

AIM API User Guide

by Hakan BULU
hbulu@stanford.edu

(Last modified at December 27, 2011)

1. Prerequisites

1.1. For general usage

1.1.1. AIMv3API.jar File

1.2. To be able to use AIM Annotation Database Server

1.2.1. org.restlet.jar File

1.2.2. If you want to access Stanford Berkeley Database, you need to be in Stanford network or you need to connect to the network with your SuNet ID by using VPN.

2. Example Usage

2.1. Creating sample Image Annotation instance

```
ImageAnnotation imageAnnotation = new ImageAnnotation(0, "AIM.1.0", "", "2008-04-11T15:32:15", "0022",  
"1.2.288.3.2205383238.1512.1207945935.1", "112041", "Target Lesion Complete Response", "DCM", null, null);
```

```
// Person Start
```

```
Person person = new Person();  
person.setBirthDate("2000-01-01T00:00:00");  
person.setCagridId(0);  
person.setName("1.3.6.1.4.1.9328.50.1.0022");  
person.setId("409978");  
person.setSex("M");  
imageAnnotation.addPerson(person);
```

```
// Person End
```

```
// ImageReference start
```

```
DICOMImageReference dicomImageReference = new DICOMImageReference();  
dicomImageReference.setCagridId(0);
```

```
// ImageStudy start
```

```
ImageStudy imageStudy = new ImageStudy();  
imageStudy.setCagridId(0);
```

```

imageStudy.setInstanceUID("1.3.6.1.4.1.9328.50.1.11470");
imageStudy.setStartDate("2000-01-01T00:00:00");
imageStudy.setStartTime("+00:00:00.000000");
// ImageSeries start
ImageSeries imageSeries = new ImageSeries();
imageSeries.setCagridId(0);
imageSeries.setInstanceUID("1.3.6.1.4.1.9328.50.1.11563");
// Image start
Image image = new Image();
image.setCagridId(0);
image.setSopClassUID("1.2.840.10008.5.1.4.1.1.2");
image.setSopInstanceUID("1.3.6.1.4.1.9328.50.1.11623");
// Image into ImageSeries
imageSeries.addImage(image);
// ImageSeries into ImageStudy
imageStudy.setImageSeries(imageSeries);
// ImageStudy into ImageReference
dicomImageReference.setImageStudy(imageStudy);
// ImageReference into ImageAnnotation
imageAnnotation.addImageReference(dicomImageReference);
// ImageReference end

// GeometricShape start
Polyline polyline = new Polyline(0, "", "", "", "", false, -1);
TwoDimensionSpatialCoordinate spatialCoordinate_1 = new TwoDimensionSpatialCoordinate(0, 0,
"1.3.6.1.4.1.9328.50.1.10717", 0, 143.0, 300.0);
TwoDimensionSpatialCoordinate spatialCoordinate_2 = new TwoDimensionSpatialCoordinate(0, 1,
"1.3.6.1.4.1.9328.50.1.10717", 0, 92.0, 326.0);
polyline.addSpatialCoordinate(spatialCoordinate_1);
polyline.addSpatialCoordinate(spatialCoordinate_2);
imageAnnotation.addGeometricShape(polyline);
// GeometricShape end

// Equipment start
Equipment equipment = new Equipment(0, "GE MEDICAL SYSTEMS", "", "");
imageAnnotation.addEquipment(equipment);
// Equipment end

// User start
User user = new User(0, "A", "NWU", "Referring", 3);
imageAnnotation.addUser(user);
// User end

// ImagingObservation start
ImagingObservation imagingObservation = new ImagingObservation(0, "REX4010", "Calcification", "RADREX",
null, "", null, null, "");

