AEM Indexing and Query

Thomas Mueller | Best Practices

Bē
Evolution Bureau

About Me

🙋‍♂️

- Thomas Mueller
- Work for Day, then Adobe, since 2005
- Jackrabbit and Oak developer
- Query engine and indexing
- Previously, wrote Java SQL database engines Hypersonic SQL and H2

Agenda

- Related Presentations
- Query Troubleshooting
- Tools
- Index Management
- Best Practices
- Future Plans
- Links

Related Presentations

- Oak and Queries (2014)
  https://docs.adobe.com/ddc/en/gems/aem-6-oak--mongomk-and-queries.html
- Oak Lucene Indexes (2016)
  https://docs.adobe.com/ddc/en/gems/oak-lucene-indexes.html
- Indexing Best Practices and Troubleshooting (2017)
  https://helpx.adobe.com/experience-manager/kt/eseminars/ccoo-aem-indexing-recording.html
- Query Builder (2017)
  https://docs.adobe.com/ddc/en/gems/Search-forms-made-easy-with-the-AEM-querybuilder.html

Query Troubleshooting

Situation:

- Application is slow
- Suspected due to slow query

How to analyze and resolve?

Analyse Logs & Thread Dumps

- Grep for traversal messages

  ```
  grep -i -R --include=*.log "travers" crx-quickstart/logs
  ```

- If needed, enable query debug logging

  http://localhost:4502/system/console/slinglog

  DEBUG, logs/query.log, org.apache.jackrabbit.oak.query

- (If nothing found, analyze thread dumps using oak-run)

  ```
  java -jar oak-run-*.jar threaddump --filter --profile crx-quickstart/threaddumps
  ```

Slow Query In Log

- The following is found in the log file:

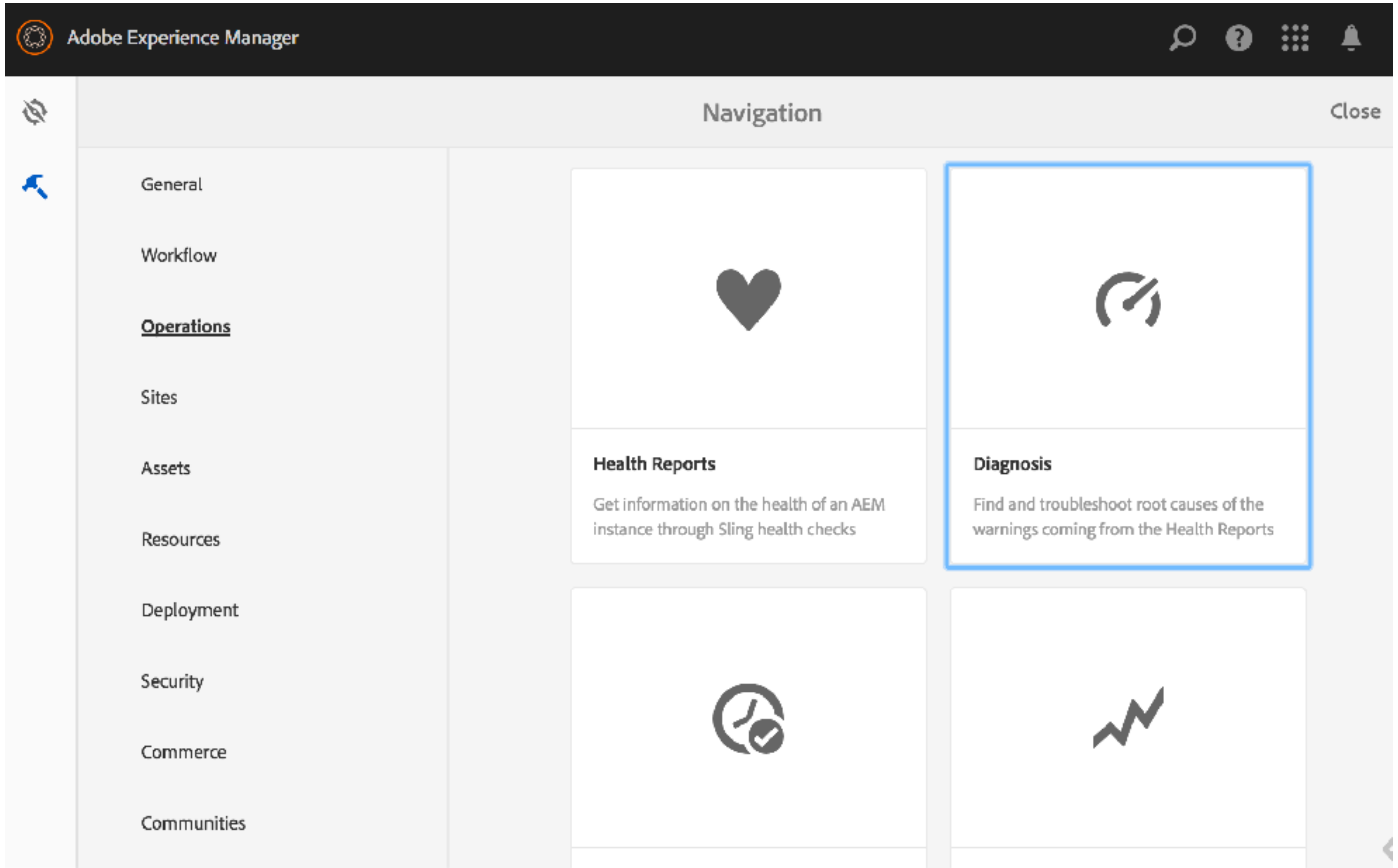
```
*WARN* org.apache.jackrabbit.oak.spi.query.Cursors$TraversingCursor
Traversed 132000 nodes with filter Filter(query=
select * from [nt:base] where isdescendantnode('/etc')
and lower([jcr:title]) like '%coat%');
consider creating an index or changing the query
```

- The XPath query is:

