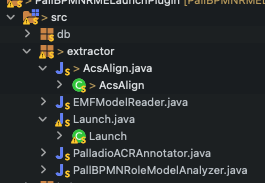
**ACSAlign Methoden Kontrollfluss**

Übersicht Projekt OrdnerStruktur mit den verwendeten Paketen und Klassen



## Kontrollfluss der einzelnen Methoden

extractor.Launch: launch()

* Wird gestartet beim Ausführen des Plugins
* Wenn im unteren Teil des Run As Menüs die Checkbox Activate AcsAlign aktiviert wird und zusätzlich eine dataTypeUsage.json ausgewählt wird, wird der Extraktionsprozess der AcsAlign Berechtigungen beim Ausführen des Plugins gestartet. In der Launch Funktion wird dann der folgende Programmcode ausgeführt:
  + Der Konstruktor mit den erforderlichen Dateipfaden zu den Modellen wird aufgerufen und danach die Extraktion mit startAcsAlign() gestartet.
  + *AcsAlign acsAlign = new AcsAlign(projectPath, dataPath, repoPath, selectedReadWriteDatatypesFilePath);*
  + AcsAlign.startAcsAlign();

extractor.AcsAlign: startAcsAlign()

* Extractor.AcsAlign: getAllDataTypesFromDataModel()
  1. Liest das Datenmodell (\*.datamodel) ein
  2. Extrahiert alle Datenobjekte
  3. Speichert die Datentypen der Datenobjekte jeweils mit Namen und Datentyp in einer HashMap<String, DataType>
  4. Gibt die HashMap<String, DataType> zurück
     1. Hashmap wird danach in dKnown gespeichert
* Extractor.AcsAlign: getAllowedDataTypeStringsFromDB()
  1. Stellt eine Verbindung zur Datenbank her
  2. Abfrage von allen erlaubten Datentypen aus der ACBPR Tabelle
  3. Liest jeweils die ELSC Id, den Prozess und eine Liste der IS Datentypen Ids aus.
  4. Speichert diese in einer HashMap <String, Set<String>> mit dem “ELSCId.Prozess” als Namen und einem HashSet mit den IS Datentypen Ids als Werte.
  5. Gibt die HashMap<String, Set<String>> (AlldAllowed) zurück die alle erlaubten Datentypen für den jeweiligen ELSC enthält
* Um die passenden erlaubten Datentypen aus der JSON Datei zu extrahieren muss nun über alle AlldAllowed iteriert werden. For each allDAllowed -> dAllowed, elscId
  + Extractor.AcsAlign: getReadWriteDataTypesFromJSON(ElscId)
    1. Liest die JSON Datei ein und stellt Werte als Klassen zur Verfügung
    2. Iteriert über alle Werte (aus dem JSON) die ELSC Ids enthalten
    3. Bei Treffern mit der übergebenen ELSC Id werden jeweils die Lese und Schreibberechtigungen in einer HashMap<String, Set<DataType>> mit READ oder WRITE als String und den Datentypen als Set gespeichert.
       - Beispieleintrag: <"READ", Set<TypeA, TypeB,...>
    4. Gibt die HashMap <String,Set<DataType>> combinedReadAndWriteDataTypes zurück

* + Für jeden Eintrag in combinedReadAndWriteDatatTypes wird nun der AcsAlign Algorithmus ausgeführt: For each combinedReadAndWriteDatatTypes
    - Extractor.AcsAlign: acsAlignAlgorithm(readWriteDataType, dKnown, dAllowed, dataTypekey, elscId)
      * Überprüft Datentype und castet zu Composite,Collection oder Primitive
      * Setzt allowed
      * Wenn allowed und composite
        + Iteriert über innere Datentypen und ruft Algorithmus rekursiv auf
        + allowed = acsAlignAlgorithm(readWriteDataType, dKnown, dAllowed, dataTypekey, elscId)
      * Wenn allowed und collection nicht in dKnown
        + Setzt allowed true wenn innerer Collection type nicht in dKnown
      * Gibt den boolean zurück ob die Berechtigung erlaubt ist = allowed
  + Nach jeder Ausführung des Algorithmus wird das Ergebnis in die Datenbank geschrieben:
    - db:insertACRBreachDataToTable(elscId, dataTypekey + " " + readWriteName, allowed);
      * Schreibt daten in die Tabelle ACRBreach