```

```

ImagingObservationCharacteristic imagingObservationCharacteristic = new ImagingObservationCharacteristic(0,
"REX4020", "LIDC Calcification 1", "RADREX", null, "", null, "");
imagingObservation.addImagingObservationCharacteristic(imagingObservationCharacteristic);
imageAnnotation.addImagingObservation(imagingObservation);
// ImagingObservation end

// Create a Calculation instance
Calculation calculation = new Calculation();
calculation.setCagridId(0);
calculation.setAlgorithmVersion("algorithmVersion");
calculation.setUid("0");
calculation.setDescription("description");
calculation.setCodeValue("codeValue");
calculation.setCodeMeaning("codeMeaning");
calculation.setCodingSchemeDesignator("codingSchemeDesignator");
// Create a CalculationResult instance
CalculationResult calculationResult = new CalculationResult();
calculationResult.setCagridId(0);
calculationResult.setType(CalculationResultIdentifier.Scalar);
calculationResult.setUnitOfMeasure("ratio");
calculationResult.setNumberOfDimensions(0);
// Create a CalculationData instance
CalculationData calculationData = new CalculationData();
calculationData.setCagridId(0);
calculationData.setValue(2.3);
calculationData.addCoordinate(0, 0, 0);
// Create a Dimension instance
Dimension dimension = new Dimension(0, 0, 0, "Edge Sharpness Scale Histogram Bin 1");
// Add calculationData to calculationResult
calculationResult.addCalculationData(calculationData);
// Add dimension to calculationResult
calculationResult.addDimension(dimension);
// Add calculationResult to calculation
calculation.addCalculationResult(calculationResult);
// Add calculation to imageAnnotation
imageAnnotation.addCalculation(calculation);

```

2.2. Saving Image Annotation instance to File

```

// Always I check the ImageAnnotation if it valid based on the AIM XML Schema
AnnotationBuilder.saveToFile(imageAnnotation, "./test.xml", pathXSD);

```

2.3. Saving Image Annotation instance to Server

```
// Required Parameters to be able to access BerkeleyDB
String namespace = "gme://caCORE.caCORE/3.2/edu.northwestern.radiology.AIM";
String serverUrlUpload = "http://rufus.stanford.edu:8100/annotations/upload";
String serverUrlDownload = "http://rufus.stanford.edu:8100/annotations/xquery";

// SAVING ANY IMAGE ANNOTATION TO THE XML DATABASE
// I choose 'test_db' as my collection name. It can be changed.
// Before send my ImageAnnotation to the server, I check if it is already exist.
// Because we don't want to insert more than one ImageAnnotation which has same UniqueIdentifier value
if (!AnnotationGetter.isExistInTheServer(serverUrlDownload, namespace, "test_db", imageAnnotation
.getUniqueIdentifier())) {
    // I'm sending my ImageAnnotation to the server.
    String res = AnnotationBuilder.saveToServer(imageAnnotation, serverUrlUpload, serverUrlDownload,
namespace, "test_db", pathXSD);
    // Displaying Server's response.
    System.out.println(res);
}
```

2.4. Get an Image Annotation from File

```
// Using AnnotationGetter static class
ImageAnnotation imageAnnotation = AnnotationGetter.getImageAnnotationFromFile("C:/AIM_Sample.xml",
"C:/AIM_v3.xsd");
```

2.5. Get an Image Annotation from Server by its UniqueIdentifier

```
// Required Parameters to be able to access BerkeleyDB
String namespace = "gme://caCORE.caCORE/3.2/edu.northwestern.radiology.AIM";
String serverUrlDownload = "http://rufus.stanford.edu:8100/annotations/xquery";

// Using AnnotationGetter static class
ImageAnnotation imageAnnotation =
AnnotationGetter.getImageAnnotationFromServerByUniqueIdentifier(serverUrlDownload, namespace, "test_db",
"1.2.288.3.2205383238.1512.1207945935.5", "C:/AIM_v3.xsd");
```