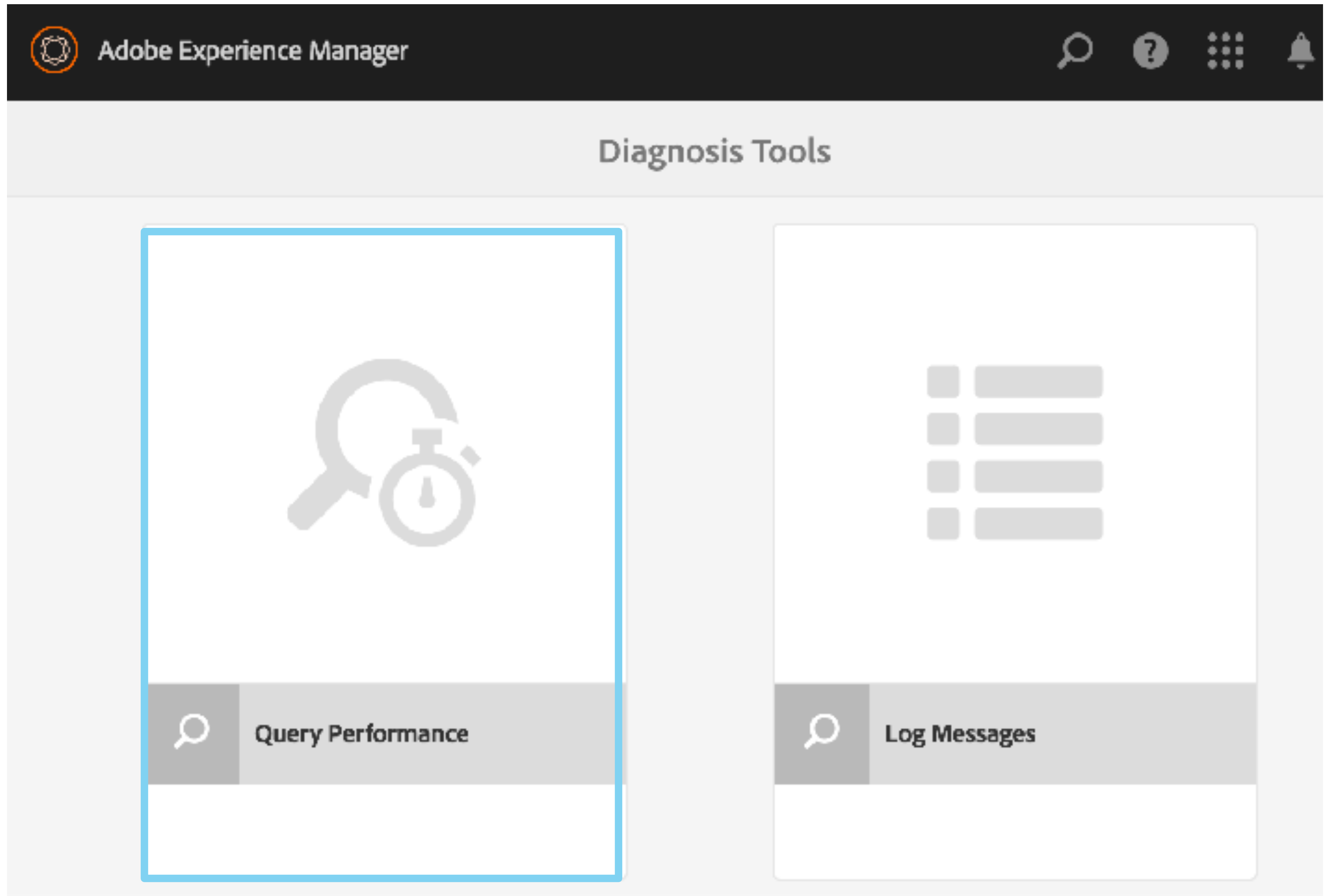
  `/jcr:root/etc//*[jcr:like(fn:lower-case(@jcr:title), '%coat%')]`

- This query traversed many nodes. Let's find out why.

Query Plan

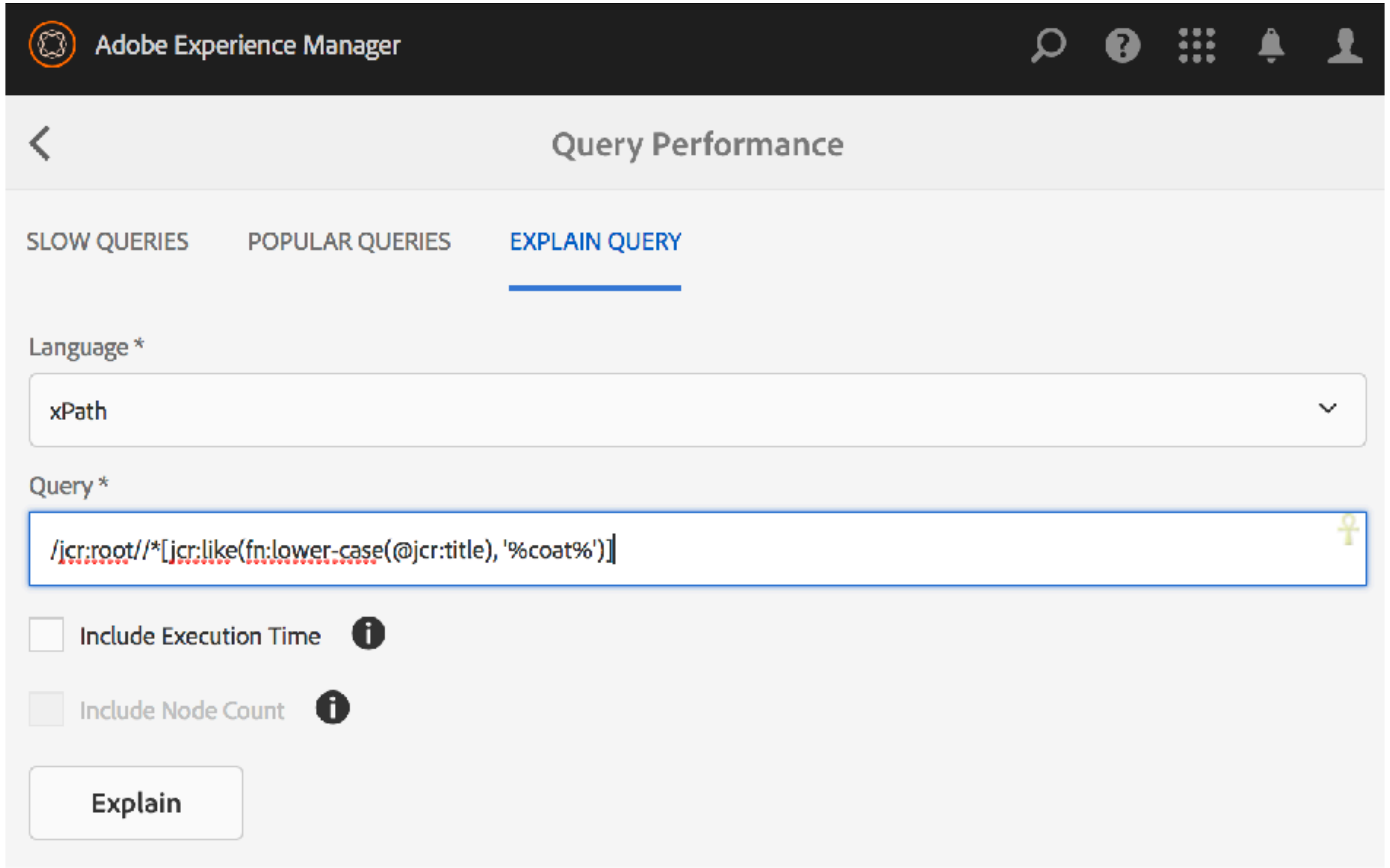


Adobe Experience Manager
Navigation
Close
General
Workflow
Operations
Sites
Assets
Resources
Deployment
Security
Commerce
Communities
Health Reports
Get information on the health of an AEM instance through Sling health checks
Diagnosis
Find and troubleshoot root causes of the warnings coming from the Health Reports



