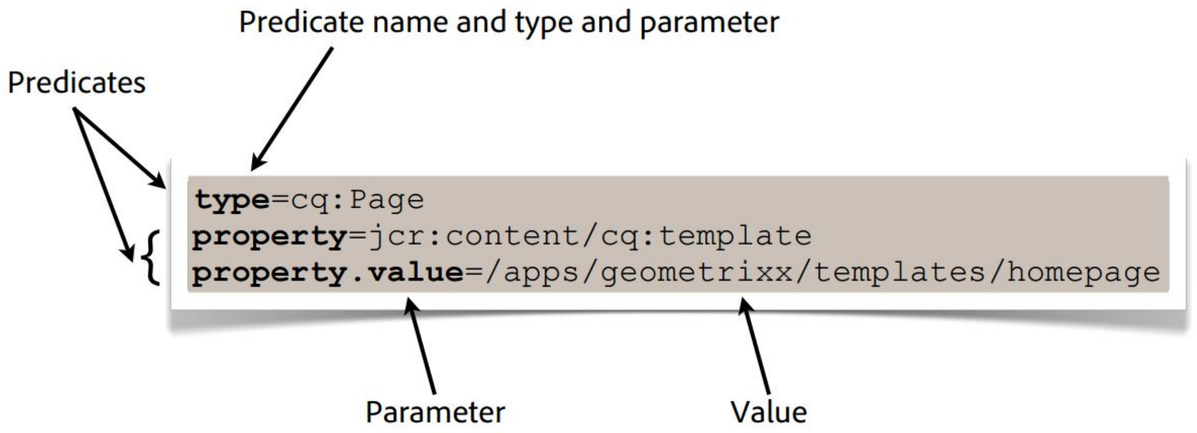
Equals

```
1_property.operation=equals
1_property=jcr:title
1_property.value=Apples
```

Result count:

```
p.limit=-1
```

What predicate is?



```
type=cq:Page
property=jcr:content/cq:template
property.value=/apps/geometrixx/templates/homepage
```

Predicate's type is mirrored as parameter internally:

```
type.type=cq:Page
property.property=jcr:content/cq:template
property.value=/apps/geometrixx/templates/homepage
```

AXAMIT

Predicates from the box.

Most Popular

path : This is used to search under a particular hierarchy only.

- **path.self=true** : If true searches the subtree including the main node given in path, if false searches the subtree only.
- **path.exact=true** : If true exact path is matched, if false all descendants are included.
- **path.flat=true** : If true searches only the direct children .

type: It is used for searching for a particular nodetype only.

property: This is used to search for a specific property only.

- **property.value** : the property value to search .
- **property.depth** : The number of additional levels to search under a node. eg. if property.depth=2 then the property is searched under
- **property.and** : If multiple properties are present , by default an OR operator is applied. If you want an AND , you may use property.and=true
- **property.operation** : "equals" for exact match (default), "unequals" for unequality comparison, "like" for using the jcr:like xpath function , "not" for no match , (value param will be ignored) or "exists" for existence match .(value can be true – property must exist).

Most Popular

fulltext: It is used to search terms for fulltext search

- **fulltext.relPath** : the relative path to search in (eg. property or subnode) eg. fulltext.relPath=jcr:content or fulltext.relPath=jcr:content/@cq:tags

daterange : This predicate is used to search a date property range.

- **daterange.property** : Specify a property which is searched.
- **daterange.lowerBound** : Fix a lower bound eg. 2010-07-25
- **daterange.lowerOperation** : “>” (default) or “>=”
- **daterange.upperBound**: Fix a lower bound eg. 2013-07-26
- **daterange.upperOperation:** “<” (default) or “<=”

relativedaterange: It is an extension of daterange which uses relative offsets to server time. It also supports 1s 2m 3h 4d 5w 6M 7y

- **relativedaterange.lowerBound** : Lower bound offset, default=0
- **relativedaterange.upperBound** : Upper bound Offset .

AXAMIT

Most Popular

nodename: This is used to search exact nodenames for the result set. It allows few wildcards like: nodename=text* will search for any character or no character after text. nodename=text? will search for any character after text.

tagid: This predicate is used to search for a particular tag on a page. You may specify the exact tagid of a tag in this predicate

- **tagid.property**: this may be used to specify the path of node where tags are stored.

group: This predicate is used to create logical conditions in your query. You can create complex conditions using OR & AND operators in different groups. e.g:

```
path=/home/users
type=rep:User
group.1_daterange.property=jcr:created
group.1_daterange.lowerBound=2015-08-18
group.1_daterange.upperBound=2016-08-19
group.2_daterange.property=cq:lastModified
group.2_daterange.lowerBound=2015-08-18
group.2_daterange.upperBound=2016-08-19
group.p.or=true
```

Most Popular

orderBy: This predicate is used to sort the result sets obtained in the query. e.g. orderby=@jcr:score or orderby=@jcr:content/cq:lastModified

