GSoC Week 4 Queries

1. I have created a method that will call the /models method of the AI Model service to pull models information. I also have created the additional settings for Chest X rays (Chest Model task, Chest Model endpoint, Chest Model version, Chest Model Thresholds). I want to understand how to convert the chest model task in the administrative settings to a dropdown and fill that drop down using the output from the created method.

2. How to access DICOM tags related information. I will use them for implementing the hook for model selection by modality. **Update 06/21/2023 -** I was able to figure this out

3. I am working on hook implementation for model selection and faced challenges. I have come up with a solution to implement. I want to address this before proceeding.

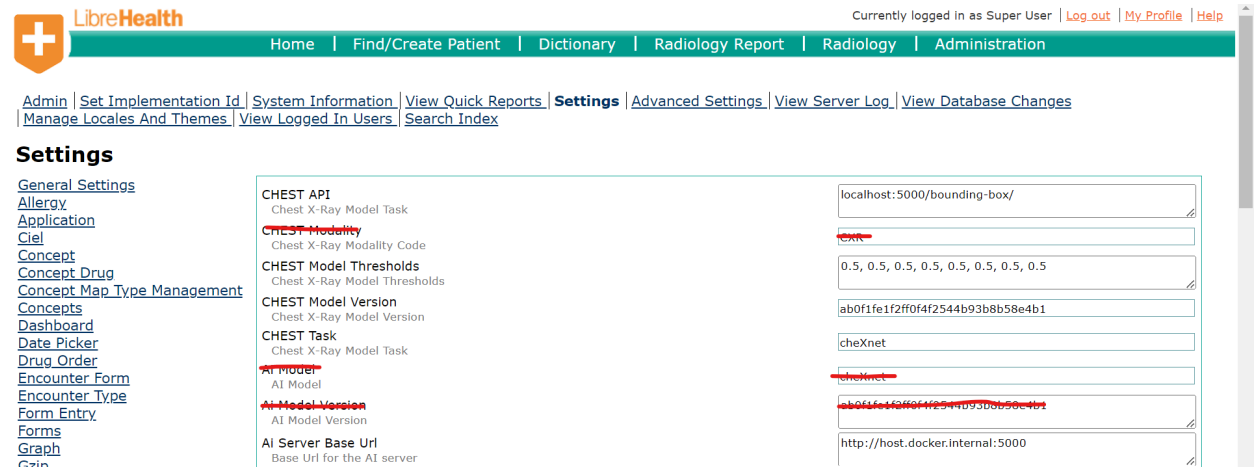
For example, For the newly created administration settings if I enter values for,