Adobe Experience Manager
Diagnosis Tools
Query Performance
Log Messages

Query Plan



Language *

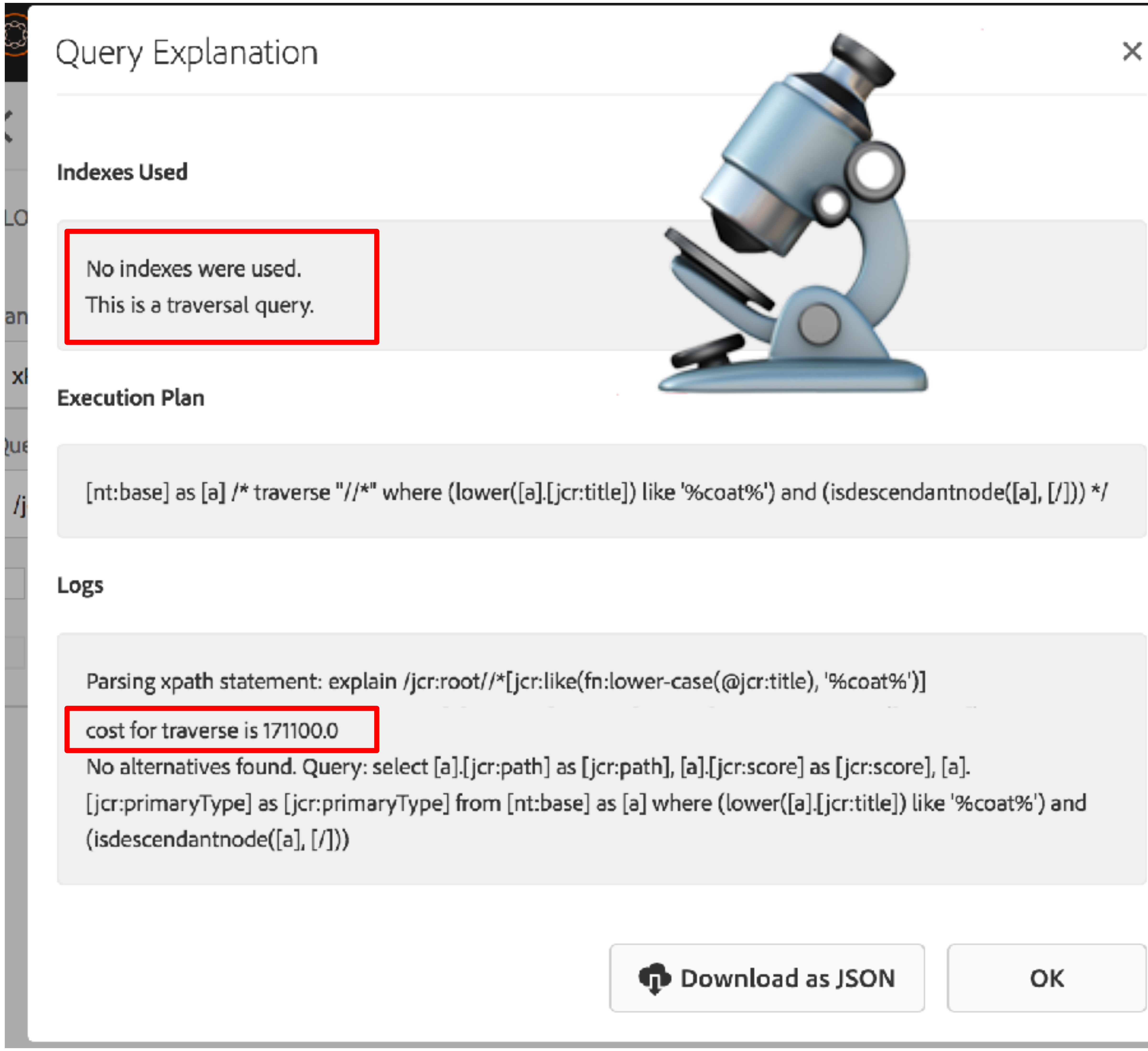
xPath

Query *

```
/jcr:root//*[jcr:like(fn:lower-case(@jcr:title), '%coat%')]
```

□ Include Execution Time

□ Include Node Count



Indexes Used

No indexes were used.
This is a traversal query.

Execution Plan

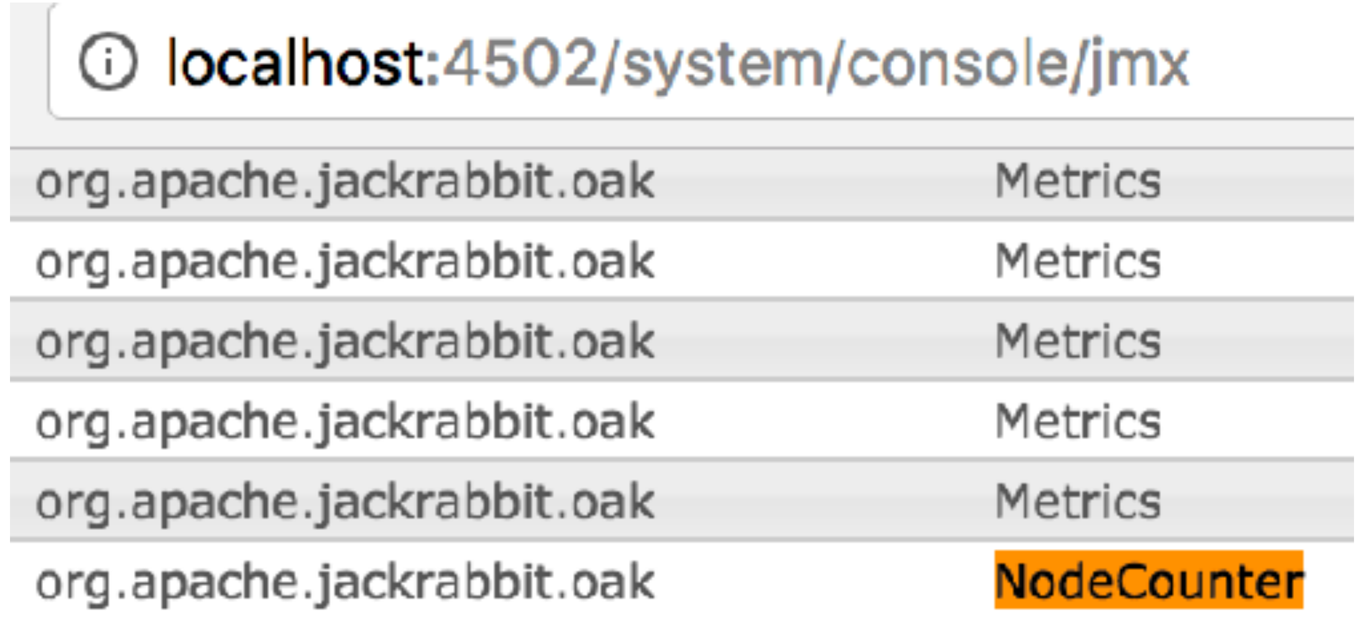
```
[nt:base] as [a] /* traverse "//*" where (lower([a].[jcr:title]) like '%coat%') and (isdescendantnode([a], [/])) */
```

Logs

```
Parsing xpath statement: explain /jcr:root//*[jcr:like(fn:lower-case(@jcr:title), '%coat%')]
cost for traverse is 171100.0
No alternatives found. Query: select [a].[jcr:path] as [jcr:path], [a].[jcr:score] as [jcr:score], [a].
[jcr:primaryType] as [jcr:primaryType] from [nt:base] as [a] where (lower([a].[jcr:title]) like '%coat%') and
(isdescendantnode([a], [/]))
```

