**Changelog:**

|  |  |  |
| --- | --- | --- |
| ***Date*** | ***Author*** | ***Description*** |
| 2010-02-23 | PH | Separation of the File Transport Service documentation as a separate working document |
| 2010-03-03 | KK | Describe the core services |
| 2010-03-04 | KK | Adding information about authorization to the Web Site |
| 2011-11-06 | PH | New structure of FTPS directories and access accounts |
| 2012-01-16 | PH | Corrections of ambiguities; FTPS account data |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# File Transport Service

This section describes example file transfer façade operations implemented for a Java client. At the same time, however, they illustrate the general rules of using file transfer mechanisms by HMDB and HMDBMed databases. Regardless of the mechanisms described here, it is necessary to use the operations of the database's web services, controlling the downloading of files from and uploading to the appropriate databases.

Below we will summarize the instructions used by our tool implemented by the standard file transport service.

An implementation of the use of file transport services is available as part of the DatabaseConnection database client.

## Upload File (URI)

An example of an operation for uploading a file in the façade:

public void uploadSessionFile(int sessionId, String description, String localFilePath) throws Exception

sessionId – identifier of the session to which the file is to be uploaded

description – description of the file

localFilePath – The path to a file on the local disk

### Implementation

To upload a session file to the server, the following operations are performed:

1. Via FTPS, the file is uploaded to the primary directory made available for remote saving by the user. This directory is the /BDR/w directory. To access it, you need to use the username bdrWriter and the password, which is currently bdrWrite.
2. If the operation is successful, a Web Service operation is invoked that moves the uploaded file to the database within the server. In the case of the above-mentioned façade operation, it will be the operation:  
   StoreSessionFile( sessionId, remoteFilePath, description, remoteFileName ) sessionId – session identifier remoteFilePath – path to a file on the server (currently an empty string) description – description of the file remoteFileName – the name of the file within the path given aboveAuthorization to the Web Service should be done for the user in accordance with the account created for a given user (it is not a PJWSTK domain account). The authentication mode mentioned here applies to services in the target variant operating over HTTPS.
3. Once this is done correctly, the file will be pulled into the database and will become available normally

## Download file (URI)

**public** String downloadSessionFile(**int** fileID, String destLocalFolder) **throws** Exception

fileID – session file id  
 destLocalFolder – the path where the file is to be recorded

return value – the full path to the file locally after downloading it from the server

### Implementation

To download the file, perform the following operations:

1. You need to "extract" the file locally from the database on the server. To do this, we call Web Service   
   String retrieveFile(fileID)  
    fileID – session file identifier (to be retrieved by the listSessionFiles service)  
    return value – path to a file on the server in the ftp account of the user testUser.Authorization to the Web Service is the same as in the case of uploading a file to the server.
2. After this operation, a downloadable file will appear in the ftp directory of the testUser user. The exact file addresses are returned as a String value when the above service is invoked.
3. To download the file, you need to connect to the server with the appropriate user account using FTPS.

## List of user accounts

Due to the necessity to provide reading and writing for each of the two databases: HMDB and HMDBMed – a total of 4 accounts with different permissions and home directories were created.

These are the following 4 FTP accounts (login/password):

Badruser / BDRAD

Badrite / Badri

medUser / medRead

medWriter / medWrite

The "writer" accounts also have permissions to the parent directory to be read.

The following FTP subdirectory structure corresponds to the use of these accounts.

/BDR

/BDR/w

/MED

/MED/w