Table of Content

1 Overview

This document is a Technical Design document for the **Staging Component for Up- and Download of Digital Content** (staging) of the eSciDoc infrastructure. For the first release, the staging component only supports HTTP based up- and downloads of digital objects.

1.1 General

The Staging Component consists of two webservices offering methods to prepare an upload or a download and two servlets managing the tasks of uploading and downloading digital content to and from the eSciDoc framework.

1.2 Class Diagram

Figure 1 shows all relevant classes for the eSciDoc staging component.

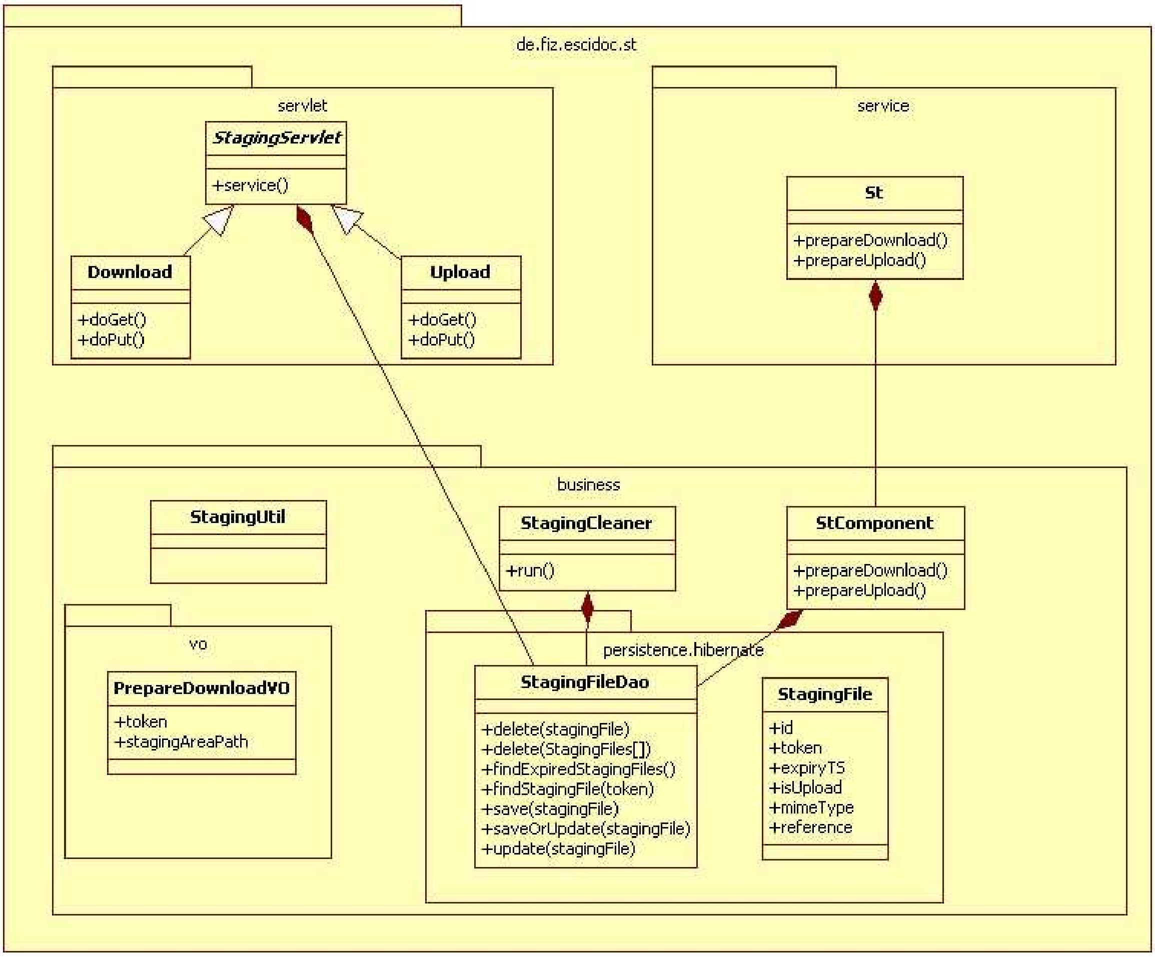


Figure 1: Class Diagram for Staging Component

1.2.1 Download

This class implements the staging servlet responsible for the download of digital objects from the framework. For the first release, this servlet only supports the HTTP protocol.

