EDIT - feature request #9134

Rule based creation of additional media representations "on the fly"

07/08/2020 05:29 PM - Andreas Kohlbecker

Status:	Closed	**Start date:**	
Priority:	Priority14	**Due date:**	
Assignee:	Andreas Kohlbecker	**% Done:**	100%
Category:	cdmlib-remote	**Estimated time:**	0:00 hour
Target version:	Release 5.18		
Severity:	normal		

Description

Often only the full sized image URIs are recorded in a db but thumbnail sized images are often required.

Based on a set of rules the missing media representation can be created at request time.

As the MediaRepresentations are created as purely volatile objects they must not be persised. To prevent from erroneously storing them as part of an cdm entity graph the classes that produce these new MediaRepresentations should only reside in cdmlib-remote.

All method in cdmlib-remote which return Media or MediaRepresentation should be equipped with this functionality.

Other methods which also make use of the according MediaUtil methods can not be adapted:

- MediaUtils.findBestMatchingRepresentation()
  - eu.etaxonomy.cdm.api.service.ClassificationServiceImpl.getAllMediaForChildNodes(TaxonNode taxonNode, List propertyPaths, int size, int height, int widthOrDuration, String[] mimeTypes)

Related issues:

Related to EDIT - feature request #9135: All media related web service end po...	**New**
Related to EDIT - task #9132: Update cyprus images to Scaler default API and ...	**In Progress**
Related to EDIT - task #9160: Avoid modified media entities in IMediaToolbox....	**New**
Related to EDIT - task #9203: Filter out generated MediaRepresentations if a ...	**New**

Associated revisions

Revision 2a4ebda1 - 07/09/2020 11:11 AM - Andreas Kohlbecker

fix #9134 MediaUriTransformation for rule based creation of additional media representations 'on the fly'

Revision 2f0cabf1 - 07/09/2020 11:26 AM - Andreas Kohlbecker

fix #9134 solving merge conflict and avoiding field solving serialization in SearchReplace

Revision 551c4f43 - 07/13/2020 01:12 PM - Andreas Kohlbecker

ref #9134 adding ducumentation for MediaUriTransformationProcessor and Mediatoolbox

Revision 193bfc9a - 07/14/2020 09:10 PM - Andreas Müller

ref #9134 fix transformation for non-square maxextend transformations

Revision 8165e2b9 - 07/16/2020 01:24 PM - Andreas Kohlbecker

ref #9134 no longer storing DefaultMediaTransformations in the db, better exception handling and more documentation

Revision 5ccd238c - 07/16/2020 04:07 PM - Andreas Kohlbecker

ref #9134 fixing bug with adding entiies to a collection

Revision 896e7ef8 - 07/16/2020 05:47 PM - Andreas Kohlbecker

ref #9134 fixing calculation of croped thumbnail images and adding tests

Revision bf5abe7d - 07/16/2020 06:07 PM - Andreas Kohlbecker

ref #9134 fixing bug in default transformation creation

Revision 8e996490 - 07/20/2020 01:52 PM - Andreas Kohlbecker

ref #9134 preventing persistance of volatile modified media

Revision 1f68b26c - 08/18/2020 03:20 PM - Andreas Kohlbecker

ref #9134 MediaUriTransformation with flag indicating maxExtend mode

Revision 6348072d - 08/19/2020 10:23 PM - Andreas Kohlbecker

ref #9134 prevent modified media from being saved

Revision 4c4eace6 - 08/19/2020 10:40 PM - Andreas Kohlbecker

ref #9134 documenting SearchReplace

Revision 41ed639d - 11/03/2020 05:42 PM - Andreas Kohlbecker

ref #9134 improving regex in DefaultMediaTransformations for digilib

History

#1 - 07/08/2020 05:34 PM - Andreas Kohlbecker

- Description updated

- Category changed from cdmlib to cdmlib-remote

#2 - 07/08/2020 05:35 PM - Andreas Kohlbecker

- Description updated

#3 - 07/08/2020 05:35 PM - Andreas Kohlbecker

- Description updated

#4 - 07/08/2020 05:37 PM - Andreas Kohlbecker

- Description updated

#5 - 07/09/2020 11:11 AM - Andreas Kohlbecker

- Status changed from New to Resolved

- % Done changed from 0 to 50

Applied in changeset cdmlib|2a4ebda1f53750568311f67326e16fb4e3d7fa2a.

#6 - 07/09/2020 11:20 AM - Andreas Kohlbecker

A set of generic transformation rules for the BGBM digilib server rules will be stored in each db. This has been done for two reasons:

1. this allows for zero configuration usage of the digilib server
2. serves as a example or template on which other rules can be based on.

```
[
  {
    "height": 200,
    "width": 200,
    "mimeType": "image/jpeg",
    "pathQueryFragment": {
      "search": "digilib/Scaler/IIIF/([^\\!]+)\\!([^\\/]+)(.*)",
      "replace": "digilib/Scaler/IIIF/$1!$2/full/!200,200/0/default.jpg"
    },
    "host": {
      "search": "pictures.bgbm.org",
      "replace": "pictures.bgbm.org"
    },
    "scheme": null
  },
  {
    "height": 400,
    "width": 400,
    "mimeType": "image/jpeg",
    "pathQueryFragment": {
      "search": "digilib/Scaler/IIIF/([^\\!]+)\\!([^\\/]+)(.*)",
```

```
      "replace": "digilib/Scaler/IIIF/$1!$2/full/!400,400/0/default.jpg"
    },
    "host": {
      "search": "pictures.bgbm.org",
      "replace": "pictures.bgbm.org"
    },
    "scheme": null
  },
  {
    "height": 400,
    "width": 400,
    "mimeType": "image/jpeg",
    "pathQueryFragment": {
      "search": "digilib/Scaler/\\?fn=([^\\\\/]+)/(\\w+)(.*)",
      "replace": "digilib/Scaler/IIIF/$1!$2/full/!400,400/0/default.jpg"
    },
    "host": {
      "search": "pictures.bgbm.org",
      "replace": "pictures.bgbm.org"
    },
    "scheme": null
  },
  {
    "height": 200,
    "width": 200,
    "mimeType": "image/jpeg",
    "pathQueryFragment": {
      "search": "digilib/Scaler/\\?fn=([^\\\\/]+)/(\\w+)(.*)",
      "replace": "digilib/Scaler/IIIF/$1!$2/full/!200,200/0/default.jpg"
    },
    "host": {
      "search": "pictures.bgbm.org",
      "replace": "pictures.bgbm.org"
    },
    "scheme": null
  }
]
```


#7 - 07/09/2020 11:28 AM - Andreas Kohlbecker

- Assignee changed from Andreas Kohlbecker to Andreas Müller

please review

#8 - 07/09/2020 11:38 AM - Andreas Kohlbecker

- Related to feature request #9135: All media related web service end points with MediaUriTransformation and/or bestMatchingRepresentation filter added

#9 - 07/09/2020 03:24 PM - Andreas Kohlbecker

TODO: Skip the creation of a MediaRepresentation if a part with exactly the same URL already exists.

#10 - 07/14/2020 09:12 PM - Andreas Müller

- Status changed from Resolved to Feedback

- Assignee changed from Andreas Müller to Andreas Kohlbecker

As far as I can see this works well.

Some minor issues:

- The MediaToolbox methods change the media itself by adding or removing representations. This might be dangerous in case the ToolBox is used whithin a not read-only transaction. Currently the toolbox belongs and is used only in remote so it is probably not used in such a way. However, to be on the save side it might be a good idea to use Media clones instead.
- MediaToolbox: shouldn't we add a @Component annotation to show that it is meant to be a bean (even if not searched by component scan)?
- documentation for MediaToolbox is still missing
- MediaToolbox is only logging JsonExceptions but not throwing them. Is this wanted (I am not familiar with the exception handling in remote therefore I ask)

- MediaUriTransformationProcessor.calculateTargetSize() does not handle maxextend transformations with trans.getWidth() != trans.getHeight() really correct by only returning "return new Point(trans.getWidth(), trans.getHeight());" . The algorithm should not depend on the fact that the required box is a square. I adapted the code accordingly (it even became simpler this way) and added tests. (193bfc9a54d2)

#11 - 07/14/2020 09:14 PM - Andreas Müller

There is one issue that needs to be solved:

- MediaToolbox.readTransformations is creating default preference values and stores them in the database. This is not the correct way to handle preferences. Default values should be handled in the PreferencePredicate itself. This way the database is not littered by default value records. Also this way it is easier to change the default behavior without having to update database records. Simply by changing the code. So preferences should only be stored in the database if they deviate from the default value.

#12 - 07/14/2020 09:18 PM - Andreas Müller

- Subject changed from rule based creation of additional media representations "on the fly" to Rule based creation of additional media representations "on the fly"

#13 - 07/15/2020 04:44 PM - Andreas Kohlbecker

- Related to task #9132: Update cyprus images to Scaler default API and add thumbnails added

#15 - 07/16/2020 01:00 PM - Andreas Kohlbecker

new default transformation for digilib:

```
[ {
  "scheme" : null,
  "host" : {
    "replace" : "pictures.bgbm.org",
    "search" : "pictures.bgbm.org"
  },
  "pathQueryFragment" : {
    "replace" : "digilib/Scaler/?fn=$1/$2&mo=crop&dw=200&dh=147&uvfix=1",
    "search" : "digilib/Scaler/IIIF/([^\\!]+)\\!([^\\/]+)(.*)"
  },
  "mimeType" : "image/jpeg",
  "width" : 200,
  "height" : 200
}, {
  "scheme" : null,
  "host" : {
    "replace" : "pictures.bgbm.org",
    "search" : "pictures.bgbm.org"
  },
  "pathQueryFragment" : {
    "replace" : "digilib/Scaler/IIIF/$1!$2/full/!400,400/0/default.jpg",
    "search" : "digilib/Scaler/IIIF/([^\\!]+)\\!([^\\/]+)(.*)"
  },
  "mimeType" : "image/jpeg",
  "width" : 400,
  "height" : 400
}, {
  "scheme" : null,
  "host" : {
    "replace" : "pictures.bgbm.org",
    "search" : "pictures.bgbm.org"
  },
  "pathQueryFragment" : {
    "replace" : "digilib/Scaler/IIIF/$1!$2/full/!400,400/0/default.jpg",
    "search" : "digilib/Scaler/\\?fn=([^\\\\/]+)/(\\w+)(.*)"
  },
  "mimeType" : "image/jpeg",
  "width" : 400,
  "height" : 400
}, {
  "scheme" : null,
  "host" : {
    "replace" : "pictures.bgbm.org",
    "search" : "pictures.bgbm.org"
  },
  "pathQueryFragment" : {
    "replace" : "digilib/Scaler/?fn=$1/$2&mo=crop&dw=200&dh=147&uvfix=1",
```

```
    "search" : "digilib/Scaler/\\?fn=([^\\\\/]+)/(\\w+)(.*)"
  },
  "mimeType" : "image/jpeg",
  "width" : 200,
  "height" : 200
} ]
```

#16 - 07/16/2020 03:39 PM - Andreas Müller

Does it make sense to use

"width" : 200,
"height" : 200

if we use &mo=crop&dw=200&dh=**147**

Do we need width and heigth at all if we use crop?