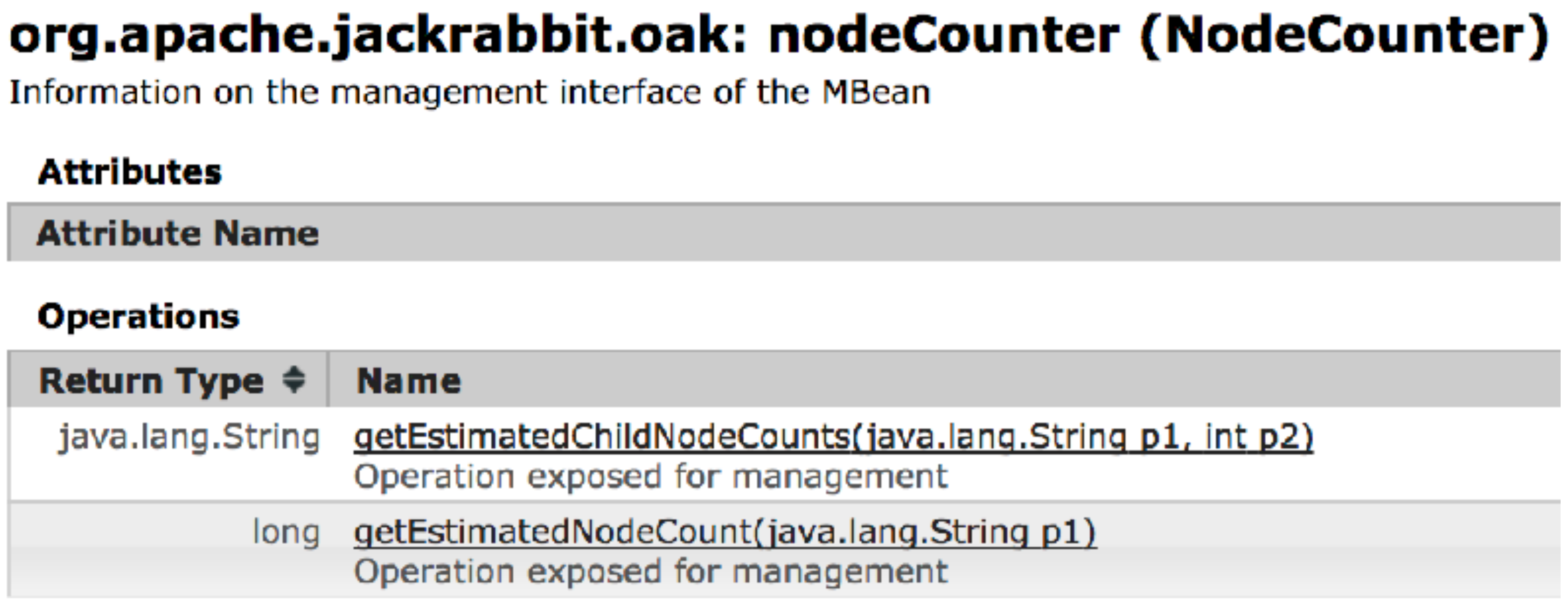
Path Restriction

- Estimate the node count



ⓘ localhost:4502/system/console/jmx

org.apache.jackrabbit.oak	Metrics
org.apache.jackrabbit.oak	Metrics
org.apache.jackrabbit.oak	Metrics
org.apache.jackrabbit.oak	Metrics
org.apache.jackrabbit.oak	Metrics
org.apache.jackrabbit.oak	NodeCounter



org.apache.jackrabbit.oak: nodeCounter (NodeCounter)

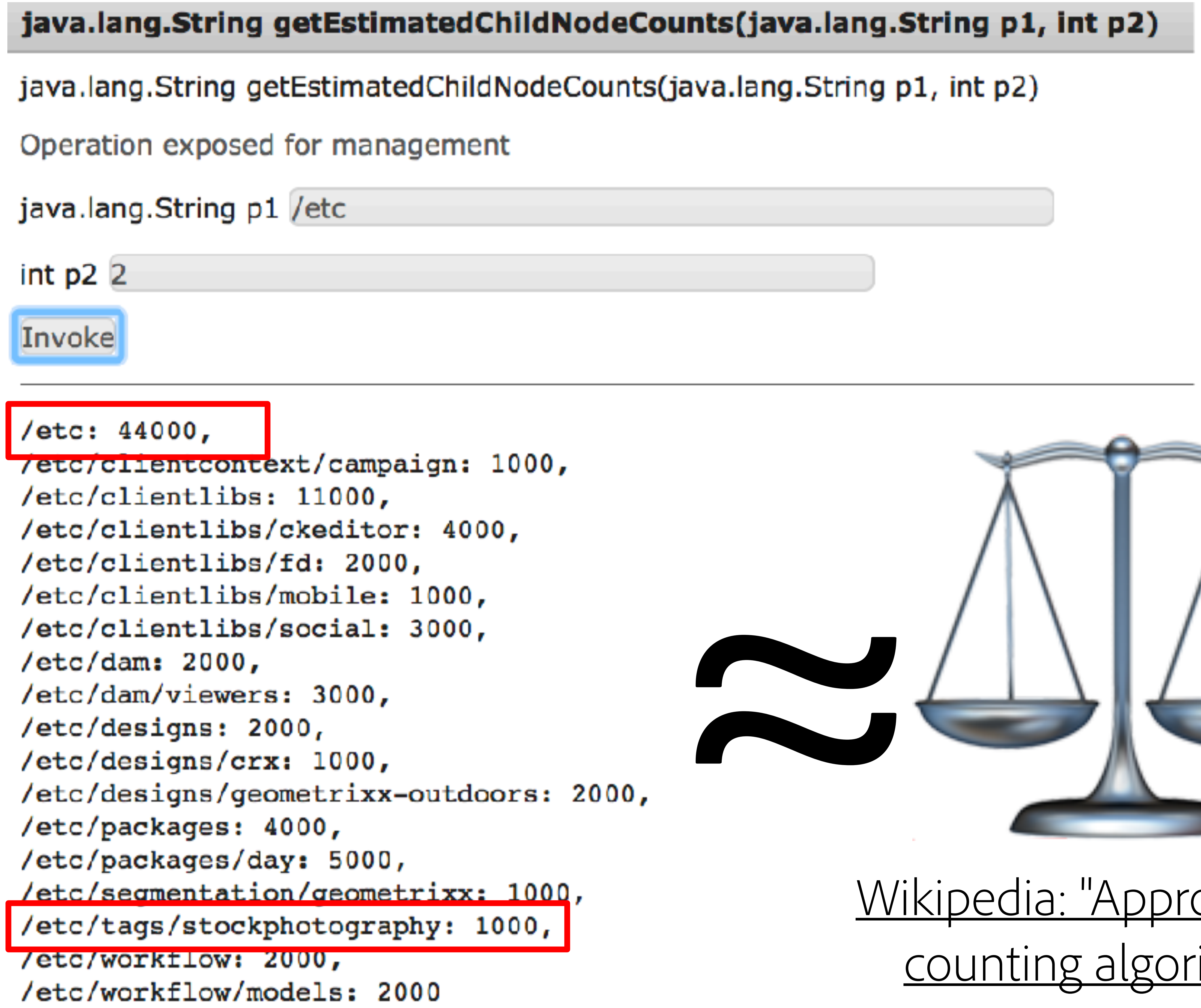
Information on the management interface of the MBean