- **orderby.sort:** You may define the sorting way for the search results e.g. desc for descending and "" for ascending.
- **orderby:path** : this can also be used to sort by path.

p.hits=full: Use this when you want to return all the properties in a node.Example

p.hits=selective: Use this if you want to return selective properties in search result. Use this with

p.properties=sling:resourceType jcr:primaryType Example

p.nodedepth: Use this when you need properties of a node and its child nodes in the same search result. Use this with p.hits=full Example

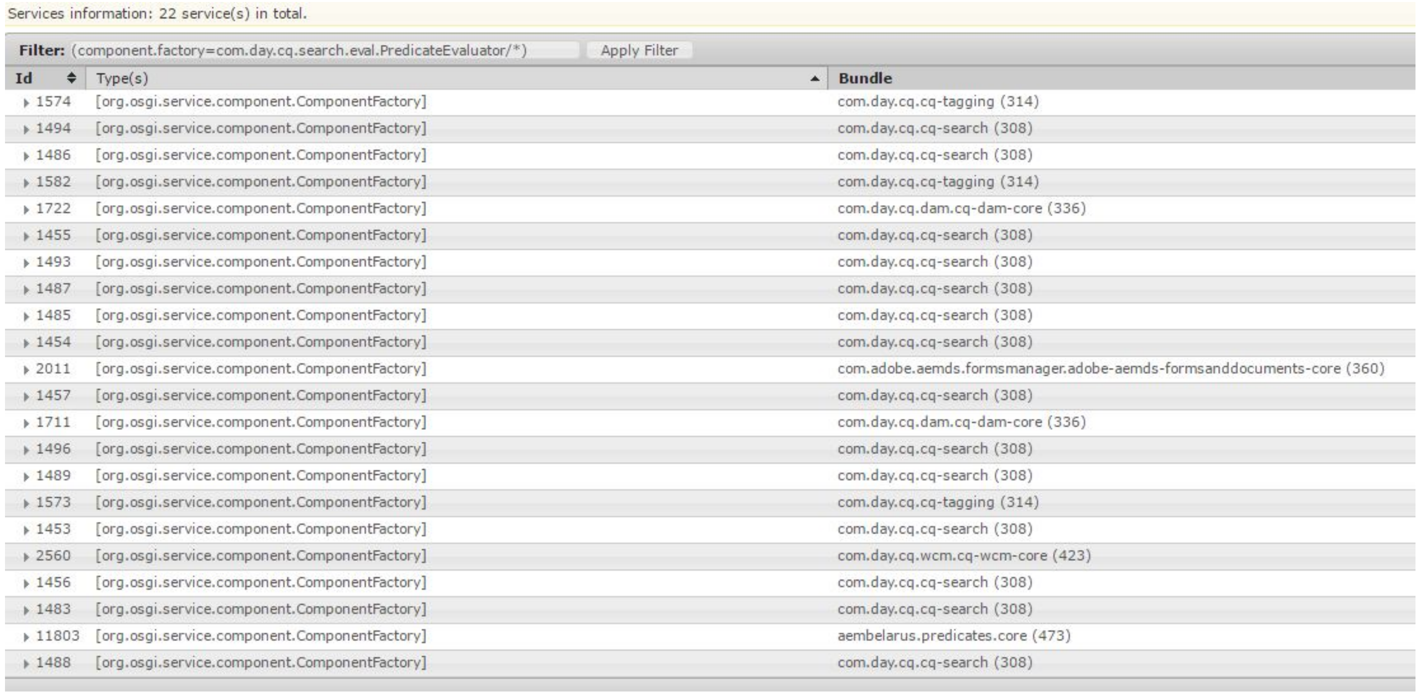
p.facets=true : This will be used to Search Facets based search for the assigned Query. If you want to calculate the count of tags which are present in your search result or you want to know how many templates for a particular page are there etc, you may go with Facets based search . Example

AXAMIT

Gotta Catch ‘Em All !

http://localhost:4502/system/console/services?filter=%28component.factory%3Dcom.day.cq.search.eval.PredicateEvaluator%2F*%29

https://docs.adobe.com/docs/en/aem/6-1/ref/javadoc/com/day/cq/search/eval/PredicateEvaluator.html



Services information: 22 service(s) in total.