#17 - 07/16/2020 03:50 PM - Andreas Kohlbecker

Andreas Müller wrote:

> ...
>
> - The MediaToolbox methods change the media itself by adding or removing representations. This might be dangerous in case the ToolBox is used whithin a not read-only transaction. Currently the toolbox belongs and is used only in remote to it is probably not used in such a way. However, to be on the save side it might be a good idea to use Media clones instead.

Using media clones can also cause problems: Rest service clients like the data portals may rely on the uuid of the media object as being a stable identifier. Cloned objects would have a different UUIDs than the original. Cloned objects with the same UUID - also not a good idea. May be we could set Media.id = 0, this would most probably cause an error when hibernate attempts to persist such an entity (not tested).

Another option is to return MediaDTOs instead of Media entities. It consider this as the best solution since it is more transparent than all others.

I implemented all of your suggestions and discovered a bug in my code:

```
media.getRepresentations().addAll(newRepr);
```

can you see the problem?

getRepresentations() must return an unmodifiableSet to avoid errors as above. I think we should protect all collections of entities where the add() and remove() methods do more than collection.add(), collection.addAll(), collection.remove(), etc.

#18 - 07/16/2020 03:53 PM - Andreas Kohlbecker

Andreas Müller wrote:

> Does it make sense to use
>
> "width" : 200,
> "height" : 200
>
> if we use &mo=crop&dw=200&dh=**147**
>
> Do we need width and heigth at all if we use crop?

I guess you mean the ./!400,400/0/default.jpg URLs? Theses are for the preview images in the taxon general page.

#19 - 07/16/2020 04:09 PM - Andreas Müller

Andreas Kohlbecker wrote:

> Andreas Müller wrote:
>
> > Does it make sense to use
> >
> > "width" : 200,
> > "height" : 200
> >
> > if we use &mo=crop&dw=200&dh=**147**
> >
> > Do we need width and heigth at all if we use crop?

> I guess you mean the ./!400,400/0/default.jpg URLs? Theses are for the preview images in the taxon general page.

Why 400? I know they are for the profile image. But above I mentioned

> "width" : 200,
> "height" : 200
> if we use &mo=crop&dw=200&dh=**147**

so we talk about the thumbnails here. To me it sounds strange we have height=200 but dh=147 . But I did not go into detail. They might be a reason for this.

And the second issue was that I was wondering if we do need the width and heigth parameter at all for the algorithm if we use crop because crop will cut the image anyway to the size that is needed. But also here I did not really check against the code. It just came into my mind that this could be something that is not needed anymore with crop.

#20 - 07/16/2020 04:28 PM - Andreas Müller

Andreas Kohlbecker wrote:

> Andreas Müller wrote:
>
> Using media clones can also cause problems: Rest service clients like the data portals may rely on the uuid of the media object as being a stable identifier. Cloned objects would have a different UUIDs than the original. Cloned objects with the same UUID - also not a good idea. May be we could set Media.id = 0, this would most probably cause an error when hibernate attempts to persist such an entity (not tested).

Hmm, true, using a clone does not fully solve the problem, as long as we do not make sure that we are out of an transaction. However, CdmBase.clone() sets the id of the new object to 0 anyway, so it is correct that setting the uuid to the old value will throw an exception if anyone will try to use this object within a transaction. So I think it is not dangerous to do it this way. However, a DTO might also be a good (and maybe more performant) solution but will take more time to implement.

#21 - 07/16/2020 04:42 PM - Andreas Müller

Andreas Kohlbecker wrote:

> Andreas Müller wrote:
>
> I implemented all of your suggestions and discovered a bug in my code:
>
> ```
> media.getRepresentations().addAll(newRepr);
> ```
>
> can you see the problem?
>
> getRepresentations() must return an unmodifiableSet to avoid errors as above. I think we should protect all collections of entities where the add() and remove() methods do more than collection.add(), collection.addAll(), collection.remove(), etc.

Yes, it is a known issue (discussed at the very early days) that in this context it is a bit dangerous to return the collections themself. I am not sure anymore why we decided to not do it the immutable way. One reason could be that in some cases you want to check if a collection is a PersistentCollection. You can't do this from outside if the collection that you get back is not the collection that is really attached to the object (except for using reflections which is a bit dirty and not performant). However, as this was never really an issue over the last 15 years (I remember only 1-2 cases where developers did this wrong) so I guess this is not really critical.

By the way, it can also be dangerous the other way round. E.g. we have the code

```
this.getRepresentations().remove(representation);
```

in Media.removeRepresentation(). This will not result in the expected behavior if getRepresentations() will not return the real set. Sometimes IDEs automatically do replace assignments by using getters and could create such problems.