1.2.2 Upload

This class implements the staging servlet responsible for the upload of digital objects to the framework. For the first release, this servlet only supports the HTTP protocol.

1.2.3 St

This class implements the St service interface and provides the services prepareUpload and prepareDownload.

1.2.4 StComponent

This is the business layer component of the staging area.

1.2.5 Staging File

The class *StagingFile* holds all the information about uploading and downloading of digital content. The *token* is used to identify a specific upload/download operation. The token expiry timestamp (*expiryTS*) is used to protect the digital content and to decide if files in the staging area can be removed. The reference points to the file in the local file system, if it has been uploaded. Additionally, the StagingFile holds information about the transfer direction (*isUpload*) and the file's mime type (*mimeType*), if this is known.

1.2.6 Staging File Dao

StagingFile objects are stored in a database. Access to these objects is realized through staging file data access objects (*StagingFileDao*). The class StagingFileDao provides methods for saving, updating, deleting and retrieving StagingFile objects.

1.2.7 Staging Cleaner

The class *StagingCleaner* can be used to clean up the staging area. It removes outdated StagingArea objects from the database and its files from the file system.
This class extends java.util.TimerTask which enables the usage of StagingCleaner as a scheduled Task in the spring framework. The following code shows this, assuming the StagingCleaner is defined as a spring bean with id “st.StagingCleaner”:

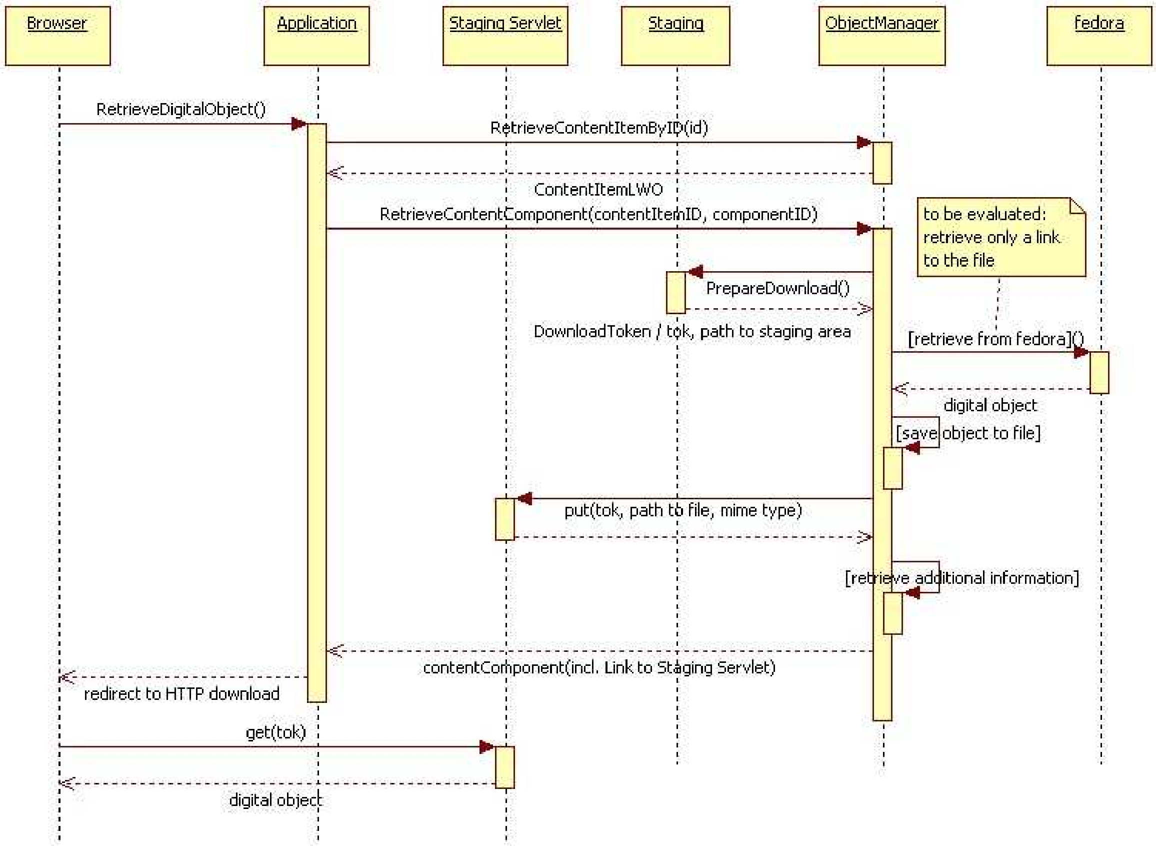
```
<bean id="st.ScheduledTask"
class="org.springframework.scheduling.timer.ScheduledTimerTask">
  <property name="delay" value="3600000"/>
  <property name="period" value="3600000"/>
  <property name="timerTask" ref="st.StagingCleaner"/>
</bean>

<bean id="st.TimerFactory"
class="org.springframework.scheduling.timer.TimerFactoryBean">
    <property name="scheduledTimerTasks">
        <list>
            <ref bean="st.ScheduledTask"/>
        </list>
    </property>
    <property name="daemon" value="false"/>
</bean>
```