Filter: (component.factory=com.day.cq.search.eval.PredicateEvaluator/*) Apply Filter

Id	Type(s)	Bundle
1574	[org.osgi.service.component.ComponentFactory]	com.day.cq.cq-tagging (314)
1494	[org.osgi.service.component.ComponentFactory]	com.day.cq.cq-search (308)
1486	[org.osgi.service.component.ComponentFactory]	com.day.cq.cq-search (308)
1582	[org.osgi.service.component.ComponentFactory]	com.day.cq.cq-tagging (314)
1722	[org.osgi.service.component.ComponentFactory]	com.day.cq.dam.cq-dam-core (336)
1455	[org.osgi.service.component.ComponentFactory]	com.day.cq.cq-search (308)
1493	[org.osgi.service.component.ComponentFactory]	com.day.cq.cq-search (308)
1487	[org.osgi.service.component.ComponentFactory]	com.day.cq.cq-search (308)
1485	[org.osgi.service.component.ComponentFactory]	com.day.cq.cq-search (308)
1454	[org.osgi.service.component.ComponentFactory]	com.day.cq.cq-search (308)
2011	[org.osgi.service.component.ComponentFactory]	com.adobe.aemds.formsmanager.adobe-aemds-formsanddocuments-core (360)
1457	[org.osgi.service.component.ComponentFactory]	com.day.cq.cq-search (308)
1711	[org.osgi.service.component.ComponentFactory]	com.day.cq.dam.cq-dam-core (336)
1496	[org.osgi.service.component.ComponentFactory]	com.day.cq.cq-search (308)
1489	[org.osgi.service.component.ComponentFactory]	com.day.cq.cq-search (308)
1573	[org.osgi.service.component.ComponentFactory]	com.day.cq.cq-tagging (314)
1453	[org.osgi.service.component.ComponentFactory]	com.day.cq.cq-search (308)
2560	[org.osgi.service.component.ComponentFactory]	com.day.cq.wcm.cq-wcm-core (423)
1456	[org.osgi.service.component.ComponentFactory]	com.day.cq.cq-search (308)
1483	[org.osgi.service.component.ComponentFactory]	com.day.cq.cq-search (308)
11803	[org.osgi.service.component.ComponentFactory]	aembelarus.predicates.core (473)
1488	[org.osgi.service.component.ComponentFactory]	com.day.cq.cq-search (308)

Sample Queries

Search for pages tagged with a certain tag

querybuilder

querydebug

Search under multiple paths (using groups)

querybuilder

querydebug

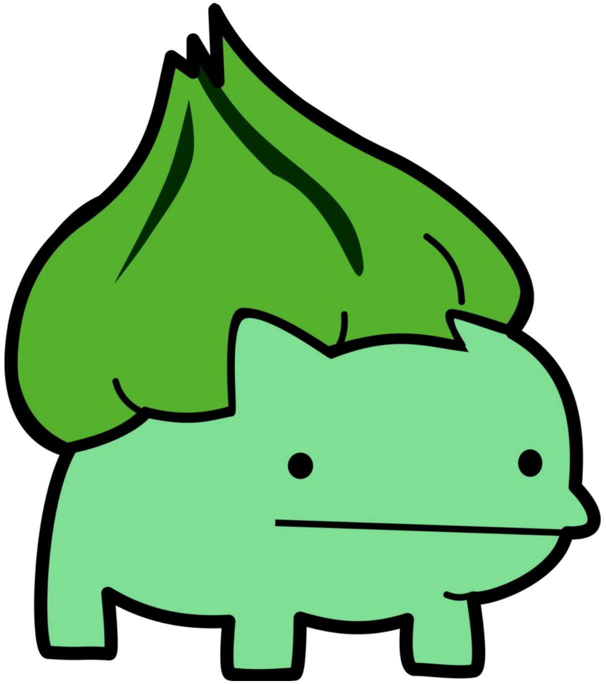
Specify p.hits=full, in which case all properties will be included for each node:

querybuilder

querydebug

AXAMIT

Custom Predicates



AXAMIT

Predicate types

XPath Predicate

This is used to create a Backend XPATH Query using the new custom predicates which can be defined as per need. Many of the inbuilt CQ predicates are XPATH predicates.