#22 - 07/16/2020 05:56 PM - Andreas Kohlbecker

Andreas Müller wrote:

- MediaUriTransformationProcessor.calculateTargetSize() does not handle maxextend transformations with trans.getWidth() != trans.getHeight() really correct by only returning "return new Point(trans.getWidth(), trans.getHeight());" . The algorithm should not depend on the fact that the required box is a square. I adapted the code accordingly (it even became simpler this way) and added tests. (193bfc9a54d2)

I actually don't expect that we will use maxextend transformations with trans.getWidth() != trans.getHeight() in the dataportal. I can't see the use case for this. More importantly for us is cropping of images. I modified the size calculation so that it works like:

- trans.getWidth() != trans.getHeight() ==> crop
- trans.getWidth() == trans.getHeight() ==> max extend

I adapted the test class also.

To cover all cases we would need to add a ~~mode enum field to the MediaUriTransformation class with SCALE, MAX_EXTEND, CROP~~ boolean field for isMaxextend

#23 - 07/16/2020 06:13 PM - Andreas Kohlbecker

Andreas Müller wrote:

> "width" : 200,
> "height" : 200
> if we use &mo=crop&dw=200&dh=**147**

so we talk about the thumbnails here. To me it sounds strange we have height=200 but dh=147 . But I did not go into detail. They might be a reason for this.

And the second issue was that I was wondering if we do need the width and heigth parameter at all for the algorithm if we use crop because crop will cut the image anyway to the size that is needed. But also here I did not really check against the code. It just came into my mind that this could be something that is not needed anymore with crop.

Now I got what you are referring to.

1. There actually was a bug in the code creating the default transformations - this is fixed, see bf5abe7ddd
2. The crop function crops the image to the specified size, that it to the width and height. The resulting image exactly has this width and size. There is no other way to define what is needed than defining the desired size.

#24 - 07/16/2020 06:14 PM - Andreas Kohlbecker

the fixed default transformations as json for reference:


```
[ {
  "scheme" : null,
  "host" : {
    "replace" : "pictures.bgbm.org",
    "search" : "pictures.bgbm.org"
  },
  "pathQueryFragment" : {
    "replace" : "digilib/Scaler/?fn=$1/$2&mo=crop&dw=200&dh=147&uvfix=1",
    "search" : "digilib/Scaler/IIIF/([^\\!]+)\\!([^\\/]+)(.*)"
  },
  "mimeType" : "image/jpeg",
  "width" : 200,
  "height" : 147
}, {
  "scheme" : null,
  "host" : {
    "replace" : "pictures.bgbm.org",
    "search" : "pictures.bgbm.org"
  },
  "pathQueryFragment" : {
    "replace" : "digilib/Scaler/IIIF/$1!$2/full/!400,400/0/default.jpg",
    "search" : "digilib/Scaler/IIIF/([^\\!]+)\\!([^\\/]+)(.*)"
  },
  "mimeType" : "image/jpeg",
  "width" : 400,
  "height" : 400
}, {
  "scheme" : null,
  "host" : {
    "replace" : "pictures.bgbm.org",
    "search" : "pictures.bgbm.org"
  },
  "pathQueryFragment" : {
    "replace" : "digilib/Scaler/IIIF/$1!$2/full/!400,400/0/default.jpg",
    "search" : "digilib/Scaler/\\?fn=([^\\\\/]+)/(\\w+)(.*)"
  },
  "mimeType" : "image/jpeg",
  "width" : 400,
  "height" : 400
}, {
```

```
  "scheme" : null,
  "host" : {
    "replace" : "pictures.bgbm.org",
    "search" : "pictures.bgbm.org"
  },
  "pathQueryFragment" : {
    "replace" : "digilib/Scaler/?fn=$1/$2&mo=crop&dw=200&dh=147&uvfix=1",
    "search" : "digilib/Scaler/\\?fn=([^\\\\/]+)/(\\w+)(.*)"
  },
  "mimeType" : "image/jpeg",
  "width" : 200,
  "height" : 147
} ]
```

#25 - 07/20/2020 02:05 PM - Andreas Kohlbecker

- Status changed from Feedback to Resolved

- Assignee changed from Andreas Kohlbecker to Andreas Müller

Andreas Müller wrote:

> Andreas Kohlbecker wrote:
>
> > Andreas Müller wrote:
> >
> > Using media clones can also cause problems: Rest service clients like the data portals may rely on the uuid of the media object as being a stable identifier. Cloned objects would have a different UUIDs than the original. Cloned objects with the same UUID - also not a good idea. May be we could set Media.id = 0, this would most probably cause an error when hibernate attempts to persist such an entity (not tested).
>
> Hmm, true, using a clone does not fully solve the problem, as long as we do not make sure that we are out of an transaction. However, CdmBase.clone() sets the id of the new object to 0 anyway, so it is correct that setting the uuid to the old value will throw an exception if anyone will try to use this object within a transaction. So I think it is not dangerous to do it this way. However, a DTO might also be a good (and maybe more performant) solution but will take more time to implement.

I modified the code a bit so that there is only one method left which actually modifies media objects. All other methods only operate on MediaRepresentation list. In which case unwanted persisting of modified media objects is not a case to take into account. The last method which modifies Media is IMediaToolbox.filterPreferredMediaRepresentations(List<Media> mediaList, Class<? extends MediaRepresentationPart> type, String[] mimeTypes, Integer widthOrDuration, Integer height, Integer size). This method can be replaced easily and is no source of danger at the same time since is in only being used in the TaxonPortalController. I anyway set the entity id of the modified media entities to 0 to raise an error in case of an attempt to save it to the db.