1.3 Sequence Diagrams

This section shows the sequence diagrams describing the staging component and staging servlets in detail. To introduce into the overall context of the eSciDoc framework and its staging component, the next two figures repeat the sequence of downloading and uploading digital content. This can be found more detailed in the functional specification of the ST component.

Figure 2 shows the sequence diagram for the download of digital content from the eSciDoc framework, and Figure 3 shows the sequence diagram for the upload of digital content to the eSciDoc framework.



Browser
Application
Staging Servlet
Staging
ObjectManager
fedora
RetrieveDigitalObject()
RetrieveContentItemByID(id)
ContentItemLWO
RetrieveContentComponent(contentItemID, componentID)
to be evaluated: retrieve only a link to the file
PrepareDownload()
DownloadToken / tok, path to staging area
[retrieve from fedora]()
digital object
[save object to file]
put(tok, path to file, mime type)
[retrieve additional information]
contentComponent(incl. Link to Staging Servlet)
redirect to HTTP download
get(tok)
digital object

Figure 2: Sequence diagram for the download of digital content

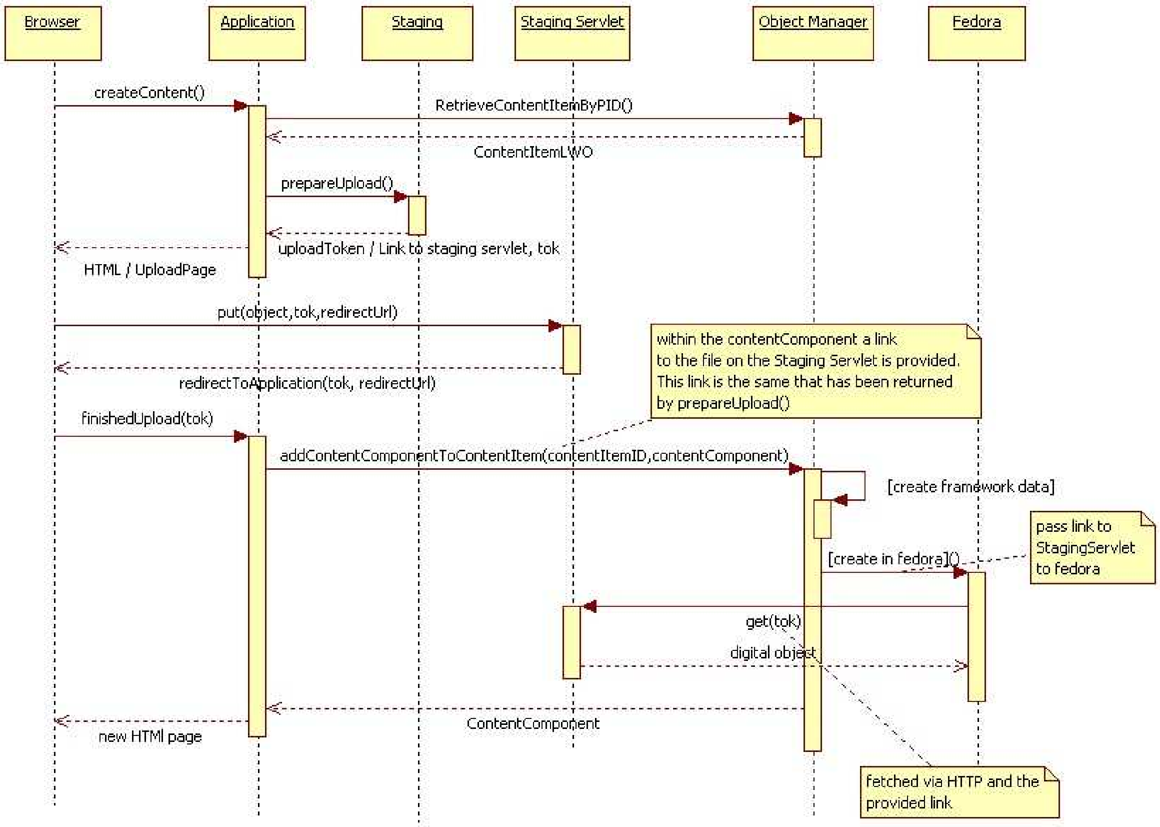

Browser
Application
Staging
Staging Servlet
Object Manager
Fedora
createContent()
RetrieveContentItemByPID()
ContentItemLWO
prepareUpload()
uploadToken / Link to staging servlet, tok
HTML / UploadPage
put(object,tok,redirectUrl)
redirectToApplication(tok, redirectUrl)
within the contentComponent a link to the file on the Staging Servlet is provided. This link is the same that has been returned by prepareUpload()
