OLE Data Load at Startup Technical Document

Table of Contents

Last Published May 7, 2015

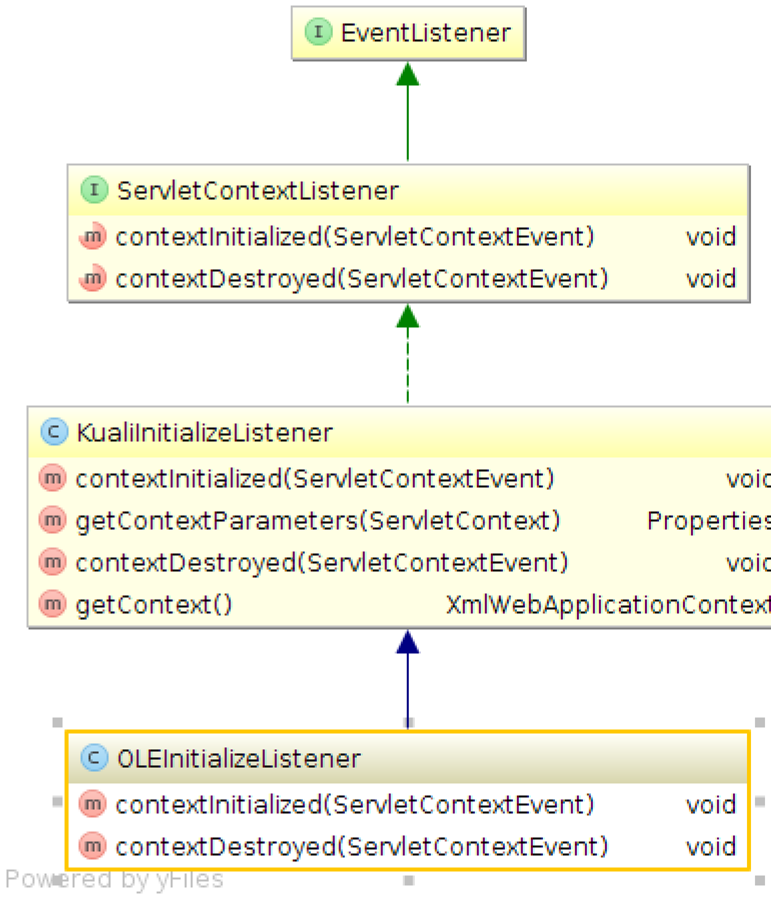
Purpose

OLE allows alternate ways to load data into the system. This document describes about the process that loads the data during server startup so that the user gets to start using the data as soon as the application is live and not wait for it to be loaded as a separate process. Patrons, Circulation Policies, Licenses, eResources and Ingest Profile can be loaded through this method.

Dependencies / Logical Data Model / Physical Data Model

The process is an alternate way to load data into the system and doesn't make use of any special tables. The data is held in tables designated for the respective services.

Service Interface Design (Java)

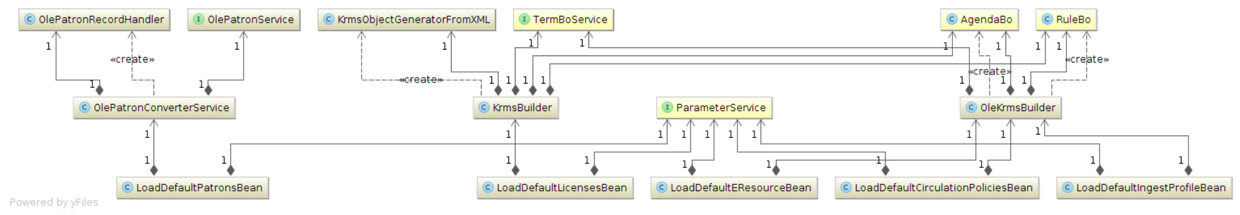


EventListener
ServletContextListener
contextInitialized(ServletContextEvent) void
contextDestroyed(ServletContextEvent) void
KualiInitializeListener
contextInitialized(ServletContextEvent) void
getContextParameters(ServletContext) Properties
contextDestroyed(ServletContextEvent) void
getContext() XmlWebApplicationContext
OLEInitializeListener
contextInitialized(ServletContextEvent) void
contextDestroyed(ServletContextEvent) void
Powered by yFiles

Last Published May 7, 2015

The *OLEInitializeListener* class is where the loading of the Patrons, Circulation policy, Licenses, eResources and Ingest profile happens. It extends *KualiInitializeListener* class which in turn implements the *ServletContextListener* class of Java Servlet.

The configuration as to whether auto ingest should happen or not is defined in the *olefs-config-defaults.xml* file under the parameter, *autoIngestDefaults.* If set to true, the services are injected during initialization. The bean definitions are mentioned in the *KRMSLocalSpringBeans.xml* file. Under each bean definition, *fileName* is also mentioned as a property.



Patron

In case of Patron, the class *LoadDefaultPatronsBean* class is where the process is initiated. Based on the System Parameter and the arguments passed in the method, the persistPatronFromFileContent method of *OlePatronConverterService* class is called. Here, the buildPatronFromFileContent method of *OlePatronRecordHandler* class creates patron records from the file content.

