EDIT - feature request #9146

filter image metadata by include and exclude lists of key words

07/13/2020 12:59 PM - Andreas Kohlbecker

Status:	Closed	**Start date:**	
Priority:	Highest	**Due date:**	
Assignee:	Andreas Kohlbecker	**% Done:**	100%
Category:	cdmlib	**Estimated time:**	0:00 hour
Target version:	Release 5.17		
Severity:	normal		

Description

Not all image metadata read from image files (stored in EXIF and ICPT data fields) it to be exposed to the public via web services and the dataporal.

Therefore filtering at the service level has to be applied.

TODO:

- CdmProperties for include and exclude lists of key names (EXIF and IPCT data are stored as key-value-pairs)
- order of application: 1. include, 2. exlcude
- provide a default for the include keywords

Related issues:

Related to EDIT - feature request #9152: PreferencesService caches CdmPrefere...	**Closed**
Related to EDIT - feature request #9153: Handle additional information in ima...	**New**
Related to EDIT - feature request #9200: extend application possibilities of ...	**New**

Associated revisions

Revision 036b973d - 07/13/2020 03:32 PM - Andreas Kohlbecker

ref #9146 media metadata service method with cdm property based include and exclude filter definitions

Revision 1c0438e9 - 07/13/2020 03:34 PM - Andreas Kohlbecker

ref #9146 typo

Revision 2eb4c6b4 - 07/13/2020 06:11 PM - Andreas Kohlbecker

ref #9146 media metadata service method with cdm property based include and exclude filter definitions - test

Revision bfc8c153 - 08/19/2020 03:25 PM - Andreas Kohlbecker

ref #9146 applying metadata filtering to all relevant controller methods

History

#1 - 07/13/2020 12:59 PM - Andreas Kohlbecker

- Target version changed from Release 5.15 to Release 5.18

#2 - 07/13/2020 01:00 PM - Andreas Kohlbecker

- Subject changed from filter image metadata by include and exclude lists to filter image metadata by include and exclude lists of key words

#3 - 07/13/2020 06:12 PM - Andreas Kohlbecker

- Status changed from New to Resolved

- Assignee changed from Andreas Kohlbecker to Andreas Müller

- % Done changed from 0 to 50

implemented with test

#4 - 07/14/2020 05:52 PM - Andreas Müller

- Status changed from Resolved to Feedback

- Assignee changed from Andreas Müller to Andreas Kohlbecker

Code looks fine and tests run so I guess it works, though no tests exist for "excludes".

I only have 2 architectural remark.

1. Wouldnt it be better to have the includes and excludes list within the signature of the service? So it would be

```
public Map<String, String> readResourceMetadataFiltered(MediaRepresentation representation, List<String> includes, List<String> excludes)
```

If includes or excludes is null the preferences values are taken. That makes the usage of the service method more flexibel.

1. For performance reasons it might be a good idea to move the main code into MediaRepresentation class and only link to there from the service class. The reason for this is: This method can also be called from outside cdmlib. E.g. from TaxEditor or Vaadin. These applications often cache Preference data. So if they have the Preference data already and if they also have the MediaRepresentation they may not need to send an expensive webservice call and/or start an expensive transaction. Instead they can directly call the method on MediaRepresentation.

#5 - 07/14/2020 06:01 PM - Andreas Müller

I don't know if this is handled in this ticket or in another one. But the current implementation does not yet handle metadata stored in the CDM. So if metadata like photographer, copyright, date, title, etc. is (only) stored in CDM data this is not handled by the code.
And ofcourse, as CDM data is still completely missing, also no preference rules are handled here. Rules like preferCdmOverIPTC_EXIF or something like this.
Such data is usually stored in the Media belonging to the MediaRrepresentation.

If there is another ticket for this please link.

#6 - 07/15/2020 02:33 PM - Andreas Kohlbecker

- Related to feature request #9152: PreferencesService caches CdmPreferences added

#7 - 07/15/2020 03:02 PM - Andreas Müller

- Related to feature request #9153: Handle additional information in image metadata added

#8 - 07/15/2020 03:03 PM - Andreas Müller

Please also have a look at #9153 when defining the default includes list

#9 - 08/17/2020 07:28 PM - Andreas Kohlbecker

- Assignee changed from Andreas Kohlbecker to Andreas Müller

Andreas Müller wrote:

> 1. Wouldn't it be better to have the includes and excludes list within the signature of the service? So it would be If includes or excludes is null the preferences values are taken. That makes the usage of the service method more flexibel.

We maybe could have both, the method as it is by now and the one witch takes the includes and excludes as parameter.
From the web service perspective i makes sense to let the the service method deal with the preferences internally so that you don't need two additional read-only transactions (could be reduced to one additional of course)

> 1. For performance reasons it might be a good idea to move the main code into MediaRepresentation class and only link to there from the service class. The reason for this is: This method can also be called from outside cdmlib. E.g. from TaxEditor or Vaadin. These applications often cache Preference data. So if they have the Preference data already and if they also have the MediaRepresentation they may not need to send an expensive webservice call and/or start an expensive transaction. Instead they can directly call the method on MediaRepresentation.