Predicate Evaluator

canXpath() - true

canFilter() - false

Filter Predicate

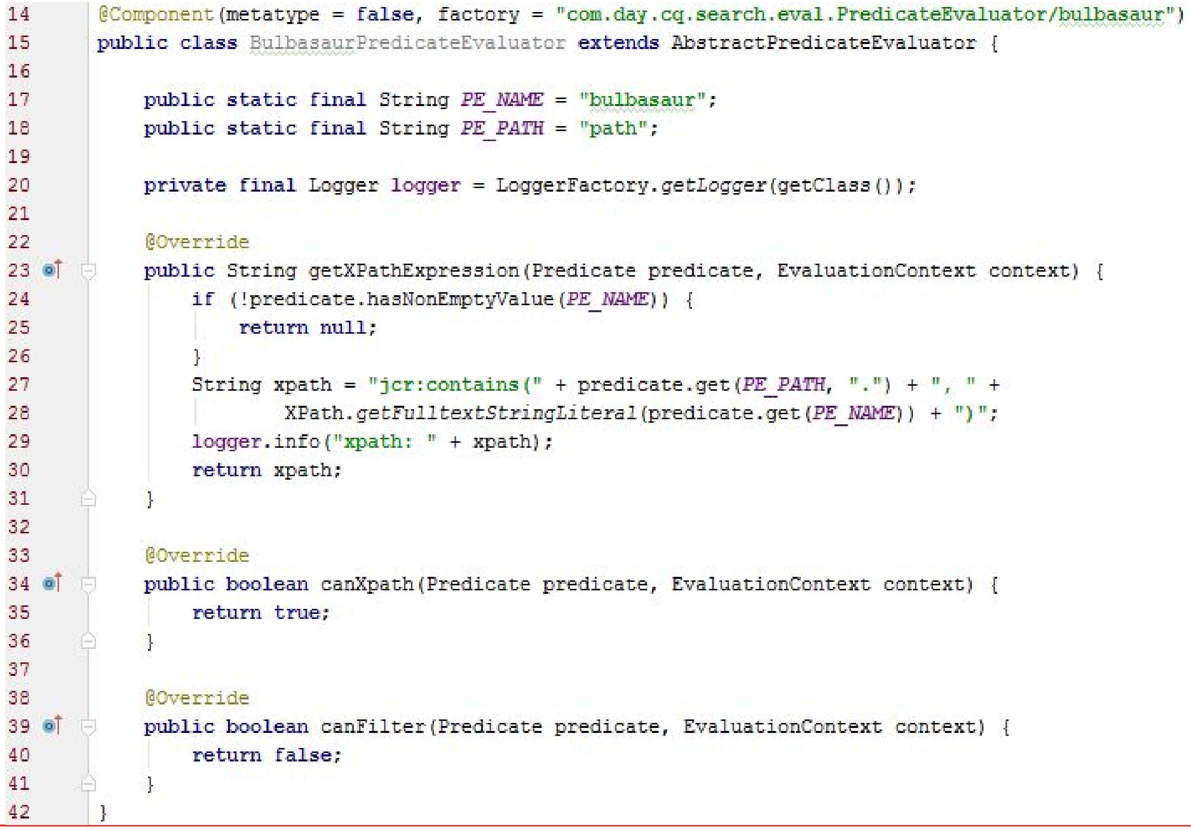
This predicate is used whenever you want to Filter out some results which are not needed in the end Search Result

Predicate Evaluator

canXpath() - false

canFilter() - true

Custom Predicate



```java
@Component(metatype = false, factory = "com.day.cq.search.eval.PredicateEvaluator/bulbasaur")
public class BulbasaurPredicateEvaluator extends AbstractPredicateEvaluator {

    public static final String PE_NAME = "bulbasaur";
    public static final String PE_PATH = "path";

    private final Logger logger = LoggerFactory.getLogger(getClass());

    @Override
    public String getXPathExpression(Predicate predicate, EvaluationContext context) {
        if (!predicate.hasNonEmptyValue(PE_NAME)) {
            return null;
        }
        String xpath = "jcr:contains(" + predicate.get(PE_PATH, ".") + ", " +
                XPath.getFulltextStringLiteral(predicate.get(PE_NAME)) + ")";
        logger.info("xpath: " + xpath);
        return xpath;
    }

    @Override
    public boolean canXpath(Predicate predicate, EvaluationContext context) {
        return true;
    }

    @Override
    public boolean canFilter(Predicate predicate, EvaluationContext context) {
        return false;
    }
}
```

Custom Predicate

Predicate name

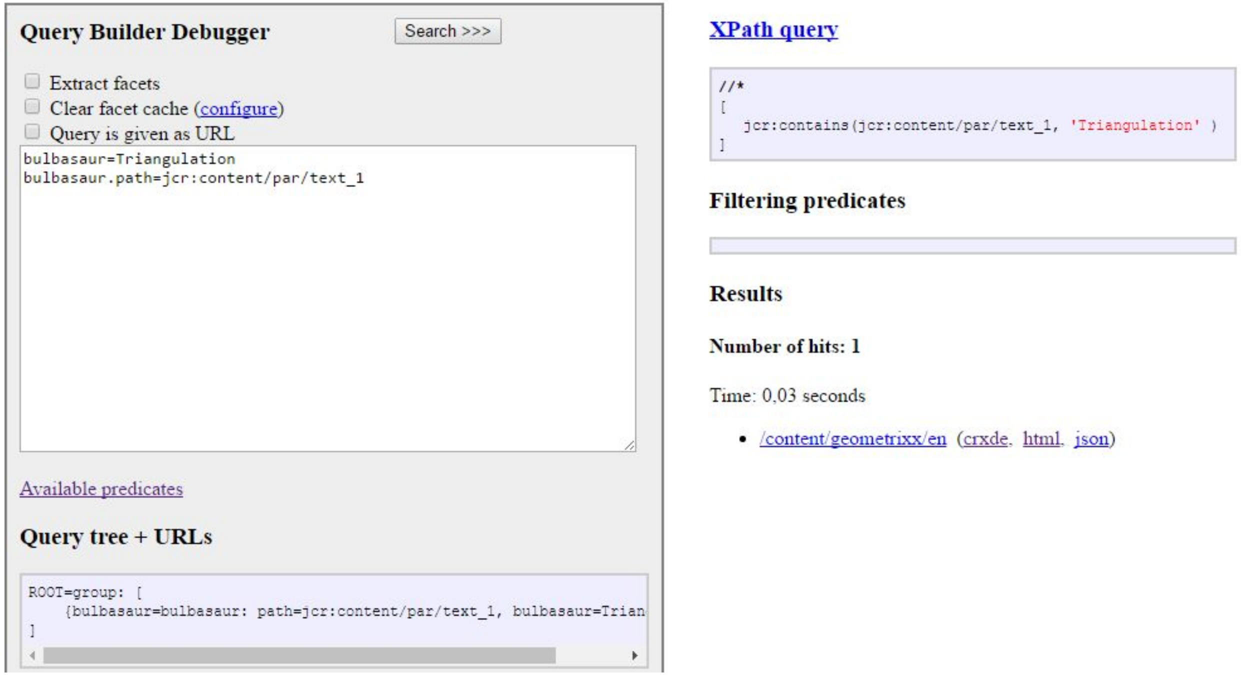
```
@Component(metatype = false, factory = "com.day.cq.search.eval.PredicateEvaluator/bulbasaur")
public class BulbasaurPredicateEvaluator  extends AbstractPredicateEvaluator {
```