The persistPatronFromFileContent method checks for the presence of the patronID in the record and based on the presence of the Id in the existing list, it is added to either create or update list. Depending on a boolean value, *addUnMatchedPatronFlag*, in the method, those records without a patronID are added to the create list or rejected list. By default, the addUnMatchedPatronFlag value is true and is hard coded and hence cannot be configured.

If the failed list contains one or more records, it is converted back to XML and saved as a failure list.

Circulation Policy, Ingest Profile and eResources

In case of circulation policy, Ingest Profile and eresources, *LoadDefaultCirculationPoliciesBean, LoadDefaultIngestProfileBean* and *LoadDefaultEResourcesBean* are the classes where the process is initiated, respectively. Similar to Patron, based on System Parameter and the arguments passed in the method, the persistKrmsFromFileContent method of *OleKrmsBuilder* class is called. From here the buildKrmsFromFileContent method of *OleKrmsObjectGeneratorFromXML* class is called wherein the file content is converted to POJO. The OleKrmsAgenda is extracted from this POJO and sent as parameter to the persist method of *OleKrmsBuilder* class.
In here, the doesKrmsExist method is used to check whether the agenda already exists. This is based on the agenda name. If it exists the existing agenda is deleted using the

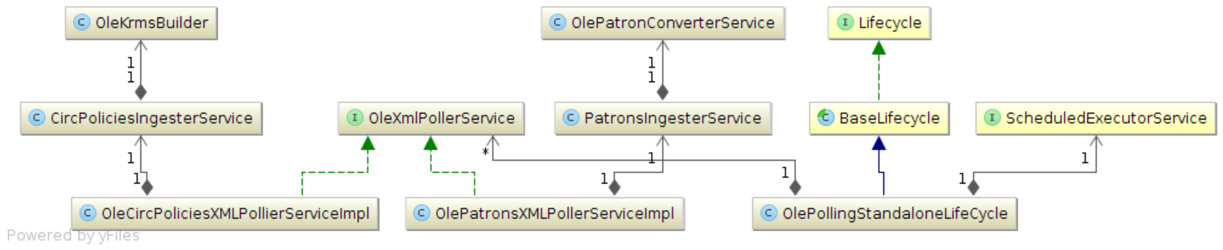
deleteAgenda method by passing the OleKrmsAgenda. Else the OleAgenda is persisted along with the rules.

Licenses

In case of Licenses, *LoadDefaultLicensesBean* class is where the process is initiated. Based on the System Parameter and the arguments passed in the method, the persistKrmsFromFileContent method of *KrmsBuilder* class is called. Here, the buildKrmsFromFileContent method of KrmsObjectGeneratorFromXML class returns the POJO with data from the file content. The Agenda is extracted from the POJO and sent as an argument to the persist method of *KrmsBuilder* class. The doesKrmsExist method checks for presence of the agenda based on the agenda name and if it exists, removed by the deleteAgenda method. The createAgenda method persists the agenda, rules and propositions.

Polling

Polling is the process where the computer waits for an external device to check for its readiness. In OLE, the poller service checks periodically for any new files in a particular location and loads the data from the file. This is available for loading Patron and Circulation Policies.


OleKrmsBuilder
OlePatronConverterService
Lifecycle
CircPoliciesIngesterService
OleXmlPollerService
PatronsIngesterService
BaseLifecycle
ScheduledExecutorService
OleCircPoliciesXMLPollierServiceImpl
OlePatronsXMLPollerServiceImpl
OlePollingStandaloneLifeCycle
Powered by yFiles

The location information is configured in *olefs-config-defaults.xml* file under *patrons.xml.root.location* and *circ.xml.root.location.* The initial delay is 30 seconds and poll interval is every 5 minutes. This is hardcoded in the *OleXmlPollerServiceImpl* class.

For more information, Javadocs can be found here.

Service Interface Design (REST if applicable)

Not Applicable

User Interface Design/Workflow

Not Applicable.

Last Published May 7, 2015

Data Importing

Not Applicable

Data Exporting (if applicable)

OLE uses a RDBMS backend and hence any data can be exported using simple SQL queries.

System Parameters

Namespace Code	Parameter Name	Description
OLE-DLVR	LOAD_DEFAULT_PATRONS_IND	The parameter controls whether the application ingests the default patrons in the next startup. Valid values are ‘Y’ and ‘N’.
OLE-DLVR	LOAD_DEFAULT_CIRCULATION_POLICIES_IND	The parameter controls whether the application ingests the default circulation policy in the next startup. Valid values are ‘Y’ and ‘N’.
OLE-DLVR	LOAD_DEFAULT_EREC_IND	The parameter controls whether the application ingests the default eResources in the next startup. Valid values are ‘Y’ and ‘N’.
OLE-DLVR	LOAD_DEFAULT_INGEST_POLICIES_IND	The parameter controls whether the application ingests the default ingest policies in the next startup. Valid values are ‘Y’ and ‘N’.
OLE-DLVR	LOAD_DEFAULT_LICENSES_IND	The parameter controls whether the application ingests the default licenses in the next startup. Valid values are ‘Y’ and ‘N’.

Roles and Permissions

This process is automated and happens at server startup and hence no roles and permissions are involved.

Last Published May 7, 2015