There now is a new ticket for cleaning up filterPreferredMediaRepresentations(List<Media> mediaList, Class<? extends MediaRepresentationPart> type, String[] mimeTypes, Integer widthOrDuration, Integer height, Integer size): #9160

With this commit I consider this ticket as solved completely.

#26 - 08/13/2020 10:50 AM - Andreas Müller

Andreas Kohlbecker wrote:

> **TODO**: Skip the creation of a MediaRepresentation if a part with exactly the same URL already exists.

Is this fixed already? (I did not check the code)

#27 - 08/13/2020 11:08 AM - Andreas Müller

The class SearchReplace is undocumented.
Also I wonder if we should maybe move it to cdmlib-common as it is context independent (neither dependent on service layer nor dependent on media) but simply defines a replace rule (have you checked if something like this maybe even exists somewhere in Java?).

Personally I would rename the class to ExpressionReplace(ment). To me the term "Search" is not so intuitive. Also the parameters "search" and "searchPattern" I would rename accordingly. But this is my personal opinion. We may keep the name if you do not agree.

#28 - 08/13/2020 11:13 AM - Andreas Müller

- Related to task #9160: Avoid modified media entities in IMediaToolbox.filterPreferredMediaRepresentations() added

#29 - 08/13/2020 11:25 AM - Andreas Müller

Andreas Müller wrote:

- MediaToolbox is only logging JsonExceptions but not throwing them. Is this wanted (I am not familiar with the exception handling in remote therefore I ask)

This still seems to be the case. On purpose or simply forgotten?

#30 - 08/13/2020 11:37 AM - Andreas Müller

Andreas Kohlbecker wrote:

> I modified the code a bit so that there is only one method left which actually modifies media objects. All other methods only operate on MediaRepresentation list. In which case unwanted persisting of modified media objects is not a case to take into account. The last method which modifies Media is IMediaToolbox.filterPreferredMediaRepresentations(List<Media> mediaList, Class<? extends MediaRepresentationPart> type, String[] mimeTypes, Integer widthOrDuration, Integer height, Integer size). This method can be replaced easily and is no source of danger at the same time since is in only being used in the TaxonPortalController. I anyway set the entity id of the modified media entities to 0 to raise an error in case of an attempt to save it to the db.

To me it looks like processAndFilterPreferredMediaRepresentations() also still modifies media objects. There is the code:

```
for(MediaRepresentation r : newReprs) {
    media.addRepresentation(r);
}
```

which definetely modifies the media object. So we should either deprecate this method too or we keep both methods undeprecated and simply document that they may modify the input data and mention that there will be a DTO based method in future which does not do so. For me deprecating a method should only be done if it is either extremely dangerous to use (not really the case here, but maybe true) or if there is already an alternative to use.

#31 - 08/13/2020 12:00 PM - Andreas Müller

- Status changed from Resolved to Feedback

- Assignee changed from Andreas Müller to Andreas Kohlbecker

Andreas Kohlbecker wrote:

> Andreas Müller wrote:
>
> - MediaUriTransformationProcessor.calculateTargetSize() does not handle maxextend transformations with trans.getWidth() != trans.getHeight() really correct by only returning "return new Point(trans.getWidth(), trans.getHeight());" . The algorithm should not depend on the fact that the required box is a square. I adapted the code accordingly (it even became simpler this way) and added tests. (193bfc9a54d2)
>
> I actually don't expect that we will use maxextend transformations with trans.getWidth() != trans.getHeight() in the dataportal. I can't see the use case for this. More importantly for us is cropping of images. I modified the size calculation so that it works like:
>
> - trans.getWidth() != trans.getHeight() ==> crop
> - trans.getWidth() == trans.getHeight() ==> max extend
>
> I adapted the test class also.
>
> To cover all cases we would need to add a ~~mode enum field to the MediaUriTransformation class with SCALE, MAX_EXTEND, CROP~~ boolean field for isMaxextend

I still do not fully understand this. For me crop and maxExtend/scale are independent from the question if the target is a square or a rectangle. Maybe there is no application for a maxextend rectangle but why shouldn't there be an application for a cropped square? Currently you can not have a square which is cropped as far as I can see. But cropped squares are very common in many applications.

I agree that we do need an additional enum then which I think should have values SCALE and CROP. MaxExtend is a special case for scale which is automatically defined by a missing heigth/width so we do not need an enum value for this.

I can implement this if you want.