Attributes

Attribute Name

Operations

Return Type ⬍	Name
java.lang.String	getEstimatedChildNodeCounts(java.lang.String p1, int p2) Operation exposed for management
long	getEstimatedNodeCount(java.lang.String p1) Operation exposed for management



java.lang.String getEstimatedChildNodeCounts(java.lang.String p1, int p2)

java.lang.String getEstimatedChildNodeCounts(java.lang.String p1, int p2)

Operation exposed for management

java.lang.String p1 /etc

int p2 2

```
/etc: 44000,
/etc/clientcontext/campaign: 1000,
/etc/clientlibs: 11000,
/etc/clientlibs/ckeditor: 4000,
/etc/clientlibs/fd: 2000,
/etc/clientlibs/mobile: 1000,
/etc/clientlibs/social: 3000,
/etc/dam: 2000,
/etc/dam/viewers: 3000,
/etc/designs: 2000,
/etc/designs/crx: 1000,
/etc/designs/geometrixx-outdoors: 2000,
/etc/packages: 4000,
/etc/packages/day: 5000,
/etc/segmentation/geometrixx: 1000,
/etc/tags/stockphotography: 1000,
/etc/workflow: 2000,
/etc/workflow/models: 2000
```

≈

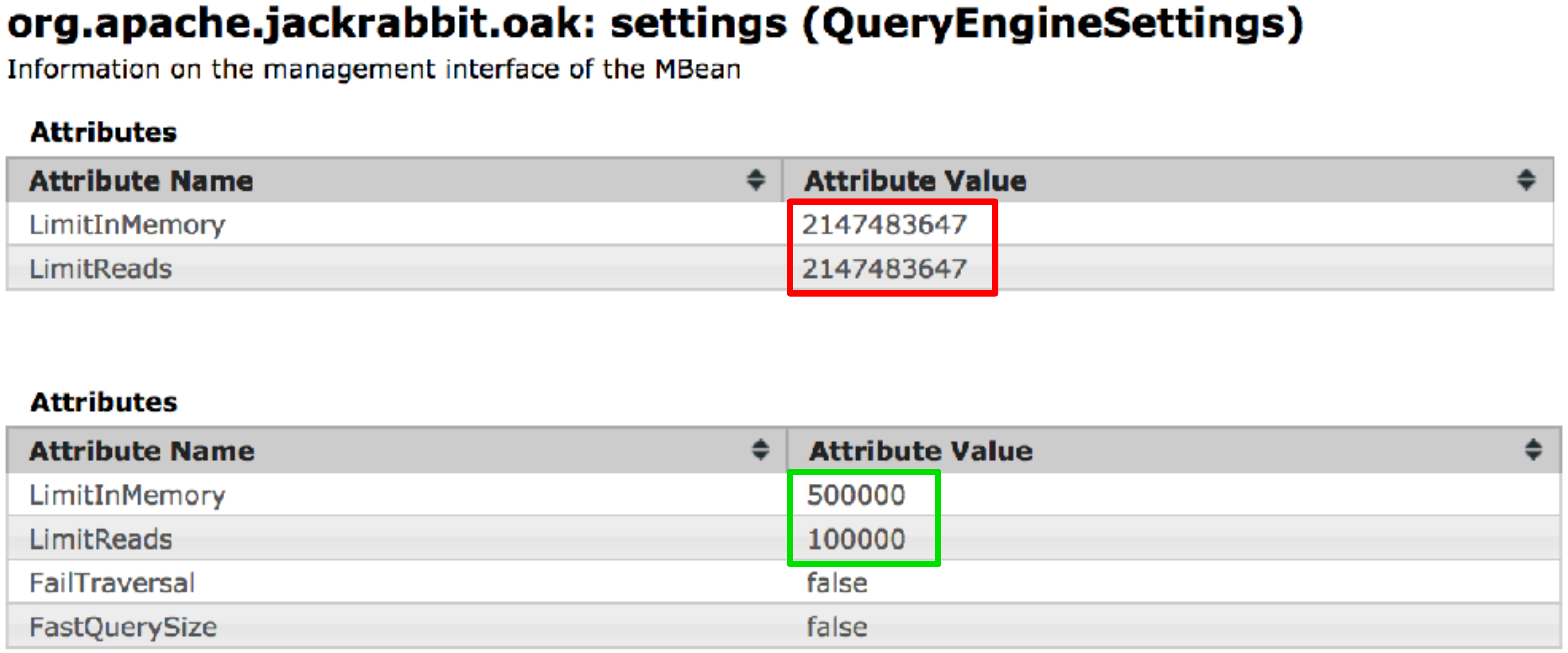
Wikipedia: "Approximate counting algorithm"

Path Restriction

- Improved path condition: cost = 2000
  `/jcr:root/etc/tags//*[...]`
- Traversal is OK if number of nodes is *guaranteed* to be low
  `... option(traversal ok)`
- Traversal limit defaults

AEM 6.0-6.2:

AEM 6.3:



org.apache.jackrabbit.oak: settings (QueryEngineSettings)

Information on the management interface of the MBean

Attributes

Attribute Name	Attribute Value
LimitInMemory	2147483647
LimitReads	2147483647

Attributes

Attribute Name	Attribute Value
LimitInMemory	500000
LimitReads	100000
FailTraversal	false
FastQuerySize	false

Nodetype Restriction

- Restriction on the node type:

  ```
  /jcr:root/etc/tags//element(*, cq:Tag)[...]
  ```

- Better: uses an index (cqTagLucene), cost = 258
- With node type restriction, the query is better, but reads all tags, with any title, and filters later:

  ```
  [jcr:like(fn:lower-case(@jcr:title), '%coat%')] -> jcr:title:[* TO *]
  ```

Prefer "contains" over "like"

- Avoid "like %...%" as no index can be used
- Prefer "contains ...", it can use a fulltext index, and can use aggregation
- In both cases, escaping is needed

🦏 ✗

```
jcr:like(@rep:authorizableId, '%joe%') or
jcr:like(@rep:principalName, '%joe%') or
jcr:like(profile/@givenName, '%joe%') or
...
```

🐅 ✓

```
jcr:contains(., 'joe')
```

Prefer "contains" over "like"

- Fulltext index is *much* faster

```
[jcr:contains(@jcr:title, 'coat')] -> full:jcr:title:coat
```

- Matches only the word, but not e.g. "sugarcoat"
- Apache Lucene query parser syntax applies:

`coat*, coa?`	Wildcard search	`coat^3`	Boost
`"coat hook"`	Phrase query	`coat OR tunic`	Conjunction operator
`{coat TO cola}`	Range, excluding	`coat -tunic`	Exclude a term

Escape to Avoid Code Injection

- Avoid code like this:

```
String param = request.getParameter("param");
String query = "/jcr:root/etc//*[jcr:contains(., '" + param + "')";
```

- Best use bind variables:

```
Query q = qm.createQuery(
    "/jcr:root/etc//*[jcr:contains(., $p)", "xpath");
q.bindValue("p", vf.createValue(param));
```