The filter method causes the image metadata to be read from the image resource URI. Putting this into model would the wrong place, from my understanding this is something which clearly should be placed into the service layer.

> I don't know if this is handled in this ticket or in another one. But the current implementation does not yet handle metadata stored in the CDM. So if metadata like photographer, copyright, date, title, etc. is (only) stored in CDM data this is not handled by the code.

Hmm, I don't think that this is needed at all. I assume that all data put into the CDM based media metadata is to be exposed via web service and portal. Anything elese would just be like a publish flag for metadata fields by the means of a filter.

> And ofcourse, as CDM data is still completely missing, also no preference rules are handled here. Rules like preferCdmOverIPTC_EXIF or something like this. Such data is usually stored in the Media belonging to the MediaRrepresentation.

This method was only meant for filtering the image resource metadata. I would not start mixing these things here. For clear separation of concern It makes more sense to combine the cdm media metadata and the media file metadata in a separate step. This way the different methods are better testable.

I consider this ticked as fully completed. Pease open a new ticket if you think that we need a second variant of this method which exposes the includes and excludes via the method parameters.

#10 - 08/19/2020 03:02 PM - Andreas Müller

Andreas Kohlbecker wrote:

> Andreas Müller wrote:
>
> > 1. Wouldn't it be better to have the includes and excludes list within the signature of the service? So it would be If includes or excludes is null the preferences values are taken. That makes the usage of the service method more flexibel.
>
> We maybe could have both, the method as it is by now and the one witch takes the includes and excludes as parameter.
> From the web service perspective i makes sense to let the the service method deal with the preferences internally so that you don't need two additional read-only transactions (could be reduced to one additional of course)

I agree that both may make sense, however, the method including the includes and excludes list seems not to be implemented yet.
Note: we could also keep only 1 method and define that a null value means that a maybe existing preference is taken. This allows setting only 1 value explicitly but taking the other value from preferences (don't know if this is a usecase but just in case)

#11 - 08/19/2020 03:15 PM - Andreas Müller

Andreas Kohlbecker wrote:

> > 1. For performance reasons it might be a good idea to move the main code into MediaRepresentation class and only link to there from the service class. The reason for this is: This method can also be called from outside cdmlib. E.g. from TaxEditor or Vaadin. These applications often cache Preference data. So if they have the Preference data already and if they also have the MediaRepresentation they may not need to send an expensive webservice call and/or start an expensive transaction. Instead they can directly call the method on MediaRepresentation.
>
> The filter method causes the image metadata to be read from the image resource URI. Putting this into model would the wrong place, from my understanding this is something which clearly should be placed into the service layer.

I see your point that the real model classes themself should not handle external services. However, in cdmlib-commons we do have methods like CdmImageInfo.readMetaData() which do similar things. The performance issue is still valid I think. And also the method is not dependent on the repository/DB and therefore does not need to be handled on server side (this may also improve server performance). Maybe a solution is to have a utility class like CdmImageInfo in model which handles this. Or maybe we could even have it in cdmlib-ext (which might be available in TaxEditor even if cdmlib-service is not available). Somehow this fits somehow as the data really comes from an external server (image server).

#12 - 08/19/2020 03:27 PM - Andreas Müller

> > I don't know if this is handled in this ticket or in another one. But the current implementation does not yet handle metadata stored in the CDM. So if metadata like photographer, copyright, date, title, etc. is (only) stored in CDM data this is not handled by the code.
> > Hmm, I don't think that this is needed at all. I assume that all data put into the CDM based media metadata is to be exposed via web service and portal. Anything elese would just be like a publish flag for metadata fields by the means of a filter.
> > And ofcourse, as CDM data is still completely missing, also no preference rules are handled here. Rules like preferCdmOverIPTC_EXIF or something like this. Such data is usually stored in the Media belonging to the MediaRrepresentation.
> > This method was only meant for filtering the image resource metadata. I would not start mixing these things here. For clear separation of concern It makes more sense to combine the cdm media metadata and the media file metadata in a separate step. This way the different methods are better testable.

I agree that it might be a good idea to combine the CDM base meta data in a separate step. Also, because the problem here is that we also need something like a matching algorithm. The data in CDM is often not stored under the same name there a pure string filtering/matching does not work. We also need a mapping from 1 label string to the other.
By the way, this is also something that is true for IPTC and EXIF which have often slightly different labels for soemthing that has the same meaning. Mapping these and defining priority rules might be a new ticket.

So if CDM based metadata is definitely showing up in the "additional information" this part of the ticket/comments is solved. The possibility to also filter CDM based data is a nice to have but not a must.

#13 - 08/19/2020 03:27 PM - Andreas Müller

- Assignee changed from Andreas Müller to Andreas Kohlbecker

#14 - 08/19/2020 04:53 PM - Andreas Kohlbecker

- Related to feature request #9200: extend application possibilities of MediaService.readResourceMetadataFiltered() added

#15 - 08/19/2020 08:42 PM - Andreas Müller

- Target version changed from Release 5.18 to Release 5.17

#16 - 08/19/2020 11:14 PM - Andreas Kohlbecker

- Status changed from Feedback to Closed

- % Done changed from 50 to 100