DrupalMigrationActivity::DrupalImportActivityDiagram

Main flow
- start
- 1. Collection import
- 2. Publication import
- 3. File import
- finish

General initialization
- DtFileImportGeneralMigration::__construct()
  - Run parent contructor → XMLMigration::__construct()
  - Get XML file location
  - Load XML file to memory
  - Check file_import folder
    - not exists → Create file_import folder
    - exists

getLanguages()
- validateImportType()
- Get the xpath of the type and language
- Add languages, which found by xpath
- return with an array of languages

getTranslationElementByIdAndLanguage()
- validateImportType()
- Get the element by using xpath
  - [count>0] return elements
  - [count=0] return FALSE

getPrimaryLanguage()
- getLanguages()
  - if found: 'en' then return 'en' otherwise return with the first language
  - no language → Migration message: 'No language'
  - return: FALSE

getTitleById()
- getPrimaryLanguage()
  - found → getTranslationElementByIdAndLanguage()
    - no title → return: title
  - not found → Migration message: 'No title'
  - return: FALSE

1. Collection import
- DtFileImportCollectionImport::__contruct()
  - General initialization
  - Collection import field mapping
- Iteration of collections
  - DtFileImportCollectionImport::prepareRow() → getTitleById()
  - DtFileImportCollectionImport::prepare() → getPrimaryLanguage()
  - Migration process
  - DtFileImportPublicationImport::complete()
    - Get languages of the collection → getLanguages(), getPrimaryLanguage()
    - Iterate on languages
      - getTranslationElementByIdAndLanguage()
      - title valid → Set special field values → Set translation
      - title not valid, skip
    - node_save()
- end

2. Publication import
- DtFileImportPublicationImport::__contruct()
  - General initialization
  - Publication import field mapping
- Iteration of publications
  - DtFileImportPublicationImport::prepareRow() → getTitleById()
  - DtFileImportPublicationImport::prepare() → getPrimaryLanguage()
  - Migration process
  - DtFileImportPublicationImport::complete()
    - Get languages of the publication → getLanguages(), getPrimaryLanguage()
    - Iterate on languages
      - getTranslationElementByIdAndLanguage()
      - title valid → Set special field values → Set translation
      - title not valid, skip
    - node_save()
    - Update related collections
- end

3. File import
- DtFileImportFileImport::__contruct()
  - General initialization
  - File import field mapping
- Iteration of files
  - DtFileImportFileImport::prepareRow() → getTitleById()
  - DtFileImportFileImport::prepare → getPrimaryLanguage()
  - Migration process
  - DtFileImportFileImport::complete()
    - Get languages of the file → getLanguages(), getPrimaryLanguage()
    - Iterate on languages
      - getTranslationElementByIdAndLanguage()
      - title valid → Set special field values
        - What is the file type?
          - Link to a file → Set legacy link
          - Upload the file → getFile()
        - Set translation
      - title not valid, skip
    - Isset field_core_file or field_core_legacy_link?
      - yes → node_save() → Update related publications
      - no → node_delete()
- end

getFile()
- Get file url from XML
- Set filename
- getFileContentByCurl()
- Was successful?
  - no
  - yes → Save managed file
    - Saving was successful?
      - yes: return file_object
      - no: retrun NULL