- Escape at least single quotes:

```
param = param.replaceAll("'", "''");
```

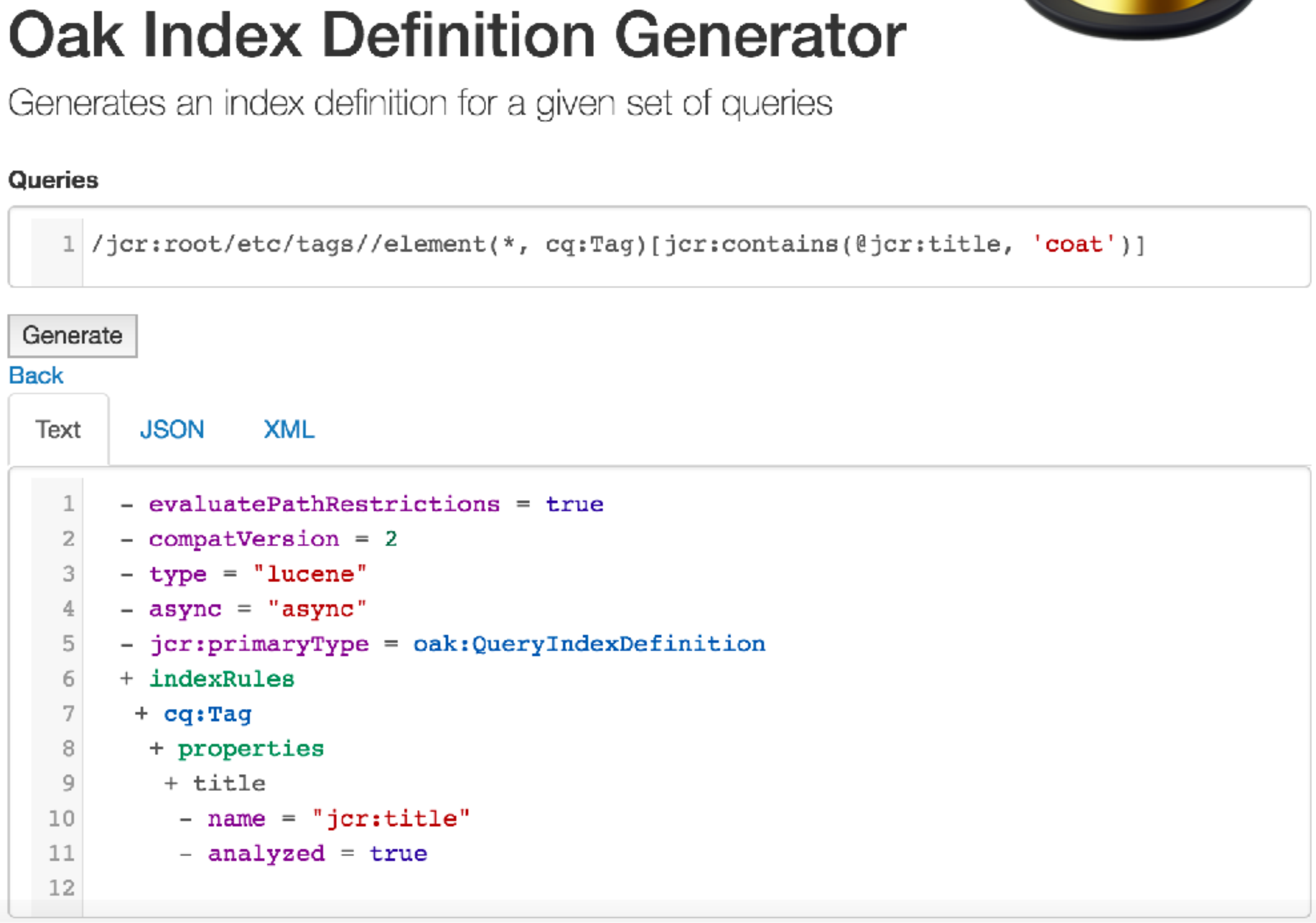
- Use escape utils (e.g. jackrabbit-jcr-commons org.apache.jackrabbit.util.Text)

Index Management

- If still slow, modify/create an index (in `/oak:index` or `/content/oak:index`)
- If the wrong index is used, report an issue (in many cases there are workarounds)
- Index generation tool, docs
  http://oakutils.appspot.com/generate/index
  http://jackrabbit.apache.org/oak/docs/query/lucene.html
- Reindexing only for cases listed, can take days!

Index Generation Tool

- Paste the query
- Get the index
- Try extending an existing index
- Avoid large indexes (see docs!)



Oak Index Definition Generator

Generates an index definition for a given set of queries

Queries

```
/jcr:root/etc/tags//element(*, cq:Tag)[jcr:contains(@jcr:title, 'coat')]
```

```
- evaluatePathRestrictions = true
- compatVersion = 2
- type = "lucene"
- async = "async"
- jcr:primaryType = oak:QueryIndexDefinition
+ indexRules
 + cq:Tag
  + properties
   + title
    - name = "jcr:title"
    - analyzed = true
```

Reindexing

- Heavyweight, can take days!
- Text pre-extraction can reduce time a *lot*
  https://jackrabbit.apache.org/oak/docs/query/pre-extract-text.html
- Disable Datastore GC
- Reindexing only for cases listed
  http://jackrabbit.apache.org/oak/docs/query/indexing.html#reindexing
  - New or changed index definition
  - Bug: Counter index out of sync
  - Bug: Lucene binary is missing or corrupt
  - Bug: Very large transaction + many child nodes
  - Bug: Partial migration using sidegrade

Best Practices

Query Languages

Query Builder	✓✓	Preferred
XPath	✓✓	Preferred (JCR API), even if deprecated in the spec!
SQL-2	✓	A bit verbose; lower level
SQL (old)	✗	Don't use for new code
GQL	✗	Don't use for new code (Google Query Language)

Best Practices

Index Types (for new queries)

Lucene	✓	Preferred
Nodetype	⚠	Use only for nodetypes with few nodes
Property	⚠	Can be synchronous, to enforce unique values
Solr	⚠	Rarely used so far
Ordered	✗	Don't use, replace (deprecated, will go away)
Traverse	⚠	Only if the result is very small

Best Practices

Query Features / Config Options

👍

getSize	⚠️	May return -1; use "fast result size" option https://jackrabbit.apache.org/oak/docs/query/query-engine.html Result Size
Traversal Limits	✔	AEM 6.3: Use defaults AEM 6.0 - 6.2: Use system properties https://jackrabbit.apache.org/oak/docs/query/query-engine.html Slow Queries and Read Limits Append "option (traversal fail)"

Best Practices

Query Features

Join (SQL-2)	✓	Best join on path
Ordering	✓	Requires ordering in index for large results; "jcr:score" is ignored
Union	✓	Runs multiple queries and combines the results; XPath union since AEM 6.3: `/jcr:root/(etc | libs)//*[...]`

Best Practices

Query Restrictions

Path	✓✓	Always if possible
Nodetype	✓✓	Always if possible
Node name	✓	Avoid requiring an index on all node names
.. and ..	✓✓	The more, the better (@a=1 and @b=2 and @c=3)
.. or ..	⚠	Avoid excessive use @a=1 or @a=2 is fine, but @a=1 or @b=1 is converted to "union"