Chest api

Chest model thresholds

Chest model version

Chest task



This will become as global properties,

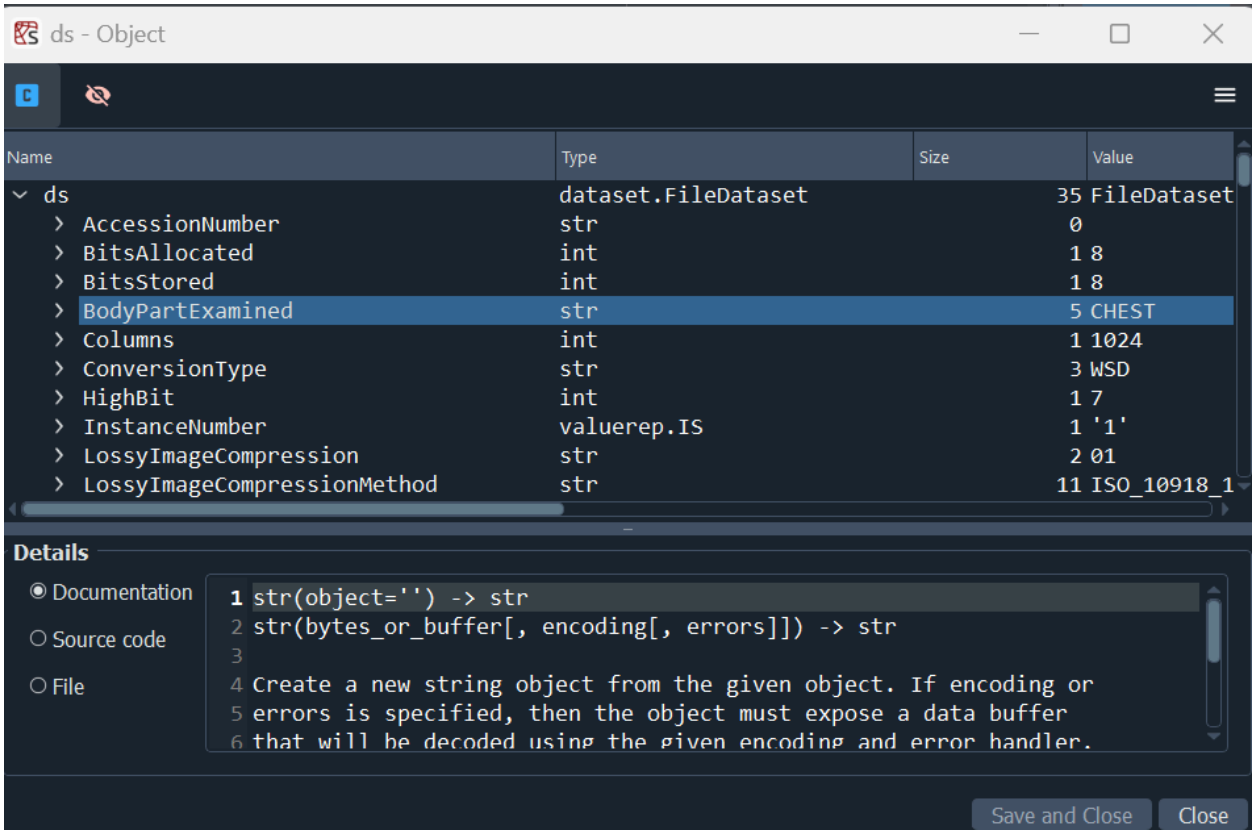
Chest api – radiology.chest.Api

Chest model threshold – radiology.chest.ModelThresholds

Chest model version – radiology. chest.ModelVersion

Chest task – radiology.chest.Task

From the DICOM metadata I can read the “BodyPartExamined” attribute and it will be “CHEST”



While calling the model using the [getAiDataFromAiServer] I can replace this code,

```
String aiServerUrl =
administrationService.getGlobalProperty("radiology.aiServerBaseUrl") +
"/bounding-box";
String aiModel = administrationService.getGlobalProperty("radiology.aiModel");
String aiModelVersion =
administrationService.getGlobalProperty("radiology.aiModelVersion");
```

With

```
String aiServerUrl =
administrationService.getGlobalProperty("radiology.aiServerBaseUrl") +
administrationService.getGlobalProperty("radiology.chest.Api")
String aiModel =
administrationService.getGlobalProperty("radiology.chest.Task");
String aiModelVersion =
administrationService.getGlobalProperty("radiology.chest.ModelVersion");
```

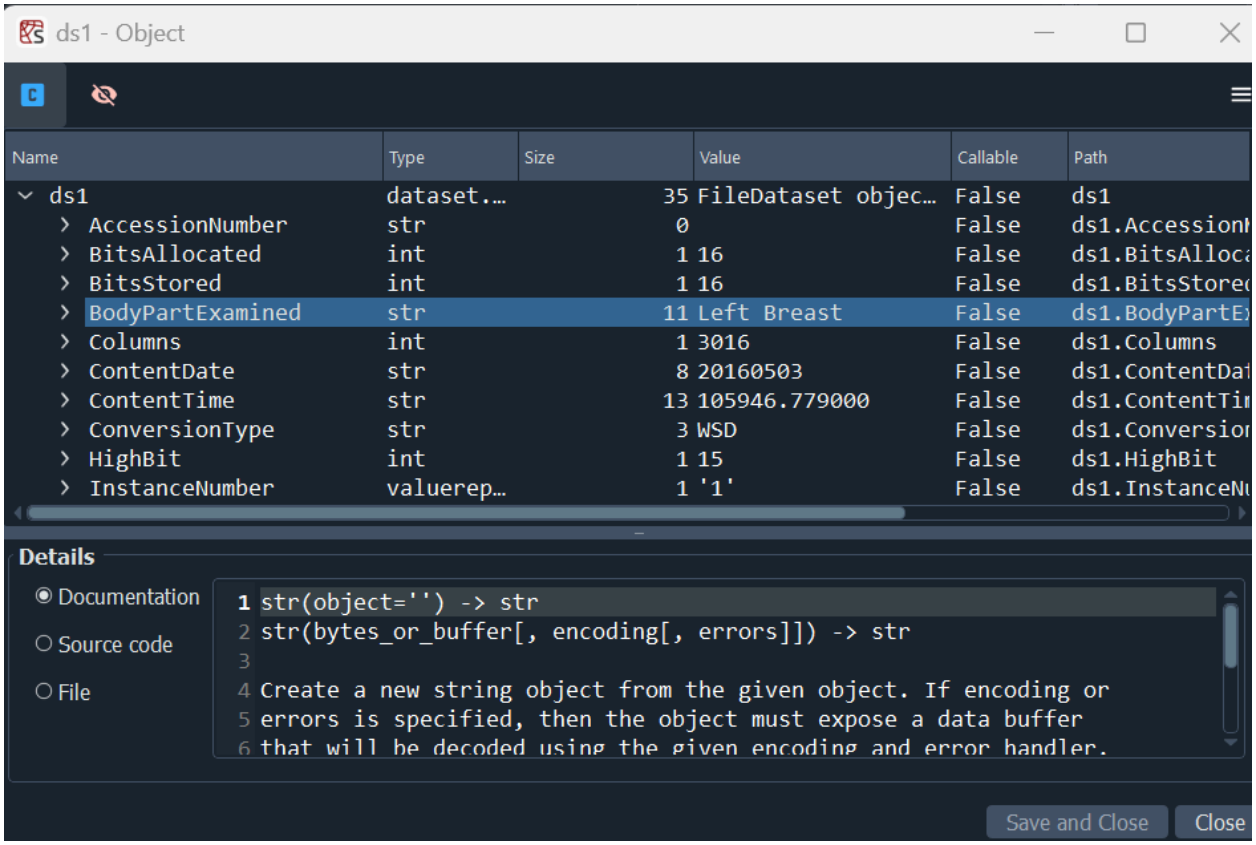
Where I replace “/bounding-box” to radiology.chest.Api, radiology.aiModel to radiology.chest.Task, and so on