Predicate implementation

```
 @Override
   public String getXPathExpression(Predicate predicate, EvaluationContext context) {
        if (!predicate.hasNonEmptyValue( PE_NAME)) {
            return null;
a           }
        String xpath =  "jcr:contains(" + predicate.get(PE_PATH, ".") + ", " +
                XPath. getFulltextStringLiteral(predicate.get(PE_NAME)) + ")";
        logger.info("xpath: " + xpath);
        return xpath;
   }
```

Custom Predicate

querybuilder
querydebugger



Query Builder Debugger Search >>>

- [ ] Extract facets
- [ ] Clear facet cache (configure)
- [ ] Query is given as URL

```
bulbasaur=Triangulation
bulbasaur.path=jcr:content/par/text_1
```

Available predicates

Query tree + URLs

```
ROOT=group: [
    {bulbasaur=bulbasaur: path=jcr:content/par/text_1, bulbasaur=Trian
]
```

XPath query

```
//*
[
   jcr:contains(jcr:content/par/text_1, 'Triangulation' )
]
```

Filtering predicates

Results

Number of hits: 1

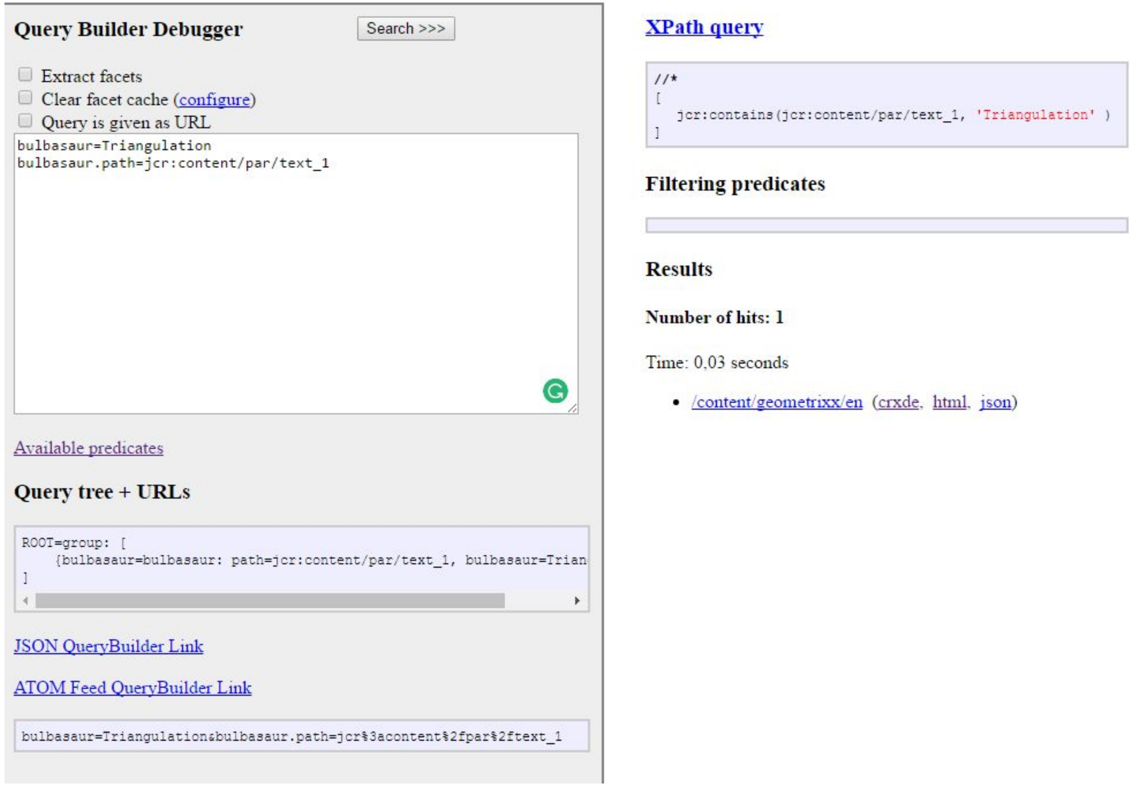
Time: 0,03 seconds

- /content/geometrixx/en (crxde, html, json)