#32 - 08/18/2020 03:19 PM - Andreas Kohlbecker

- Status changed from Feedback to In Progress

updated of the JSON export of the new defaults with maxExtend:

```
[ {
  "scheme" : null,
  "host" : {
```

```
    "replace" : "pictures.bgbm.org",
    "search" : "pictures.bgbm.org"
  },
  "pathQueryFragment" : {
    "replace" : "digilib/Scaler/?fn=$1/$2&mo=crop&dw=200&dh=147&uvfix=1",
    "search" : "digilib/Scaler/IIIF/([^\\!]+)\\!([^\\/]+)(.*)"
  },
  "mimeType" : "image/jpeg",
  "width" : 200,
  "height" : 147,
  "maxExtend" : false
}, {
  "scheme" : null,
  "host" : {
    "replace" : "pictures.bgbm.org",
    "search" : "pictures.bgbm.org"
  },
  "pathQueryFragment" : {
    "replace" : "digilib/Scaler/IIIF/$1!$2/full/!400,400/0/default.jpg",
    "search" : "digilib/Scaler/IIIF/([^\\!]+)\\!([^\\/]+)(.*)"
  },
  "mimeType" : "image/jpeg",
  "width" : 400,
  "height" : 400,
  "maxExtend" : true
}, {
  "scheme" : null,
  "host" : {
    "replace" : "pictures.bgbm.org",
    "search" : "pictures.bgbm.org"
  },
  "pathQueryFragment" : {
    "replace" : "digilib/Scaler/IIIF/$1!$2/full/!400,400/0/default.jpg",
    "search" : "digilib/Scaler/\\?fn=([^\\\\/]+)/(\\w+)(.*)"
  },
  "mimeType" : "image/jpeg",
  "width" : 400,
  "height" : 400,
  "maxExtend" : true
}, {
  "scheme" : null,
  "host" : {
    "replace" : "pictures.bgbm.org",
    "search" : "pictures.bgbm.org"
  },
  "pathQueryFragment" : {
    "replace" : "digilib/Scaler/?fn=$1/$2&mo=crop&dw=200&dh=147&uvfix=1",
    "search" : "digilib/Scaler/\\?fn=([^\\\\/]+)/(\\w+)(.*)"
  },
  "mimeType" : "image/jpeg",
  "width" : 200,
  "height" : 147,
  "maxExtend" : false
} ]
```


#33 - 08/18/2020 03:25 PM - Andreas Kohlbecker

- Status changed from In Progress to Resolved

- Assignee changed from Andreas Kohlbecker to Andreas Müller

please review

#34 - 08/19/2020 03:55 PM - Andreas Müller

did you also have a look to comments 26 - 30 ?

#35 - 08/19/2020 04:03 PM - Andreas Müller

- Status changed from Resolved to Feedback

- Assignee changed from Andreas Müller to Andreas Kohlbecker

The fix for crop / max extend looks ok

#36 - 08/19/2020 04:07 PM - Andreas Müller

Beside comments 26-30 I would suggest to orally discuss if we really want to have the BGBM image server values as default or if these values should be stored in the according databases (cyprus, cichorieae). In general, the second solution feels more correct, but there might be reasons why to do it the first way. This can be done after the release.

#37 - 08/19/2020 08:42 PM - Andreas Müller

- Target version changed from Release 5.18 to Release 5.17

#38 - 08/19/2020 10:24 PM - Andreas Kohlbecker

Andreas Müller wrote:

> Andreas Kohlbecker wrote:
>
> > I modified the code a bit so that there is only one method left which actually modifies media objects. All other methods only operate on MediaRepresentation list. In which case unwanted persisting of modified media objects is not a case to take into account. The last method which modifies Media is IMediaToolbox.filterPreferredMediaRepresentations(List<Media> mediaList, Class<? extends MediaRepresentationPart> type, String[] mimeTypes, Integer widthOrDuration, Integer height, Integer size). This method can be replaced easily and is no source of danger at the same time since is in only being used in the TaxonPortalController. I anyway set the entity id of the modified media entities to 0 to raise an error in case of an attempt to save it to the db.
>
> To me it looks like processAndFilterPreferredMediaRepresentations() also still modifies media objects. There is the code:

fixed with 6348072d40

```
for(MediaRepresentation r : newReprs) {
    media.addRepresentation(r);
}
```

> which definetely modifies the media object. So we should either deprecate this method too or we keep both methods undeprecated and simply document that they may modify the input data and mention that there will be a DTO based method in future which does not do so. For me deprecating a method should only be done if it is either extremely dangerous to use (not really the case here, but maybe true) or if there is already an alternative to use.

#39 - 08/19/2020 10:40 PM - Andreas Kohlbecker

Andreas Müller wrote:

> The class SearchReplace is undocumented.
> Also I wonder if we should maybe move it to cdmlib-common as it is context independent (neither dependent on service layer nor dependent on media) but simply defines a replace rule (have you checked if something like this maybe even exists somewhere in Java?).
>
> Personally I would rename the class to ExpressionReplace(ment). To me the term "Search" is not so intuitive. Also the parameters "search" and "searchPattern" I would rename accordingly. But this is my personal opinion. We may keep the name if you do not agree.

SearchReplace is specially dedicated to be used in MediaUriTransformation which is serializes to json cdm preference values.
I would prefer to keep it next to MediaUriTransformation. In principle is could be an public static inner class of the latter.

java doc added with 4c4eace63e

#40 - 08/19/2020 10:57 PM - Andreas Kohlbecker

Andreas Müller wrote:

> Andreas Müller wrote:
>
> - MediaToolbox is only logging JsonExceptions but not throwing them. Is this wanted (I am not familiar with the exception handling in remote therefore I ask)
>
> This still seems to be the case. On purpose or simply forgotten?

Nope, postponed, since this is a delicate one. A deserialization problem with the transformations as stored in the cdm preferences would cause all image viewers in the portal going blank. Throwing the exception in the readTransformations() and catching and swallowing it in controller methods would not be better than just logging the exception in the method where it occurs. I was thinking that we would need something like a health report for such errors that are really relevant for the maintainer but which should not break the user experience completely. This health report system could be attached to the logging framework where it would listen for ERROR level messages, collecting, assembling and counting severe events. Reports via UI or by daily email messages.