Now how do I integrate DICOM metadata to do autoselection? I am not able to find any global properties for it or how to read it.

If I have global properties for dicom I can do something like,

```
String aiServerUrl =
administrationService.getGlobalProperty("radiology.aiServerBaseUrl") +
administrationService.getGlobalProperty("radiology." +
dicom.getGlobalProperty("dicom.BodyPartExamined") + ".Api")
String aiModel = administrationService.getGlobalProperty("radiology." +
dicom.getGlobalProperty("dicom.BodyPartExamined") + ".Task");
String aiModelVersion = administrationService.getGlobalProperty("radiology." +
dicom.getGlobalProperty("dicom.BodyPartExamined") + ".ModelVersion");
```

But another issue is if we look at Mammograms the BodyPartExamined metadata of dicom can be “left breast”, “right breast”. This might not synergize with the method we used.



The next thing we can do is we can add an administrative property which looks at certain information in the DICOM metadata and calls the model. **(Update 06/21/2022 - Able to identify how to pull the image dicom metadata information)**

For example,

Chest dicom property look up – radiology.chest.DicomPropertyLookup

Can have values as a json,

{“modality”:”CR”, “BodyPartExamined”:”CHEST”}

Similarly for Mammography we can have, radiology.mammo.DicomPropertyLookup

{“modality”:”MG”, “BodyPartExamined” : [“left breast”, “right breast”]}

Here the list in the json can be an OR condition

In the getAiDataFromAiServer, we can use the AdministrationService Class we use the method

```
getGlobalPropertiesBySuffix
```

and pull the global properties that align with “DicomPropertyLookup”.

We iterate on each property’s values and match to the respective dicom property.

For example we first take all global properties that have global property suffix - “DicomPropertyLookup”

We get,

radiology.chest.DicomPropertyLookup - {“modality”:”CR”, “BodyPartExamined”:”CHEST”}

radiology.mammo.DicomPropertyLookup - {“modality”:”MG”, “BodyPartExamined” : [“left breast”, “right breast”]}

Now we iterate on their values with the DICOM image metadata,

Lets say it matches with,

radiology.chest.DicomPropertyLookup - {“modality”:”CR”, “BodyPartExamined”:”CHEST”}

we take the “chest” value from the global property string,

```
// Assuming globalProperty = radiology.chest.DicomPropertyLookup

String[] globalPropertyParts = globalProperty.split("\\.");

String globalPropertyModality = globalPropertyParts[1];

String aiServerUrl =
administrationService.getGlobalProperty("radiology.aiServerBaseUrl") +
administrationService.getGlobalProperty("radiology." +  globalPropertyModality
+ ".Api")
String aiModel = administrationService.getGlobalProperty("radiology." +
globalPropertyModality + ".Task");
String aiModelVersion = administrationService.getGlobalProperty("radiology." +
globalPropertyModality + ".ModelVersion");
```

This is one way to implement the hook. Can you advise if this is feasible?

Update 06/21/2023

I looked at the first recording again.

From the discussion I understand that only the Model Task and Model Endpoint will need to be integrated into the Administration Settings (Not include Modality).

Example,

Model Task - cheXnet, Mammo

Model Endpoint - /bounding-box, /other-box

The DICOM metadata will be read and sent to the AI Model Service along with the Model Tasks and Model Endpoints.

So in the AI Model service do we use a common endpoint that reads the DICOM metadata and based on it we pick the appropriate Model Task and Model Endpoint and redirect in the Flask app itself?

The other issue is there is no common attribute to differentiate a chest from a mammo dicom using the dicom metadata

for example for chest we may need to look at the “BodyPartExamined” attribute which has value “CHEST” in the dicom metadata BUT for mammo we look at the “Modality” which has value “MG”.

So how do we integrate this without hardcoding?