AXAMIT

Debugging

Query Builder Debugger



Query Builder Debugger Search >>>

- [ ] Extract facets
- [ ] Clear facet cache (configure)
- [ ] Query is given as URL

```
bulbasaur=Triangulation
bulbasaur.path=jcr:content/par/text_1
```

Available predicates

Query tree + URLs

```
ROOT=group: [
    {bulbasaur=bulbasaur: path=jcr:content/par/text_1, bulbasaur=Trian
]
```

JSON QueryBuilder Link

ATOM Feed QueryBuilder Link

```
bulbasaur=Triangulation&bulbasaur.path=jcr%3acontent%2fpar%2ftext_1
```

XPath query

```
//*
[
  jcr:contains(jcr:content/par/text_1, 'Triangulation' )
]
```

Filtering predicates

Results

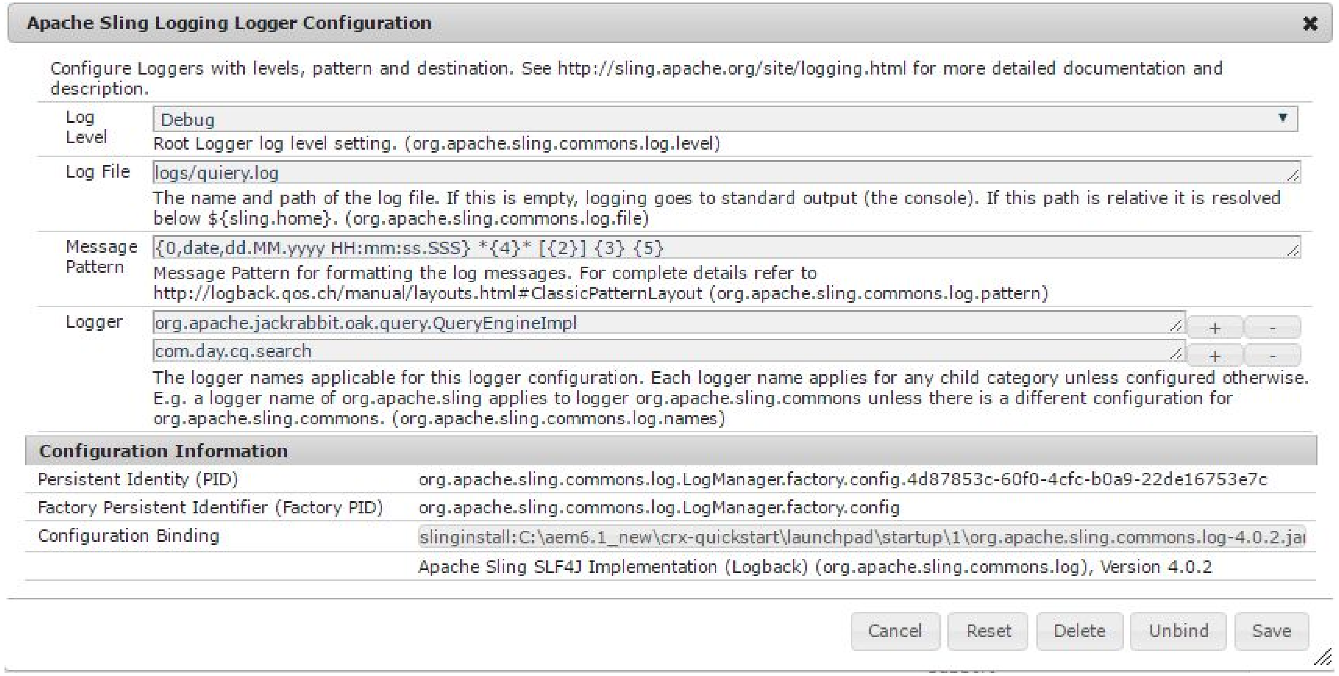
Number of hits: 1

Time: 0,03 seconds

- /content/geometrixx/en (crxde, html, json)

Logs Setup And Monitor

- com.day.cq.search
- org.apache.jackrabbit.oak.query.QueryEngineImpl



Apache Sling Logging Logger Configuration

Configure Loggers with levels, pattern and destination. See http://sling.apache.org/site/logging.html for more detailed documentation and description.