#41 - 08/19/2020 11:00 PM - Andreas Kohlbecker

- Assignee changed from Andreas Kohlbecker to Andreas Müller

- % Done changed from 50 to 70

I went through comment 26 - 30 please close this issue if you agree with my answers.
The issue pointed out in comment 26 is handled in #9203
Comment 40 may require some further discussion, which should take place in a separate ticket.

#42 - 08/19/2020 11:12 PM - Andreas Kohlbecker

- Related to task #9203: Filter out generated MediaRepresentations if a part with exactly the same URL already exists. added

#43 - 08/20/2020 08:51 AM - Andreas Müller

- Assignee changed from Andreas Müller to Andreas Kohlbecker

Andreas Kohlbecker wrote:

> Andreas Müller wrote:
>
> > Andreas Kohlbecker wrote:
> >
> > > I modified the code a bit so that there is only one method left which actually modifies media objects. All other methods only operate on MediaRepresentation list. In which case unwanted persisting of modified media objects is not a case to take into account. The last method which modifies Media is IMediaToolbox.filterPreferredMediaRepresentations(List<Media> mediaList, Class<? extends MediaRepresentationPart> type, String[] mimeTypes, Integer widthOrDuration, Integer height, Integer size). This method can be replaced easily and is no source of danger at the same time since is in only being used in the TaxonPortalController. I anyway set the entity id of the modified media entities to 0 to raise an error in case of an attempt to save it to the db.
> >
> > To me it looks like processAndFilterPreferredMediaRepresentations() also still modifies media objects. There is the code:
> >
> > ```
> > for(MediaRepresentation r : newReprs) {
> >     media.addRepresentation(r);
> > }
> > ```
> >
> > which definetely modifies the media object. So we should either deprecate this method too or we keep both methods undeprecated and simply document that they may modify the input data and mention that there will be a DTO based method in future which does not do so. For me deprecating a method should only be done if it is either extremely dangerous to use (not really the case here, but maybe true) or if there is already an alternative to use.
> > fixed with 6348072d40
>
> Danger doesn't exist anymore here. Last question: filterPreferredMediaRepresentations was set to deprecated due to the above reasons and a link to #9160 was added in the docu. Shouldn't we do the same for processAndFilterPreferredMediaRepresentations for consistency reasons?

#44 - 08/20/2020 08:53 AM - Andreas Müller

Andreas Kohlbecker wrote:

> Andreas Müller wrote:
>
> > Andreas Müller wrote:
> >
> > - MediaToolbox is only logging JsonExceptions but not throwing them. Is this wanted (I am not familiar with the exception handling in remote therefore I ask)
> >
> > This still seems to be the case. On purpose or simply forgotten?
>
> Nope, postponed, since this is a delicate one. A deserialization problem with the transformations as stored in the cdm preferences would cause all image viewers in the portal going blank. Throwing the exception in the readTransformations() and catching and swallowing it in controller methods would not be better than just logging the exception in the method where it occurs. I was thinking that we would need something like a health report for such errors that are really relevant for the maintainer but which should not break the user experience completely. This health report system could be attached to the logging framework where it would listen for ERROR level messages, collecting, assembling and counting severe events. Reports via UI or by daily email messages.

Should we create a new ticket for this and link to this note?

#45 - 08/20/2020 08:55 AM - Andreas Müller

- % Done changed from 70 to 80

Generally this ticket can be closed now (though I still don't understand wyh SearchReplace is context specific, but this is a minor issue and does not need to be further discussed).

#46 - 11/02/2020 01:57 PM - Andreas Kohlbecker

- % Done changed from 80 to 100

Ok agreed, I am closing this ticket. Please open a new one for further discussion when needed.

#47 - 11/02/2020 02:31 PM - Andreas Kohlbecker

posting the SQL INSERT statement to add the default transformation to a DB:

```
INSERT INTO CdmPreference
(key_subject, key_predicate, value, allowOverride)
VALUES('/', 'media.representationTransformations',
'[{"scheme":null,"host":{"replace":"pictures.bgbm.org","search":"pictures.bgbm.org"},"pathQueryFragment":{"rep
lace":"digilib/Scaler/?fn=$1/$2&mo=crop&dw=200&dh=147&uvfix=1","search":"digilib/Scaler/IIIF/([^\\\\!]+)\\\\
!([^\\\\
/]+)(.*)"},"mimeType":"image/jpeg","width":200,"height":147,"maxExtend":false},{"scheme":null,"host":{"replace
":"pictures.bgbm.org","search":"pictures.bgbm.org"},"pathQueryFragment":{"replace":"digilib/Scaler/IIIF/$1!$2/
full/!400,400/0/default.jpg","search":"digilib/Scaler/IIIF/([^\\\\!]+)\\\\!([^\\\\
/]+)(.*)"},"mimeType":"image/jpeg","width":400,"height":400,"maxExtend":true},{"scheme":null,"host":{"replace"
:"pictures.bgbm.org","search":"pictures.bgbm.org"},"pathQueryFragment":{"replace":"digilib/Scaler/IIIF/$1!$2/f
ull/!400,400/0/default.jpg","search":"digilib/Scaler/\\\\?fn=([^\\\\\\\\/]+)/(\\\\
w+)(.*)"},"mimeType":"image/jpeg","width":400,"height":400,"maxExtend":true},{"scheme":null,"host":{"replace":
"pictures.bgbm.org","search":"pictures.bgbm.org"},"pathQueryFragment":{"replace":"digilib/Scaler/?fn=$1/$2&mo=
crop&dw=200&dh=147&uvfix=1","search":"digilib/Scaler/\\\\?fn=([^\\\\\\\\/]+)/(\\\\
w+)(.*)"},"mimeType":"image/jpeg","width":200,"height":147,"maxExtend":false}]', 1);
```