finishedUpload(tok)
addContentComponentToContentItem(contentItemID,contentComponent)
[create framework data]
pass link to StagingServlet to fedora
[create in fedora]()
get(tok)
digital object
ContentComponent
new HTMl page
fetched via HTTP and the provided link

Figure 3: Sequence diagram for the upload of digital content

1.3.1 Preparing File Downloading

Figure 4 shows the sequence diagram for the preparing of a download of digital content from the eSciDoc framework. The prepareDownload service is for framework internal use. It is called by the object manager during handling a retrieve content component request from an application.

The ST component creates a new stagingFile object and generates a unique token to identify the data transfer. The stagingFile object is passed to the data access object and stored into the database. The service returns a PrepareDownloadVo value object including the generated token and a path to a folder in the local file system, into which the object manager shall store the download file.

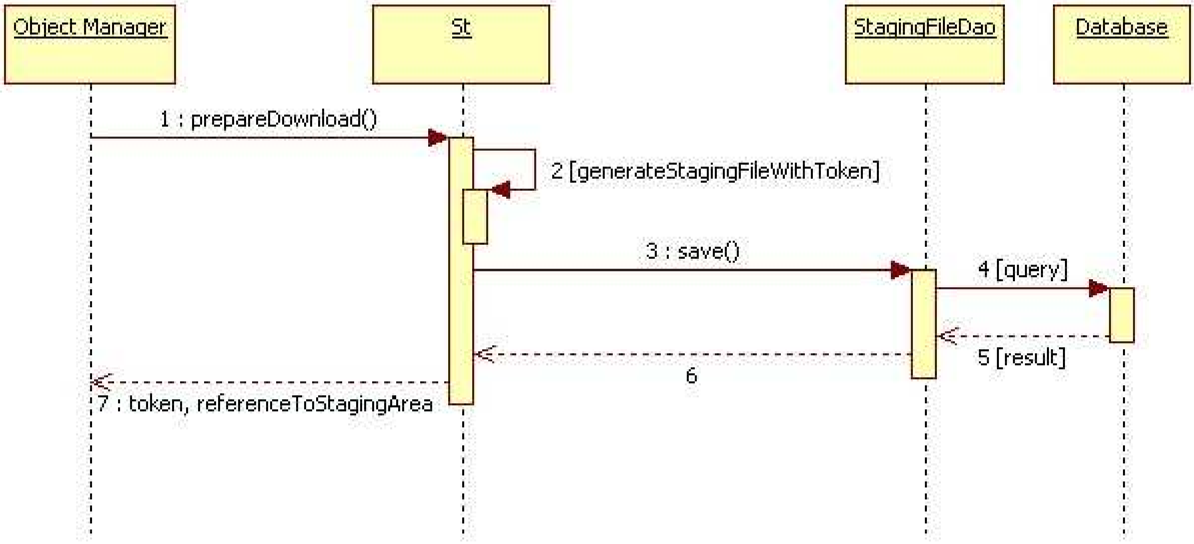


Figure 4: Sequence diagram for preparing download

1.3.2 Preparing File Uploading

Figure 5 shows the sequence diagram for the preparing of an upload of digital content to the eSciDoc framework. The prepareUpload service call is handled similar to the prepareDownload call with twodifferences: The stagingFile object is marked for upload instead of download and the service returns the URL to the upload staging servlet including the token as a parameter.