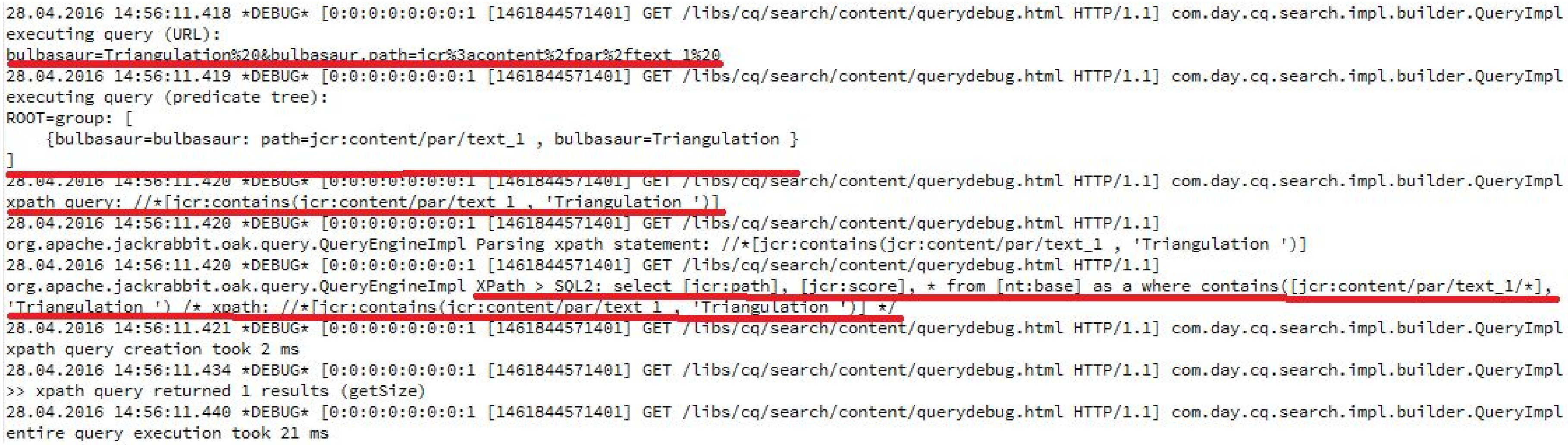
Log Level	Debug Root Logger log level setting. (org.apache.sling.commons.log.level)
Log File	logs/quiery.log The name and path of the log file. If this is empty, logging goes to standard output (the console). If this path is relative it is resolved below ${sling.home}. (org.apache.sling.commons.log.file)
Message Pattern	{0,date,dd.MM.yyyy HH:mm:ss.SSS} *{4}* [{2}] {3} {5} Message Pattern for formatting the log messages. For complete details refer to http://logback.qos.ch/manual/layouts.html#ClassicPatternLayout (org.apache.sling.commons.log.pattern)
Logger	org.apache.jackrabbit.oak.query.QueryEngineImpl com.day.cq.search The logger names applicable for this logger configuration. Each logger name applies for any child category unless configured otherwise. E.g. a logger name of org.apache.sling applies to logger org.apache.sling.commons unless there is a different configuration for org.apache.sling.commons. (org.apache.sling.commons.log.names)

Configuration Information

Persistent Identity (PID)	org.apache.sling.commons.log.LogManager.factory.config.4d87853c-60f0-4cfc-b0a9-22de16753e7c
Factory Persistent Identifier (Factory PID)	org.apache.sling.commons.log.LogManager.factory.config
Configuration Binding	slinginstall:C:\aem6.1_new\crx-quickstart\launchpad\startup\1\org.apache.sling.commons.log-4.0.2.jai
	Apache Sling SLF4J Implementation (Logback) (org.apache.sling.commons.log), Version 4.0.2