2.6. Example: Extracting the Type of the ROI and the Coordinates

```
ImageAnnotation imageAnnotation = AnnotationGetter.getImageAnnotationFromFile("C:/AIM_Sample.xml",
"C:/AIM_v3.xsd");
GeometricShapeCollection geometricShapeCollection = imageAnnotation.getGeometricShapeCollection();
for (int i = 0; i < geometricShapeCollection.getGeometricShapeList().size(); i++) {
    GeometricShape geometricShape = geometricShapeCollection.getGeometricShapeList().get(i);
    System.out.println(geometricShape.getXsiType());
    for (int j = 0; j < geometricShape.getSpatialCoordinateCollection().getSpatialCoordinateList().size();
j++) {
        SpatialCoordinate spatialCoordinate =
geometricShape.getSpatialCoordinateCollection().getSpatialCoordinateList().get(i);
        if ("TwoDimensionSpatialCoordinate".equals(spatialCoordinate.getXsiType())) {
            TwoDimensionSpatialCoordinate twoDimensionSpatialCoordinate = (TwoDimensionSpatialCoordinate)
spatialCoordinate;
            System.out.println(twoDimensionSpatialCoordinate.getCoordinateIndex());
            System.out.println(twoDimensionSpatialCoordinate.getX());
            System.out.println(twoDimensionSpatialCoordinate.getY());
        }
    }
}
```

2.7. List of Functions for Querying Image Annotation Instances from the Server

- `ImageAnnotation AnnotationGetter.getImageAnnotationFromServerByUniqueIdentifier(serverURL, namespace, collection, uniqueIdentifier, PathXSD)`
- `List<ImageAnnotation> AnnotationGetter.getImageAnnotationsFromServerByPersonNameEqual(serverURL, namespace, collection, PersonName, PathXSD)`
- `List<ImageAnnotation> AnnotationGetter.getImageAnnotationsFromServerByPersonNameContains(serverURL, namespace, collection, PersonName, PathXSD)`
- `List<ImageAnnotation> AnnotationGetter.getImageAnnotationsFromServerByDateTimeEqual(serverURL, namespace, collection, dateTime, PathXSD)`
- `List<ImageAnnotation> AnnotationGetter.getImageAnnotationsFromServerByDateTimeContains(serverURL, namespace, collection, dateTime, PathXSD)`
- `List<ImageAnnotation> AnnotationGetter.getImageAnnotationsFromServerByCagridIdEqual(serverURL, namespace, collection, cagridId, PathXSD)`
- `List<ImageAnnotation> AnnotationGetter.getImageAnnotationsFromServerByCagridIdContains(serverURL, namespace, collection, cagridId, PathXSD)`
- `List<ImageAnnotation> AnnotationGetter.getImageAnnotationsFromServerByNameEqual(serverURL, namespace, collection, name, PathXSD)`

- `List<ImageAnnotation> AnnotationGetter.getImageAnnotationsFromServerByNameContains(serverURL, namespace, collection, name, PathXSD)`
- `List<ImageAnnotation> AnnotationGetter.getImageAnnotationsFromServerByCodeMeaningEqual(serverURL, namespace, collection, codeMeaning, PathXSD)`
- `List<ImageAnnotation> AnnotationGetter.getImageAnnotationsFromServerByCodeMeaningContains(serverURL, namespace, collection, codeMeaning, PathXSD)`
- `List<ImageAnnotation> AnnotationGetter.getImageAnnotationsFromServerByCodeValueEqual(serverURL, namespace, collection, codeValue, PathXSD)`
- `List<ImageAnnotation> AnnotationGetter.getImageAnnotationsFromServerByCodeValueContains(serverURL, namespace, collection, codeValue, PathXSD)`
- `List<ImageAnnotation> AnnotationGetter.getImageAnnotationsFromServerByCodingSchemeDesignatorEqual(serverURL, namespace, collection, codingSchemeDesignator, PathXSD)`
- `List<ImageAnnotation> AnnotationGetter.getImageAnnotationsFromServerByCodingSchemeDesignatorContains(serverURL, namespace, collection, codingSchemeDesignator, PathXSD)`