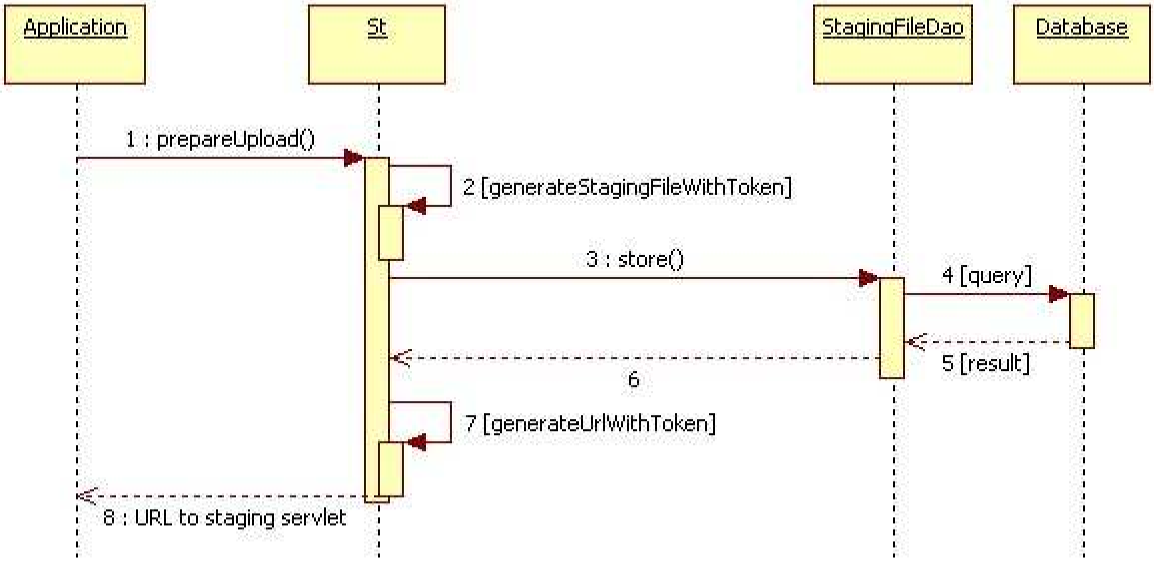


Figure 5: Sequence diagram for preparing upload

1.3.3 Putting a file into the download area

Figure 6 shows the sequence diagram for putting digital content into the download area of the eSciDoc framework. The put method of the servlet is called by the object manager after it has prepared the download and stored the file into the staging area's download folder. The request contains the token, the reference to the stored file, and the mime type of the file. The download servlet uses the provided token to retrieve the stagingFile object of this download transfer from the data access object. After successfully validating the stagingFile object, i.e. asserting that the token is not expired and there has no file been put in the staging area before, the servlet checks the file reference, i.e. if the file exists and is readable. Then it sets the file reference and the mime type in the stagingFile object and stores the new values in the database by calling the data access objects update method. Finally, it returns with the http status code OK (200).