Best Practices

Query Conditions

not null	✔	Requires "notNullCheckEnabled"
is null	⚠	Requires "nullCheckEnabled" (expensive)
not	⚠	Traverses the nodes
suggest	⚠	See documentation
upper / lower =	⚠	Traverses (index support since AEM 6.3)

Best Practices

Query Conditions

Equality	✓✓	The more, the better
Range	✓	Requires ordering in index
Contains	✓	Requires fulltext index, escaping (Lucene syntax)
Like	⚠	Efficiently using an index is tricky
Other	✓	Boost, Suggestions, Spellcheck,...: see documentation

Tools & Settings

- Query Builder Debugger
  http://localhost:4502/libs/cq/search/content/querydebug.html
- Explain Query Tool
  http://localhost:4502/libs/granite/operations/content/diagnosis/tool.html/granite_queryperformance
  (Tools-Operations-Diagnosis-Query Performance-Explain Query)
- Node Counter Bean
  http://localhost:4502/system/console/jmx - NodeCounter
- Query Engine Settings
  http://localhost:4502/system/console/jmx - QueryEngineSettings
  http://localhost:4502/system/console/configMgr - Query Engine Settings Service

Query Tool (crx/de) Limitations

- The crx/de query tool has some quirks, use with care:
  - Reported time excludes iteration
  - Exceptions are swallowed
  - Always lists nodes of first selector
  - Can generate slow queries
- For best results, use code

Sample JSP Page

```
<%@include file="/libs/foundation/global.jsp"%>
<%@page session="false" contentType="text/html; charset=utf-8"
import="javax.jcr.*, javax.jcr.query.*"%><%! %><%
// Nodes: /libs/cq/core/components/test (sling:Folder)
// sling:resourceType = /libs/cq/core/components/test
// /libs/cq/core/components/test/test.jsp (nt:file)
// Run: http://localhost:4502/libs/cq/core/components/test.html
Session s = resourceResolver.adaptTo(Session.class);
QueryManager qm = s.getWorkspace().getQueryManager();
QueryResult r = qm.createQuery("/jcr:root/tmp//*", "xpath").execute();
for (RowIterator it = r.getRows(); it.hasNext();) {
%><%= it.nextRow().getPath() %> <br /><% } %>
```

Reporting Issues

- Small, self contained, reproducible test case
- Version, configuration, indexes

  http://localhost:4502/oak:index.tidy.-1.json plus other indexes
- Log files

  crx-quickstart/logs/error.log, query.log (module "org.apache.jackrabbit.oak.query")
- Node counter data

  http://localhost:4502/system/console/jmx NodeCounter getEstimatedChildNodeCounts(p1=/, p2=2)
- Thread dumps, heap histograms

  crx-quickstart/threaddumps / jstack -l <pid> / jmap -histo <pid>

Future Plans

AEM 6.3, 6.2,...:

- Indexing using oak-run (much faster!)

AEM 6.4:

- Status overview page
- Improved index manager, crx/de, explain query
- Maybe integrated query builder debugger
- Improved health checks

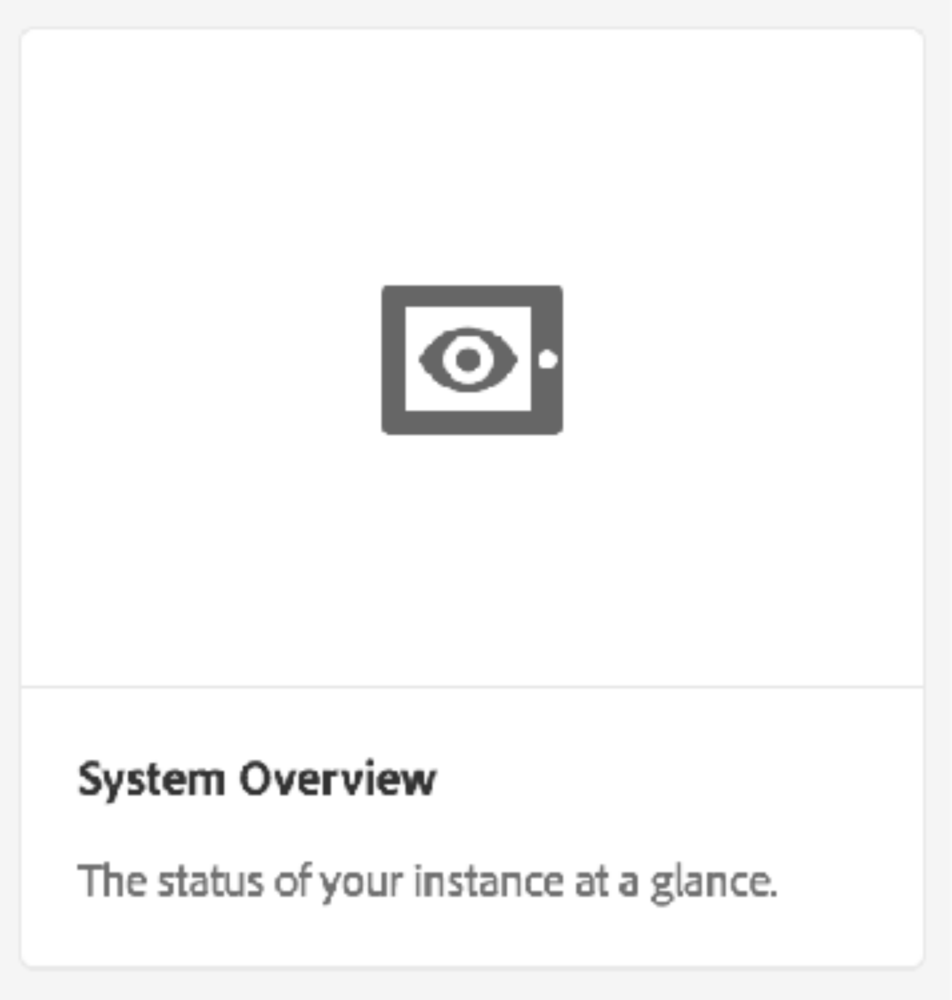
Links

- Adobe Documentation
  https://helpx.adobe.com/experience-manager/kb/Analyzing-AEM-Indexing-Issues.html
  https://docs.adobe.com/docs/en/aem/6-3/deploy/best-practices/best-practices-for-queries-and-indexing.html
  https://docs.adobe.com/docs/en/aem/6-3/deploy/platform/queries-and-indexing.html
- Oak Documentation
  http://jackrabbit.apache.org/oak/docs/query/query.html
- Apache Lucene
  https://lucene.apache.org/core/4_7_1/queryparser/org/apache/lucene/queryparser/classic/package-summary.html#Overview