2.8. Example: Getting all Image Annotation Instances by Person's(Patient's) name, and display some properties of each Image Annotation instance with its Person instance

```

List<ImageAnnotation> listAnno =
AnnotationGetter.getImageAnnotationsFromServerByPersonNameContains(serverUrlDownload, namespace, "test_db",
"1.3.6.1.4.1.9328.50.1.00", "C:/AIM_v3.xsd");
for (int i = 0; i < listAnno.size(); i++) {
    ImageAnnotation anno = listAnno.get(i);
    System.out.println("Image Annotation UniqueIdentifier: " + anno.getUniqueIdentifier());
    System.out.println("Image Annotation CodeMeaning: " + anno.getCodeMeaning());
    System.out.println("Image Annotation DateTime: " + anno.getDateTime());
    List<Person> listPerson = anno.getListPerson();
    for (int j = 0; j < listPerson.size(); j++) {
        Person person = listPerson.get(j);
        System.out.println("Person Name: " + person.getName());
        System.out.println("Person BirthDate: " + person.getBirthDate());
        System.out.println("Person Sex: " + person.getSex());
    }
}

```

2.9. Example: Reading older AIM Annotation file

```
ImageAnnotation anno = AnnotationGetter.getImageAnnotationFromFile("C:\\\\olderAIM.xml");
System.out.println(anno.getUniqueIdentifier());
anno.setCodeMeaning("test meaning");
AnnotationBuilder.saveToFile(anno, "C:\\\\outAIM.xml", pathXSDforCurrentVersion);
```

3. Annotation Extender

3.1. Add Feature to Image Annotation

```
double[] featureValue = new double[]{1.1, 2.2, 3.3};
String[] featureString = new String[]{"EdgeSharpness 1", "EdgeSharpness 2", "EdgeSharpness 3"};
double featureVersion = 1.0;

imageAnnotation = AnnotationExtender.addFeature(imageAnnotation, featureValue, featureString,
featureVersion);
```

3.1.1. Output AIM XML File

```
<Calculation algorithmVersion="1.0" cagridId="0" codeMeaning="codeMeaning" codeValue="codeValue"
codingSchemeDesignator="codingSchemeDesignator" description="description" uid="0">
  <calculationResultCollection>
    <CalculationResult cagridId="0" numberOfDimensions="0" type="Scalar" unitOfMeasure="ratio">
      <calculationDataCollection>
        <CalculationData cagridId="0" value="1.1">
          <coordinateCollection>
            <Coordinate cagridId="0" dimensionIndex="0" position="0"/>
          </coordinateCollection>
        </CalculationData>
      </calculationDataCollection>
      <dimensionCollection>
        <Dimension cagridId="0" index="0" label="EdgeSharpness 1" size="1"/>
      </dimensionCollection>
    </CalculationResult>
    <CalculationResult cagridId="0" numberOfDimensions="0" type="Scalar" unitOfMeasure="ratio">
      <calculationDataCollection>
        <CalculationData cagridId="0" value="2.2">
          <coordinateCollection>
            <Coordinate cagridId="0" dimensionIndex="0" position="0"/>
          </coordinateCollection>
        </CalculationData>
      </calculationDataCollection>
      <dimensionCollection>
        <Dimension cagridId="0" index="0" label="EdgeSharpness 2" size="1"/>
      </dimensionCollection>
    </CalculationResult>
  </calculationResultCollection>
</Calculation>
```

```
</dimensionCollection>
</CalculationResult>
<CalculationResult cagridId="0" numberOfDimensions="0" type="Scalar" unitOfMeasure="ratio">
  <calculationDataCollection>
    <CalculationData cagridId="0" value="3.3">
      <coordinateCollection>
        <Coordinate cagridId="0" dimensionIndex="0" position="0"/>
      </coordinateCollection>
    </CalculationData>
  </calculationDataCollection>
  <dimensionCollection>
    <Dimension cagridId="0" index="0" label="EdgeSharpness 3" size="1"/>
  </dimensionCollection>
</CalculationResult>
</calculationResultCollection>
</Calculation>
```