

# Concept - Optimize M5 memory consumption

## Status Quo

Running performance tests against M5 (beta2) we got the impression that M5 is consuming quite a lot of memory per user. That's why we wanted to optimize.

## Analysed Components

### ContentApps + containers

#### tree

- is not using paging - reads all child nodes when expanding
- inherits `AbstractJcrContainerupdateSize()`
  - required to write number of items to footer (although that's not the proper number)
    - we decided to drop displaying number of items in footer for now

#### list

- queries for all elements to get total size - then queries the content using a paging mechanism

#### thumbnail

- it's presenter triggers a `container#refresh` on start already - that results in already querying for all items
  - can safely be dropped
- doesn't extend `AbstractJcrContainer`, doesn't use its paging mechanism
- has its own lazy loading mechanism: loads when required but then keeps all read items (images) until we switch to another view

## Setup

For the analysis we used the contacts app. There we added 10'000 contacts all having a random street (2000 chars), city (100 chars) + organization name (20'000 chars) as well as one identical image (11kbyte). See attached script for further details.

## Tracking

[MGNLUI-1491](#)

## Results

- starting contacts app consumes 20-30 MByte of additional memory - to be analyzed what's happening there
- could not measure a diff between querying "select \*..." of "select [[jcr:uuid]..." in `AbstractJcrContainer#updateSize()`
- list view temporarily consumes about 100 MByte but this goes away on GC - this is true even when scrolling around
- thumbnail view easily consumes about 200 MByte when scrolling in it - system gets pretty slow
- `JcrItemAdapters` are quite transient - they're used when populating content views but once data is read, we only reference few instances

## Outlook

- we'll reduce number of items in queries (e.g. to 1000)
- lists will only show this number of items
- we need to somehow indicate there might be more items in repo
- if u'r looking for something specific u can always use search view - this one again returns reduced number of items