#48 - 11/03/2020 05:45 PM - Andreas Kohlbecker

I improved the regex in the DefaultMediaTransformations for digilib, this is the new default:

```
[ {
  "scheme" : null,
  "host" : {
    "replace" : "pictures.bgbm.org",
    "search" : "pictures.bgbm.org"
  },
  "pathQueryFragment" : {
    "replace" : "digilib/Scaler/?fn=$1/$2&mo=crop&dw=200&dh=147&uvfix=1",
    "search" : "digilib/Scaler/IIIF/([^\\!]+)\\!([^\\/]+)(.*)"
  },
  "mimeType" : "image/jpeg",
  "width" : 200,
  "height" : 147,
  "maxExtend" : false
}, {
  "scheme" : null,
  "host" : {
    "replace" : "pictures.bgbm.org",
    "search" : "pictures.bgbm.org"
  },
  "pathQueryFragment" : {
    "replace" : "digilib/Scaler/IIIF/$1!$2/full/!400,400/0/default.jpg",
    "search" : "digilib/Scaler/IIIF/([^\\!]+)\\!([^\\/]+)(.*)"
  },
  "mimeType" : "image/jpeg",
  "width" : 400,
  "height" : 400,
  "maxExtend" : true
}, {
  "scheme" : null,
  "host" : {
    "replace" : "pictures.bgbm.org",
    "search" : "pictures.bgbm.org"
  },
  "pathQueryFragment" : {
    "replace" : "digilib/Scaler/IIIF/$1!$2/full/!400,400/0/default.jpg",
```

```
    "search" : "digilib/Scaler/?\\?fn=([^\\\\/]+)/(\\w+)(.*)"
  },
  "mimeType" : "image/jpeg",
  "width" : 400,
  "height" : 400,
  "maxExtend" : true
}, {
  "scheme" : null,
  "host" : {
    "replace" : "pictures.bgbm.org",
    "search" : "pictures.bgbm.org"
  },
  "pathQueryFragment" : {
    "replace" : "digilib/Scaler/?fn=$1/$2&mo=crop&dw=200&dh=147&uvfix=1",
    "search" : "digilib/Scaler/?\\?fn=([^\\\\/]+)/(\\w+)(.*)"
  },
  "mimeType" : "image/jpeg",
  "width" : 200,
  "height" : 147,
  "maxExtend" : false
} ]
```


as SQL INSERT stanement:


```
INSERT INTO CdmPreference
(key_subject, key_predicate, value, allowOverride)
VALUES('/',
'[ { "scheme" : null, "host" : { "replace" : "pictures.bgbm.org", "search" : "pictures.bgbm.org" }, "pathQuery
Fragment" : { "replace" : "digilib/Scaler/?fn=$1/$2&mo=crop&dw=200&dh=147&uvfix=1", "search" : "digilib/Scaler
/IIIF/([^\\!]+)\\!([^\\
/]+)(.*)" }, "mimeType" : "image/jpeg", "width" : 200, "height" : 147, "maxExtend" : false }, { "scheme" : nul
l, "host" : { "replace" : "pictures.bgbm.org", "search" : "pictures.bgbm.org" }, "pathQueryFragment" : { "repl
ace" : "digilib/Scaler/IIIF/$1!$2/full/!400,400/0/default.jpg", "search" : "digilib/Scaler/IIIF/([^\\!]+)\\
!([^\\
/]+)(.*)" }, "mimeType" : "image/jpeg", "width" : 400, "height" : 400, "maxExtend" : true }, { "scheme" : null
, "host" : { "replace" : "pictures.bgbm.org", "search" : "pictures.bgbm.org" }, "pathQueryFragment" : { "repla
ce" : "digilib/Scaler/IIIF/$1!$2/full/!400,400/0/default.jpg", "search" : "digilib/Scaler/?\\?fn=([^\\\\/]+)/(
\\
w+)(.*)" }, "mimeType" : "image/jpeg", "width" : 400, "height" : 400, "maxExtend" : true }, { "scheme" : null,
 "host" : { "replace" : "pictures.bgbm.org", "search" : "pictures.bgbm.org" }, "pathQueryFragment" : { "replac
e" : "digilib/Scaler/?fn=$1/$2&mo=crop&dw=200&dh=147&uvfix=1", "search" : "digilib/Scaler/?\\?fn=([^\\\\/]+)/(
\\w+)(.*)" }, "mimeType" : "image/jpeg", "width" : 200, "height" : 147, "maxExtend" : false } ]');
```


#49 - 01/18/2022 04:41 PM - Andreas Kohlbecker

- Status changed from Feedback to Closed

- Target version changed from Release 5.17 to Release 5.18

I think this can be closed now, the fix made comitted with 41ed639d also solves the problem that recently occurred in the cyprus portal.