

A Service Oriented Architecture for the NetarchiveSuite Frontend

.... or "How Hard Can It Be To Find A Webdeveloper?"

.... or "How To Build A Full-stack Webdev in Two Easy Lessons"

.... or "How I Learnt To Stop Worrying And Love Javascript"

(Based on work from Innovation Week September 2017 at KB-Aarhus.)

Where We Are Now

<http://kb-test-adm-001.kb.dk:8078/>

The screenshot shows a web browser window with the following details:

- Browser tabs: Selective Harvest, Netarchive Status, http://kb-test...st/status/all
- Address bar: kb-test-adm-001.kb.dk:8078/HarvestDefinition/Definitions-edit-selective
- Language selection: Dansk, English, Deutsch, Italiano, Français
- Page title: Selective Harvest
- Form fields: Harvest name, Audience, Comments, Schedule (set to 'Once_a_week')
- Message: There are 0 domain configurations in this harvest definition.
- Table with 3 columns: Domain, Choose configuration, Remove from list
- Text: Enter domain(s) to add to the harvest here:
- Form field for domain input and an 'Add domains' button.

What do we see?

- A big ball of jsp and servlets
- No clear separation between business logic and presentation
- Multiple back ends
 - Database via DAO classes
 - External NAS applications via JMX/JMS
 - Heritrix via REST

So can we learn to make a web frontend like real web-developers?

Let's do it in two stages

Create a Service Layer For the NAS Backend

All communication between the NAS GUI and the backend system should be via a well-defined REST API

1. Create new webmodule in the existing NAS GUI
2. Configure its web.xml to scan for classes that expose REST methods

```
<servlet>
  <servlet-name>Jersey REST Service</servlet-name>

  <servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet
-class>
  <init-param>
    <param-name>jersey.config.server.provider.packages</param-name>
    <param-value>dk.netarkivet.common.api</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
```

3. Implement some methods

```

@Path("/status")
public class Status {

    @GET
    @Path("/all")
    @Produces(MediaType.APPLICATION_JSON)
    public JMXStatusEntry[] getAllStatus() {
        String query = "dk.netarkivet.common.logging:*,index=0";
        ArrayList<JMXStatusEntry> entries = new ArrayList<>();

        List<StatusEntry> entries1 = null;
        try {
            entries1 = JMXStatusEntry.queryJMX(query);
        } catch (MalformedObjectNameException e) {
            throw new RuntimeException(e);
        }
        for (StatusEntry entry: entries1) {
            entry.getLogMessage(Locale.getDefault());
            entries.add((JMXStatusEntry) entry);
        }
        return entries.toArray(new JMXStatusEntry[ ]{});
    }
}