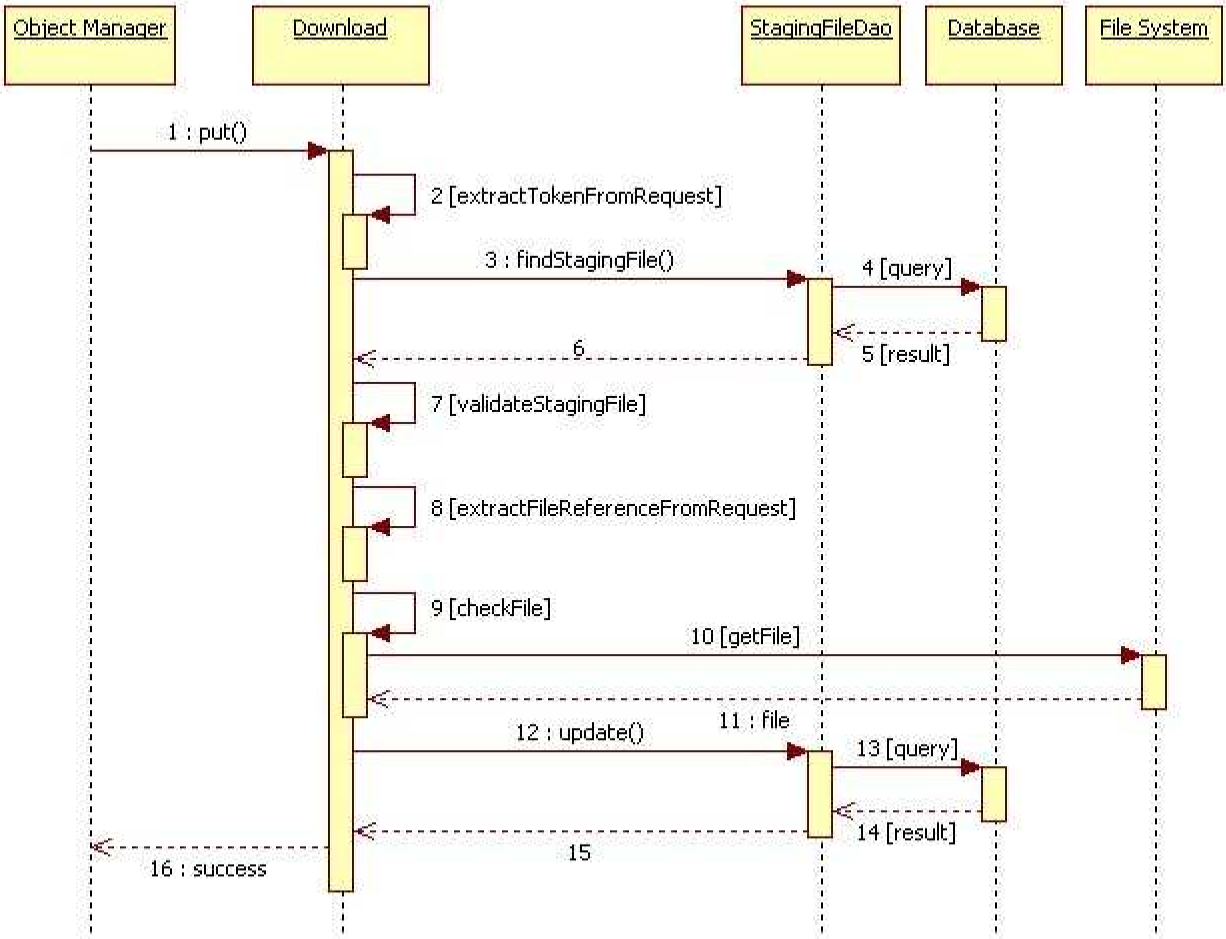


Figure 6: Sequence diagram for putting digital content into the download area

1.3.4 Putting a file into the upload area

Figure 7 shows the sequence diagram for putting digital content into the upload area of the eSciDoc framework. This method is called by the user's browser after the application has redirected it to the servlet. The request contains a token identifying the data transfer and a redirect URL linking back to the application.
The servlet gets the token from the request and retrieves the related stagingFile object from the data access object. After validating the stagingFile object (token has not expired and no file has been put into the staging area before), retrieving the digital content from the request's http body and storing it to a file in the local file system, the servlet sets the reference to the stored file in the staging file object and updates its persistent data by calling the data access object's update method. Finally, it takes the redirect URL from the request and sends a redirect to the application back to the browser. This redirect contains the token as a header value.

