**JSF安全動態圖片顯示**

1. Servlet部分:

|  |
| --- |
| /\*\*  \* 顯示圖片 Servlet  \*/  @WebServlet(name = "ImageServlet", urlPatterns = {"/ImageServlet"})  public class ImageServlet extends HttpServlet {  public final Logger logger = LoggerFactory.getLogger(ImageServlet.class);    @Override  protected void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {  String showImgKey = request.getParameter("**showImgKey**");// 安全檢核碼  String fileName = request.getParameter("**fileName**");// 完整實體路徑  String contentType = request.getParameter("**contentType**");// 檔案類型  String fileExt = request.getParameter("**fileExt**");// 副檔名  HttpSession session = request.getSession();  if( session!=null ){  // 安全檢核  **if( session.getAttribute(showImgKey)!=null && ((Boolean)session.getAttribute(showImgKey)) ){**  logger.debug("ImageServlet session.getLastAccessedTime() = "+session.getLastAccessedTime());  // 檔案輸出至 OutputStream  response.setContentType(contentType);  File f = new File(fileName);  BufferedImage bi = ImageIO.read(f);  OutputStream out = response.getOutputStream();  ImageIO.write(bi, fileExt, out);  out.close();  }  }  }  } |

1. JSF 部分:

|  |
| --- |
| <!-- 顯示圖片 -->  <h:panelGroup rendered="#{projectController.prepareImage(row1)}" >  <img **src**="#{projectController.getImageSrc(row1)}" width="50" height="50" align="middle" />  </h:panelGroup> |

1. Managed Bean部分:

|  |
| --- |
| // 顯示圖片  private Map<Long, String> imageSrcMap;// 存放每筆資料圖片顯示URL  private String showImgKey; // 安全檢核碼  **@PostConstruct**  public void init() {  // 顯示圖片URL  imageSrcMap = new HashMap<Long, String>();  showImgKey = UUID.randomUUID().toString();// 驗證碼  **JsfUtils.setSessionAttr(showImgKey, Boolean.TRUE, false); // 驗證碼放行**  }  **@PreDestroy**  public void onDestroy(){  **JsfUtils.removeSessionAttr(showImgKey);// 移除驗證碼**  }  /\*\*  \* 準備顯示圖片 URL 資訊  \* @param projectDocumentVO  \* @return  \*/  public boolean prepareImage(ProjectDocumentVO projectDocumentVO){  // 圖片URL  StringBuilder url = new StringBuilder(JsfUtils.getContextPath()).append("**/ImageServlet**");  // 附件資訊  editDocument = pmisDocumentFacade.find(projectDocumentVO.getId());  **AttachmentVO** attachmentVO = fileController.getLastAttachment(editDocument);  // 產出完整圖片URL  if( attachmentVO!=null ){  String contentType = attachmentVO.getContentType();  String fullFileName = fileController.**getFullFileName(attachmentVO)**;// 完整實體名稱  if( contentType!=null && fullFileName!=null && contentType.toLowerCase().startsWith("image/") ){  try{  String fileExt = fullFileName.lastIndexOf(".")>=0? fullFileName.substring(fullFileName.lastIndexOf(".")+1):"";  url.append("?fileName=").append(URLEncoder.encode(fullFileName, "UTF-8"));  url.append("&contentType=").append(URLEncoder.encode(contentType, "UTF-8"));  url.append("&fileExt=").append(URLEncoder.encode(fileExt, "UTF-8"));  url.append("&showImgKey=").append(URLEncoder.encode(showImgKey, "UTF-8"));  logger.debug("prepareImage url = "+url.toString());    // 文件ID 與 圖片URL 對應表  **imageSrcMap.put(projectDocumentVO.getId(), url.toString());**  return true;  }catch(Exception e){  logger.error("prepareImage Exception ...\n", e);  }  }  }  return false;  }  /\*\*  \* 取得指定圖片 URL  \* @param projectDocumentVO  \* @return  \*/  public String getImageSrc(ProjectDocumentVO projectDocumentVO){  return (imageSrcMap!=null)?**imageSrcMap.get(projectDocumentVO.getId())**:null;  } |