```
14
15 /**
16  *
17  * @author cm
18  */
19 public class FileServlet extends HttpServlet {
20
21     /**
22      * Processes requests for both HTTP <code>GET</code> and <code>POST</code> methods.
23      * @param request servlet request
24      * @param response servlet response
25      * @throws ServletException if a servlet-specific error occurs
26      * @throws IOException if an I/O error occurs
27      */
28     private static final Logger logger = Logger.getLogger(FileServlet.class.getName());
29     public static final String FILE_DIRECTORY = "opr.FILE_DIRECTORY";
30
31     @EJB
32     private FileLocal fb;
33
34     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
35             throws ServletException, IOException {
36
37         OutputStream os = null;
38         try {
39         String requestedFile = request.getPathInfo().substring(1);
40         String filename = requestedFile;
41
42         int size = -1;
43         boolean scaleCubic = false;
44         if (requestedFile.matches("(xy)?[1-9][0-9]*px-.*")) {
45
46             if (requestedFile.startsWith("xy")) {
47                 scaleCubic = true;
48                 requestedFile = requestedFile.substring(2);
49             }
50
51             String[] parts = requestedFile.split("px-");
52             try {
53                 size = Integer.parseInt(parts[0]);
54             } catch (NumberFormatException ex) {
55                 size = -1;
56             }
57
58             if(parts.length == 2) {
59                 filename = parts[1];
60             }
61         }
62
63         long id;
64         try {
65             id = Long.parseLong(filename);
66         } catch (NumberFormatException e) {
67             response.sendError(404);
68             return;
69         }
70
71         File file = fb.getById(id);
72
73
74         if (file == null) {
75             response.sendError(404);
76             return;
77         }
78
```

```
79          response.setContentType(file.getMime());
80          response.addHeader("Content-Disposition", "filename="+ file.getName() +";");
81          os = response.getOutputStream();
82
83
84          if (size > 0 && file != null) {
85
86              byte[] bytes;
87              if( (bytes = fb.getThumbnail(file, size, scaleCubic)) != null) {
88                  os.write(bytes);
89              }
90
91              return;
92          }
93
94
95          os.write(file.getContent());
96
97          return;
98
99          } finally {
100             try {
101                 os.close();
102             } catch(Exception ex) {
103                 logger.log(Level.SEVERE, null, ex);
104             }
105         }
106
107     }
108
109     // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">
110     /**
111      * Handles the HTTP <code>GET</code> method.
112      * @param request servlet request
113      * @param response servlet response
114      * @throws ServletException if a servlet-specific error occurs
115      * @throws IOException if an I/O error occurs
116      */
117     @Override
118     protected void doGet(HttpServletRequest request, HttpServletResponse response)
119             throws ServletException, IOException {
120         processRequest(request, response);
121     }
122
123     /**
124      * Handles the HTTP <code>POST</code> method.
125      * @param request servlet request
126      * @param response servlet response
127      * @throws ServletException if a servlet-specific error occurs
128      * @throws IOException if an I/O error occurs
129      */
130     @Override
131     protected void doPost(HttpServletRequest request, HttpServletResponse response)
132             throws ServletException, IOException {
133         processRequest(request, response);
134     }
135
136     /**
137      * Returns a short description of the servlet.
138      * @return a String containing servlet description
139      */
140     @Override
141     public String getServletInfo() {
142         return "Short description";
143     }// </editor-fold>
144 }
145
146
```