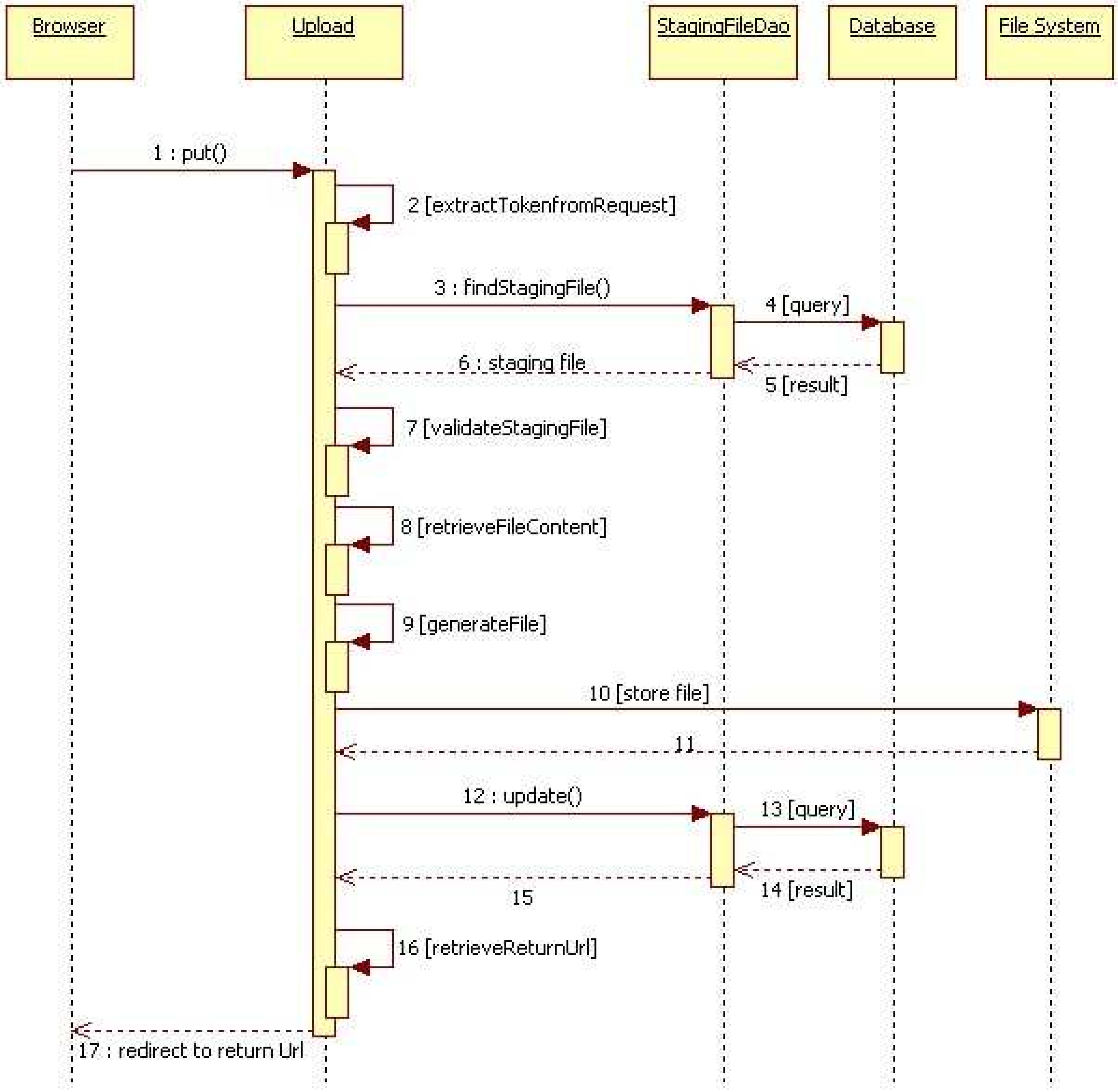


Figure 7: Sequence diagram for putting digital content into the upload area

1.3.5 Getting a file from the download area

Figure 8 shows the sequence diagram for getting digital content from the download area of the eSciDoc framework. This method is called by the user’s browser after the application has redirected it to the servlet during a retrieval of a content component.
The servlet gets the token from the request and retrieves the related stagingFile object from the data access object. After validating the stagingFile object (token has not expired and a readable file exists for this download data transfer), the servlet takes the file from the local file system and puts its content into the response body that is sent back to the browser.