```

4. Profit! <http://kb-test-adm-001.kb.dk:8078/rest/rest/status/all>

```
Selective Harvest x Netarchive Status x http://kb-test...st/status/all x +
kb-test-adm-001.kb.dk:8078/rest/rest/status/all
[{"logMessage":"2017-09-08 08:39:48.225 [main] INFO dk.netarkivet.archive.bitarchive.distribute.BitarchiveServer.<init> - Broadcasting heartbeats on:
CSR_KB_THE_BAMON\r\n", "physicalLocation":"K", "machineName":"KB-TEST-BAR-
015.bitarkiv.kb.dk", "httpPort":"","applicationInstanceId":"","harvestPriority":"","archiveReplicaName":"KBN", "applicationName":"dk.netarkivet.archive.bi
tarchive.BitarchiveApplication", "index":"0"}, {"logMessage":"2017-09-08 08:40:12.734 [main] INFO
dk.netarkivet.archive.bitarchive.distribute.BitarchiveServer.<init> - Broadcasting heartbeats on: CSR_KB_THE_BAMON\r
\n", "physicalLocation":"K", "machineName":"KB-TEST-BAR-
016.bitarkiv.kb.dk", "httpPort":"","applicationInstanceId":"BitApp_1", "harvestPriority":"","archiveReplicaName":"KBN", "applicationName":"dk.netarkivet.ar
chive.bitarchive.BitarchiveApplication", "index":"0"}, {"logMessage":"2017-09-08 08:40:12.734 [main] INFO
dk.netarkivet.archive.bitarchive.distribute.BitarchiveServer.<init> - Broadcasting heartbeats on: CSR_KB_THE_BAMON\r
\n", "physicalLocation":"K", "machineName":"KB-TEST-BAR-
016.bitarkiv.kb.dk", "httpPort":"","applicationInstanceId":"BitApp_2", "harvestPriority":"","archiveReplicaName":"KBN", "applicationName":"dk.netarkivet.ar
chive.bitarchive.BitarchiveApplication", "index":"0"}, {"logMessage":"2017-09-08 08:40:12.734 [main] INFO
dk.netarkivet.archive.bitarchive.distribute.BitarchiveServer.<init> - Broadcasting heartbeats on: CSR_KB_THE_BAMON\r
\n", "physicalLocation":"K", "machineName":"KB-TEST-BAR-
016.bitarkiv.kb.dk", "httpPort":"","applicationInstanceId":"BitApp_3", "harvestPriority":"","archiveReplicaName":"KBN", "applicationName":"dk.netarkivet.ar
chive.bitarchive.BitarchiveApplication", "index":"0"}, {"logMessage":"2017-09-08 08:40:42.877 [main] INFO
dk.netarkivet.archive.checksum.distribute.ChecksumFileServer.<init> - ChecksumFileServer '130.226.228.5' initialised. Using precomputedChecksums during
upload: false\r\n", "physicalLocation":"K", "machineName":"kb-test-ac-
001.kb.dk", "httpPort":"","applicationInstanceId":"","harvestPriority":"","archiveReplicaName":"CSN", "applicationName":"dk.netarkivet.archive.checksum.Ch
ecksumFileApplication", "index":"0"}, {"logMessage":"2017-09-08 08:40:43.541 [Timer-1] INFO
d.netarkivet.harvester.indexserver.distribute.IndexRequestServer.run - Enabling listening to the indexserver channel '[Queue
'CSR COMMON INDEX SERVER']\n", "physicalLocation":"K", "machineName":"kb-test-ac-
001.kb.dk", "httpPort":"","applicationInstanceId":"","harvestPriority":"","archiveReplicaName":"KBN", "applicationName":"dk.netarkivet.harvester.indexserv
er.IndexServerApplication", "index":"0"}, {"logMessage":"2017-09-08 08:40:43.211 [main] INFO dk.netarkivet.viewerproxy.WebProxy.<init> - Starting
viewerproxy jetty on port 8078\r\n", "physicalLocation":"K", "machineName":"kb-test-ac-
001.kb.dk", "httpPort":"8078", "applicationInstanceId":"","harvestPriority":"","archiveReplicaName":"KBN", "applicationName":"dk.netarkivet.viewerproxy.Vie
werProxyApplication", "index":"0"}, {"logMessage":"2017-09-08 08:39:22.340 [main] INFO dk.netarkivet.archive.arcrepository.ArcRepository.<init> -
Starting the ArcRepository\r\n", "physicalLocation":"K", "machineName":"kb-test-adm-
001.kb.dk", "httpPort":"","applicationInstanceId":"","harvestPriority":"","archiveReplicaName":"KBN", "applicationName":"dk.netarkivet.archive.arcposito
ry.ArcRepositoryApplication", "index":"0"}, {"logMessage":"2017-09-08 08:40:12.658 [imqConsumerReader-0-1705560651219968256-0] INFO
dk.netarkivet.archive.bitarchive.BitarchiveMonitor.signOfLife - Bitarchive '172.17.0.56 BitApp_3' is now known by the bitarchive
monitor\r\n", "physicalLocation":"K", "machineName":"kb-test-adm-
001.kb.dk", "httpPort":"","applicationInstanceId":"KBBM", "harvestPriority":"","archiveReplicaName":"KBN", "applicationName":"dk.netarkivet.archive.bitarch
ive.BitarchiveMonitorApplication", "index":"0"}, {"logMessage":"2017-09-08 08:40:54.292 [imqConsumerReader-0-1705560651219977984-0] INFO
dk.netarkivet.archive.bitarchive.BitarchiveMonitor.signOfLife - Bitarchive '130.225.27.141' is now known by the bitarchive
monitor\r\n", "physicalLocation":"K", "machineName":"kb-test-adm-
001.kb.dk", "httpPort":"","applicationInstanceId":"SBBM", "harvestPriority":"","archiveReplicaName":"SBN", "applicationName":"dk.netarkivet.archive.bitarch
ive.BitarchiveMonitorApplication", "index":"0"}, {"logMessage":"2017-09-08 08:40:49.564 [imqConsumerReader-0-1705560651219870976-0] INFO
dk.netarkivet.harvester.scheduler.HarvestChannelRegistry.logStatus - HarvestChannelRegistry status:\n\t- LOWPRIORITY { kblow001_h3, kblow001_h3 on
kb-test-har-003, kblow002_h3, kblow002_h3 on kb-test-har-004 }\n\t- HIGHPRIORITY { kbhigh_h3, kbhigh_h3 on kb-test-har-004, sbhigh_h3, sbhigh_h3 on
sb-test-har-001 }\n", "physicalLocation":"K", "machineName":"kb-test-adm-
001.kb.dk", "httpPort":"","applicationInstanceId":"","harvestPriority":"","archiveReplicaName":"KBN", "applicationName":"dk.netarkivet.harvester.scheduler
.HarvestJobManagerApplication", "index":"0"}, {"logMessage":"2017-09-08 08:48:28.136 [http-nio-8078-exec-5] INFO
dk.netarkivet.monitor.jmx.HostForwarding.<init> - Constructing a HostForwarding object for query dk.netarkivet.common.logging:* on Mbeanserver
com.sun.jmx.mbeanserver.JmxMBeanServer@188834e\r\n", "physicalLocation":"K", "machineName":"kb-test-adm-
001.kb.dk", "httpPort":"8078", "applicationInstanceId":"","harvestPriority":"","archiveReplicaName":"KBN", "applicationName":"dk.netarkivet.common.webinter
face.GUIApplication", "index":"0"}, {"logMessage":"2017-09-08 08:40:27.457 [imqConsumerReader-0-1705560651236899840-0] INFO
dk.netarkivet.harvester.heritrix3.HarvestControllerServer.beginListeningIfSpaceAvailable - Starts to listen to new jobs on queue '[Queue
'CSR COMMON JOB SNAPSHOT LOWPRIORITY']\n", "physicalLocation":"K", "machineName":"kb-test-har-
003.kb.dk", "httpPort":"","applicationInstanceId":"kblow001_h3", "harvestPriority":"LOWPRIORITY", "archiveReplicaName":"KBN", "applicationName":"dk.netarkiv
et.harvester.heritrix3.HarvestControllerApplication", "index":"0"}, {"logMessage":"2017-09-08 08:40:34.612 [imqConsumerReader-0-1705560651238699264-0]
INFO dk.netarkivet.harvester.heritrix3.HarvestControllerServer.beginListeningIfSpaceAvailable - Starts to listen to new jobs on queue '[Queue
'CSR COMMON JOB PARTIAL HIGHPRIORITY']\n", "physicalLocation":"K", "machineName":"kb-test-har-
004.kb.dk", "httpPort":"","applicationInstanceId":"kbhigh_h3", "harvestPriority":"HIGHPRIORITY", "archiveReplicaName":"KBN", "applicationName":"dk.netarkive
t.harvester.heritrix3.HarvestControllerApplication", "index":"0"}, {"logMessage":"2017-09-08 08:40:34.594 [imqConsumerReader-0-1705560651238683904-0]
```