Adobe

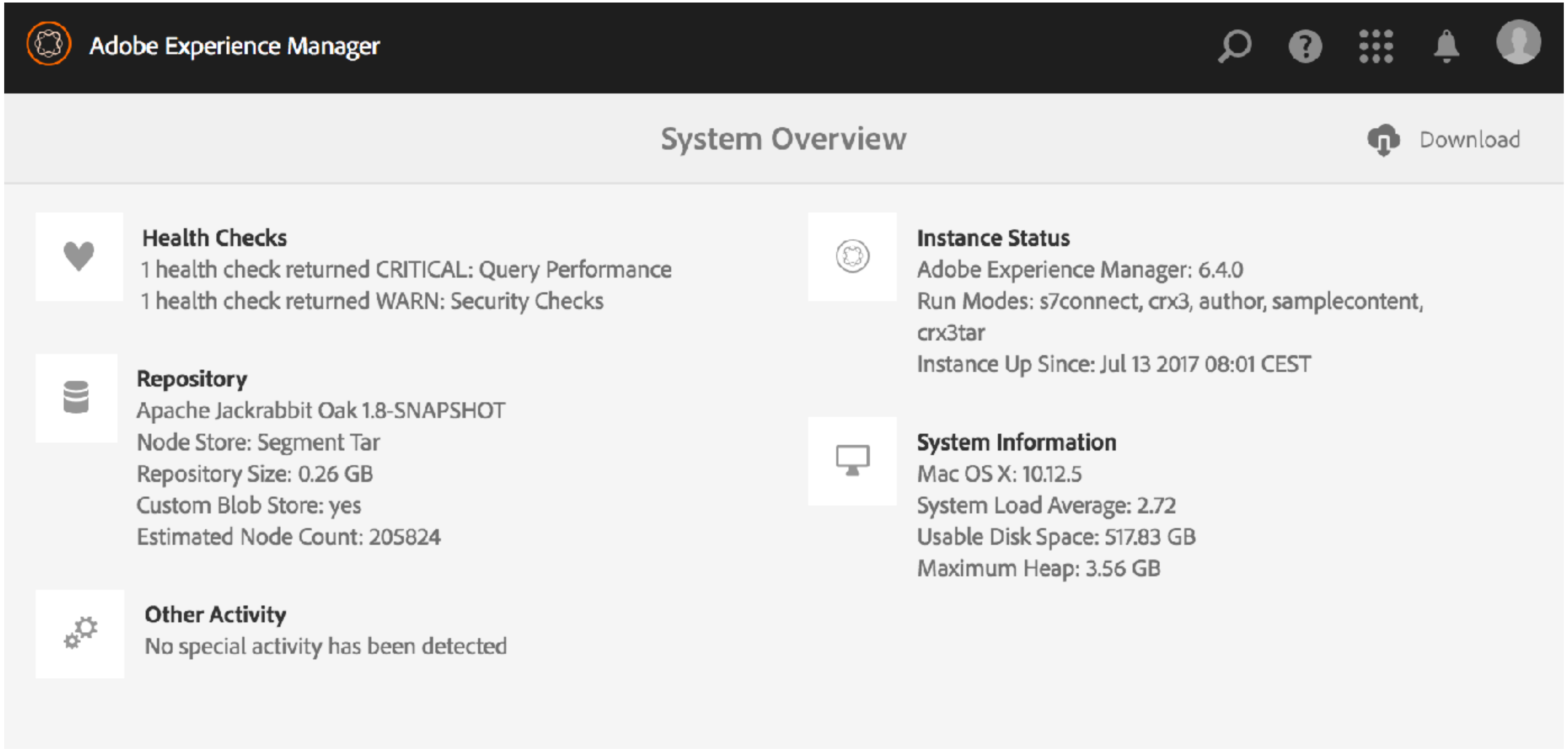
MAKE IT AN EXPERIENCE

Status Overview (Mockup)



System Overview

The status of your instance at a glance.



Adobe Experience Manager

System Overview

Health Checks
1 health check returned CRITICAL: Query Performance
1 health check returned WARN: Security Checks

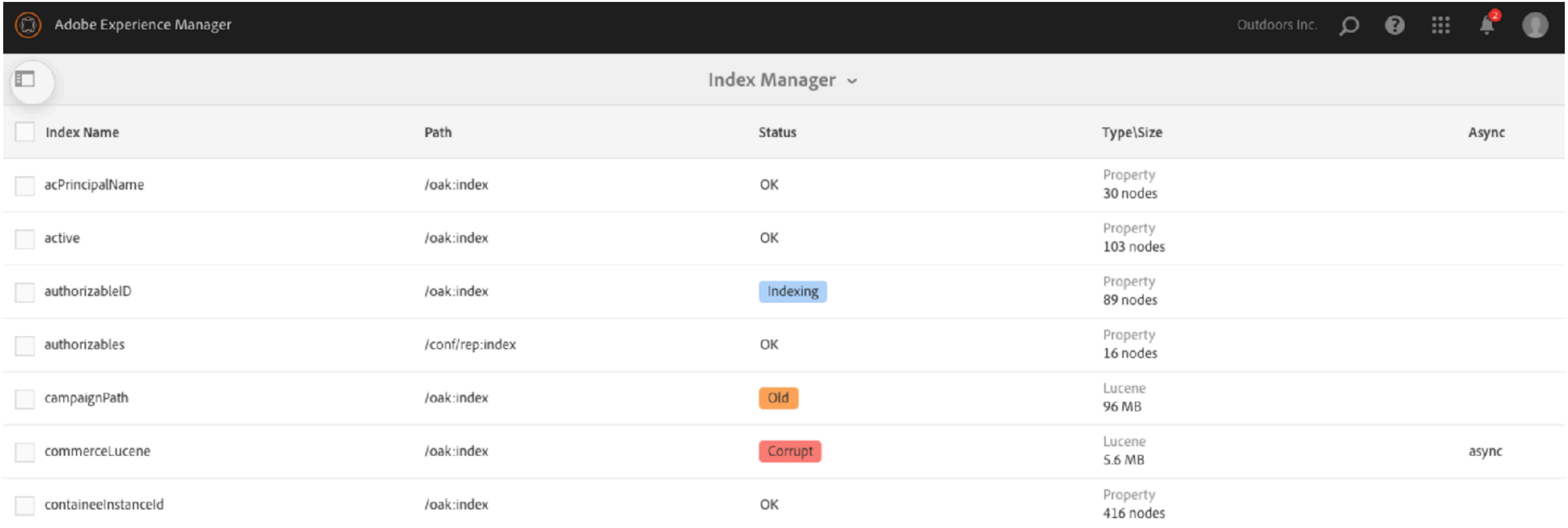
Repository
Apache Jackrabbit Oak 1.8-SNAPSHOT
Node Store: Segment Tar
Repository Size: 0.26 GB
Custom Blob Store: yes
Estimated Node Count: 205824

Other Activity
No special activity has been detected

Instance Status
Adobe Experience Manager: 6.4.0
Run Modes: s7connect, crx3, author, samplecontent, crx3tar
Instance Up Since: Jul 13 2017 08:01 CEST

System Information
Mac OS X: 10.12.5
System Load Average: 2.72
Usable Disk Space: 517.83 GB
Maximum Heap: 3.56 GB

Index Manager (Mockup)



Index Manager ⌄

□ Index Name	Path	Status	Type\Size	Async
□ acPrincipalName	/oak:index	OK	Property 30 nodes	
□ active	/oak:index	OK	Property 103 nodes	
□ authorizableID	/oak:index	Indexing	Property 89 nodes	
□ authorizables	/conf/rep:index	OK	Property 16 nodes	
□ campaignPath	/oak:index	Old	Lucene 96 MB	
□ commerceLucene	/oak:index	Corrupt	Lucene 5.6 MB	async
□ containeeInstanceId	/oak:index	OK	Property 416 nodes	