AXAMIT

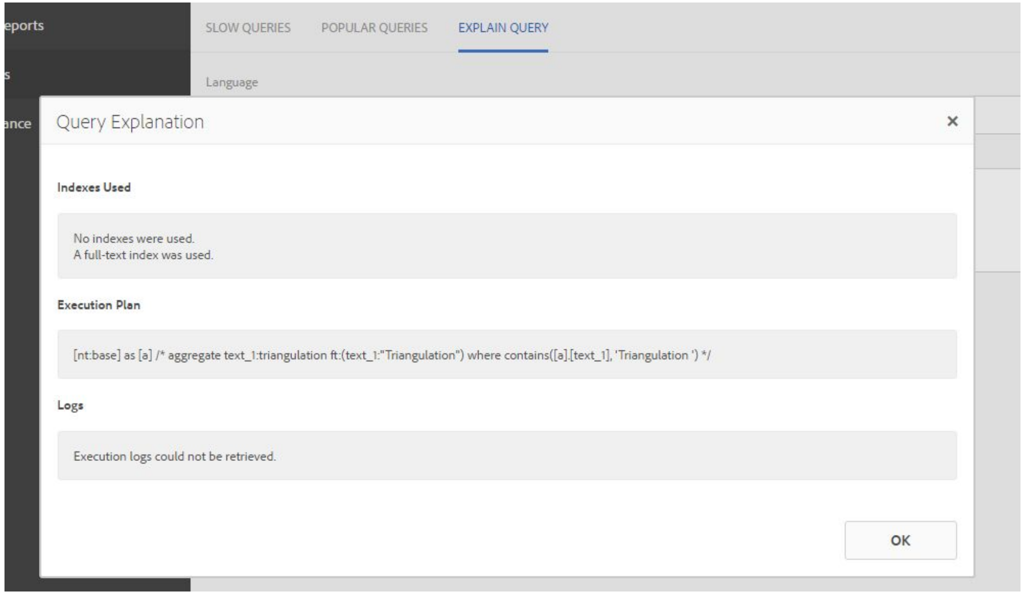
Logs Setup And Monitor



```
28.04.2016 14:56:11.418 *DEBUG* [0:0:0:0:0:0:0:1 [1461844571401] GET /libs/cq/search/content/querydebug.html HTTP/1.1] com.day.cq.search.impl.builder.QueryImpl
executing query (URL):
bulbasaur=Triangulation%20&bulbasaur.path=jcr%3acontent%2fpar%2ftext_1%20
28.04.2016 14:56:11.419 *DEBUG* [0:0:0:0:0:0:0:1 [1461844571401] GET /libs/cq/search/content/querydebug.html HTTP/1.1] com.day.cq.search.impl.builder.QueryImpl
executing query (predicate tree):
ROOT=group: [
    {bulbasaur=bulbasaur: path=jcr:content/par/text_1 , bulbasaur=Triangulation }
]
28.04.2016 14:56:11.420 *DEBUG* [0:0:0:0:0:0:0:1 [1461844571401] GET /libs/cq/search/content/querydebug.html HTTP/1.1] com.day.cq.search.impl.builder.QueryImpl
xpath query: //*[jcr:contains(jcr:content/par/text_1 , 'Triangulation ')]
28.04.2016 14:56:11.420 *DEBUG* [0:0:0:0:0:0:0:1 [1461844571401] GET /libs/cq/search/content/querydebug.html HTTP/1.1]
org.apache.jackrabbit.oak.query.QueryEngineImpl Parsing xpath statement: //*[jcr:contains(jcr:content/par/text_1 , 'Triangulation ')]
28.04.2016 14:56:11.420 *DEBUG* [0:0:0:0:0:0:0:1 [1461844571401] GET /libs/cq/search/content/querydebug.html HTTP/1.1]
org.apache.jackrabbit.oak.query.QueryEngineImpl XPath > SQL2: select [jcr:path], [jcr:score], * from [nt:base] as a where contains([jcr:content/par/text_1/*],
'Triangulation ') /* xpath: //*[jcr:contains(jcr:content/par/text_1 , 'Triangulation ')] */
28.04.2016 14:56:11.421 *DEBUG* [0:0:0:0:0:0:0:1 [1461844571401] GET /libs/cq/search/content/querydebug.html HTTP/1.1] com.day.cq.search.impl.builder.QueryImpl
xpath query creation took 2 ms
28.04.2016 14:56:11.434 *DEBUG* [0:0:0:0:0:0:0:1 [1461844571401] GET /libs/cq/search/content/querydebug.html HTTP/1.1] com.day.cq.search.impl.builder.QueryImpl
>> xpath query returned 1 results (getSize)
28.04.2016 14:56:11.440 *DEBUG* [0:0:0:0:0:0:0:1 [1461844571401] GET /libs/cq/search/content/querydebug.html HTTP/1.1] com.day.cq.search.impl.builder.QueryImpl
entire query execution took 21 ms
```

Query Analyzer

http://localhost:4502/libs/granite/operations/content/diagnosis/tool.html/_granite_queryperformance
//*[jcr:contains(@text_1 , 'Triangulation ')]



Indexes Used

```
No indexes were used.
A full-text index was used.
```

Execution Plan

```
[nt:base] as [a] /* aggregate text_1:triangulation ft:(text_1:"Triangulation") where contains([a].[text_1], 'Triangulation ') */
```

Logs

```
Execution logs could not be retrieved.
```

Links

https://docs.adobe.com/docs/en/aem/6-2/develop/search/querybuilder-api.html

http://www.slideshare.net/alexkli/cq5-querybuilder-adapttoberlin-2011

https://hashimkhan.in/aem-adobecq5-code-templates/query-builder/

http://aemtipsandtricks.blogspot.com.by/2014/07/structuring-content-for-faceted.html

https://hashimkhan.in/tag/predicate/

AXAMIT