Now Just Write a Modern Dynamic Interactive WebGui ...

On advice I started with jQuery.

1. As before, created a new web-context for the GUI in the existing NAS tomcat - although any web-server will do
2. Enabling jQuery just means adding some scripts in your html body and some css in your html head

```
<head>
  <meta charset="utf-8" />
  <link rel="stylesheet" type="text/css"
href="//cdn.datatables.net/1.10.16/css/jquery.dataTables.css">
  <link rel="stylesheet"
href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">
  .
  .
  .
</head>

<body>
  .
  .
  .
<script src="//code.jquery.com/jquery-1.11.2.min.js"></script>
<script src="//code.jquery.com/jquery-migrate-1.2.1.min.js"></script>
<script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
<script type="text/javascript" charset="utf8"
src="https://cdn.datatables.net/1.10.16/js/jquery.dataTables.min.js">
</script>
<script src="site.js"></script>
</body>
```

3. The html for the Status page now looks like this

```
<table id="status_table">
  <thead></thead>
  <tbody></tbody>
</table>
```

... literally, that's the whole thing, because all the action happens in the site.js:

```

$(document).ready(function() {
  $('#tabs').tabs({
    load: function( event, ui ) {
      if(event.currentTarget != null &&
event.currentTarget.firstChild.textContent === "Status") {
        $('#status_table').DataTable( {
          ajax: {
            url: '../rest/rest/status/all',
            dataSrc: ''
          },
          columns: [
            { data: 'physicalLocation' },
            { data: 'applicationName' },
            { data: 'logMessage' }
          ]
        } );
      }
    }
  });
});

```

What matters here is that we are leveraging the entire functionality of the jQuery-UI DataTable plugin to format the JSON we get from the REST service.