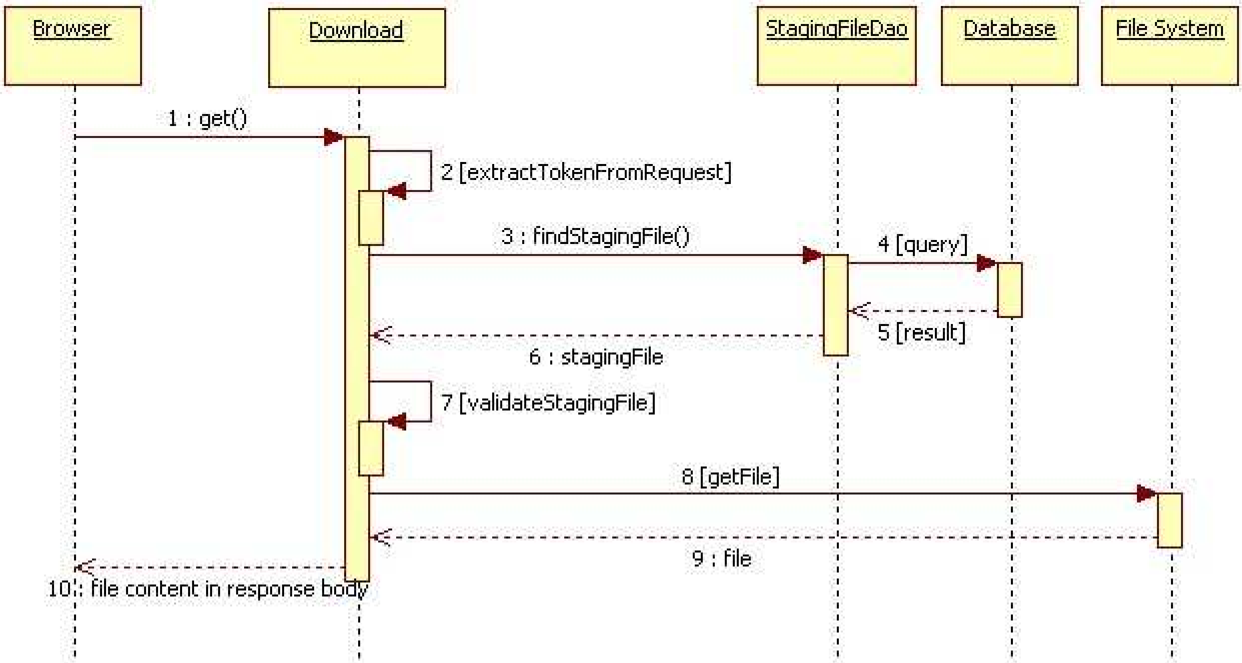


Figure 8: Sequence diagram for getting digital content from the download area

1.3.6 Getting a file from the upload area

Figure 9 shows the sequence diagram for getting digital content from the upload area of the eSciDoc framework. This method is called by the fedora repository during adding a content component to a content item.
The servlet gets the token from the request and retrieves the related stagingFile object from the data access object. After validating the stagingFile object (token has not expired and a readable file exists for this upload data transfer), the servlet takes the file from the local file system and puts its content into the response body that is sent back to fedora.

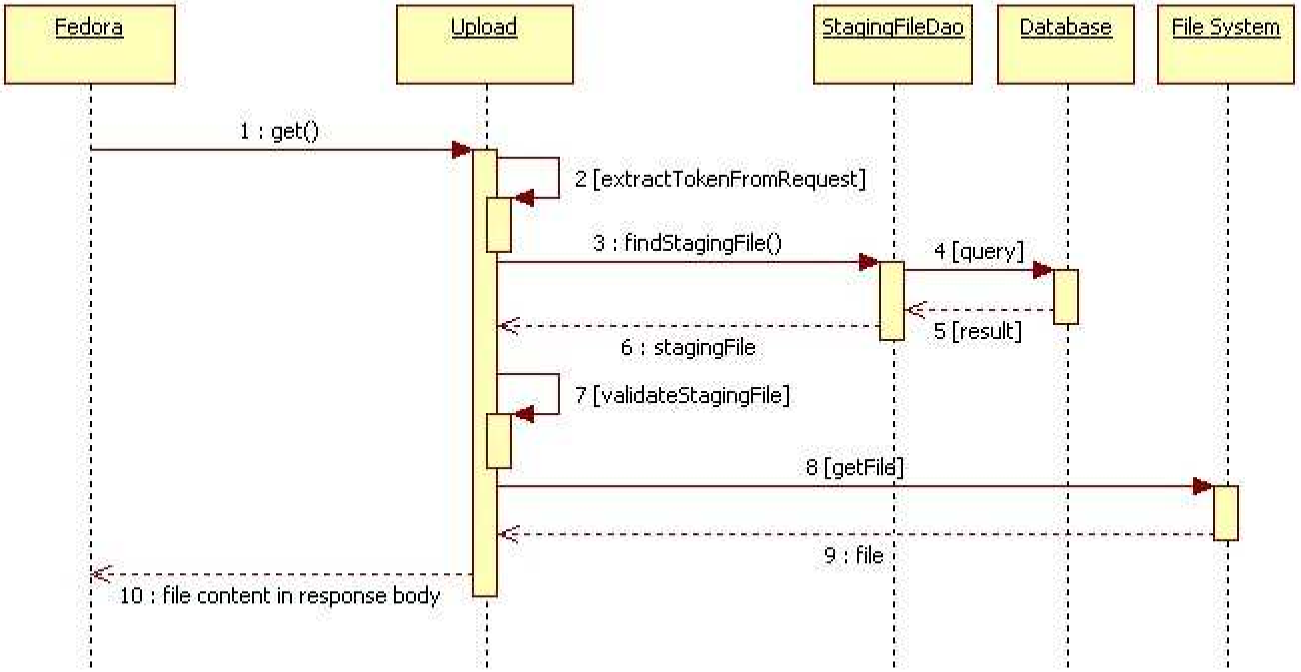


Figure 9: Sequence diagram for getting digital content from the upload area