4. And here's what it looks like <http://kb-test-adm-001.kb.dk:8078/webgui/>

Selective Harvest x Netarchive Status x http://kb-test...st/status/all x +

kb-test-adm-001.kb.dk:8078/webgui/ Search ☆ ↴ ⌂ 🔍 🌐 ☰

HarvestDefinition Status

hd

Selective Harvest x Netarchive Status x http://kb-test...st/status/all x +

kb-test-adm-001.kb.dk:8078/webgui/

HarvestDefinition Status

Show 10 entries Search:

K	dk.netarkivet.archive.bitarchive.BitarchiveApplication	2017-09-08 08:39:48.225 [main] INFO dk.netarkivet.archive.bitarchive.distribute.BitarchiveServer. - Broadcasting heartbeats on: CSR_KB_THE_BAMON
K	dk.netarkivet.archive.bitarchive.BitarchiveApplication	2017-09-08 08:40:12.734 [main] INFO dk.netarkivet.archive.bitarchive.distribute.BitarchiveServer. - Broadcasting heartbeats on: CSR_KB_THE_BAMON
K	dk.netarkivet.archive.bitarchive.BitarchiveApplication	2017-09-08 08:40:12.734 [main] INFO dk.netarkivet.archive.bitarchive.distribute.BitarchiveServer. - Broadcasting heartbeats on: CSR_KB_THE_BAMON
K	dk.netarkivet.archive.bitarchive.BitarchiveApplication	2017-09-08 08:40:12.734 [main] INFO dk.netarkivet.archive.bitarchive.distribute.BitarchiveServer. - Broadcasting heartbeats on: CSR_KB_THE_BAMON
K	dk.netarkivet.archive.checksum.ChecksumFileApplication	2017-09-08 08:40:42.877 [main] INFO dk.netarkivet.archive.checksum.distribute.ChecksumFileServer. - ChecksumFileServer '130.226.228.5' initialised. Using precomputedChecksums during upload: false
K	dk.netarkivet.harvester.indexserver.IndexServerApplication	2017-09-08 08:40:43.541 [Timer-1] INFO d.netarkivet.harvester.indexserver.distribute.IndexRequestServer.run - Enabling listening to the indexserver channel [Queue 'CSR_COMMON_INDEX_SERVER']
K	dk.netarkivet.viewerproxy.ViewerProxyApplication	2017-09-08 08:40:43.211 [main] INFO dk.netarkivet.viewerproxy.WebProxy. - Starting viewerproxy jetty on port 8078
K	dk.netarkivet.archive.arcrepository.ArcRepositoryApplication	2017-09-08 08:39:22.340 [main] INFO dk.netarkivet.archive.arcrepository.ArcRepository. - Starting the ArcRepository
K	dk.netarkivet.archive.bitarchive.BitarchiveMonitorApplication	2017-09-08 08:40:12.658 [mqConsumerReader-0-1705560651219968256-0] INFO dk.netarkivet.archive.bitarchive.BitarchiveMonitor.signOfLife - Bitarchive '172.17.0.56_BitApp_3' is now known by the bitarchive monitor
K	dk.netarkivet.archive.bitarchive.BitarchiveMonitorApplication	2017-09-08 08:40:54.292 [mqConsumerReader-0-1705560651219977984-0] INFO dk.netarkivet.archive.bitarchive.BitarchiveMonitor.signOfLife - Bitarchive '130.225.27.141' is now known by the bitarchive monitor

Showing 1 to 10 of 19 entries Previous 1 2 Next

e

Note that even with absolutely minimal configuration and styling we get

- Pagination
- Column sorting
- Search

None of which we have now.

So Where To Now

... or can I really ask the Steering Committee for permission to start a complete frontend rewrite?

The Good Stuff

1. This is the result of three days' work by one developer, of which two days were on the backend. Getting started with this web-development stuff isn't as hard as I expected.
2. We can leverage a universe of existing solutions to instead of having to reinvent the wheel.
3. A clean API should make testing NAS much easier.
4. This is already way more fun than hacking through a mess of servlets & jsp and should make it easier to integrate new developers in the project.

The Challenges (not all technical!)

1. Known unknowns and unknown unknowns - we need a much longer pilot-program to be able to present a really convincing case. We haven't even looked at interactions yet.
2. Is jQuery the right technology?
3. How much reskilling do we need to do in the NAS developer group?
4. What about "higher level" skills - how to manage, structure, test, standardise a web-development project? These are not skills we currently have in NAS.

Code for this project is available in a branch of the main netarchivesuite github. As of presentation-day, see this commit: <https://github.com/netarchivesuite/netarchivesuite/commit/b02a8773e6b1a6f20cd57fe0099885406a0ea179>