# **Установка библиотеки usercheck в приложение**

(На 19.11.2018 последняя версия лежит в папке U:\G\_NIIAS\_ABD\_ARM\AUPH\_USER\ShirmanEP\NSI\USERCHECK20181119)

1. Состоит из двух библиотек UCheck-1.0.jar, UCheck-1.0-client.jar
2. UCheck-1.0.jar деплоится на WildFly. На сервере приложений должен быть JNDI java/CheckDS
3. UCheck-1.0-client.jar ставится в приложение (так как в интерфейс добавился новая команда remove(), нужно установить клиет-jar заново)
   1. Копируется сама библиотека в папку ../имяприложения/имя приложения-web/lib
   2. В pom.xml добавляется зависимость

<dependency>

<groupId>check</groupId>

<artifactId>check</artifactId>

<version>1.0</version>

<type>jar</type>

</dependency>

Находим в зависимостях check-1.0 и из контекстного меню выбираем установить артефакт вручную. Выбираем UCheck-1.0-client.jar лежащий в папке lib.

Пример использования в фильтре. (Замечание прямой lookup ejb убираем):

*import java.io.IOException;*

*import javax.ejb.EJB;*

*import javax.inject.Inject;*

*import javax.servlet.Filter;*

*import javax.servlet.FilterChain;*

*import javax.servlet.FilterConfig;*

*import javax.servlet.RequestDispatcher;*

*import javax.servlet.ServletException;*

*import javax.servlet.ServletRequest;*

*import javax.servlet.ServletResponse;*

*import javax.servlet.http.HttpServletRequest;*

*import javax.servlet.http.HttpServletResponse;*

*import com.vniizht.ucheck.UserCheckRemote;*

*import javax.naming.InitialContext;*

*public class SetCharacterEncodingFilter implements Filter {*

*protected String encoding = null;*

*protected FilterConfig filterConfig = null;*

*protected boolean ignore = true;*

*// убираем lookup*

*//@EJB //(lookup = "java:global/UCheck-1.0/UserCheck!com.vniizht.ucheck.UserCheckRemote")*

*//private UserCheckRemote ejbUserCheck;*

*@Inject*

*transient private PrilForms prilForms;*

*// private UserCheck getUserCheck() {*

*// return ejbUserCheck;*

*// }*

*public PrilForms getPrilForms() {*

*return this.prilForms;*

*}*

*public void setPrilForms(PrilForms prilForms) {*

*this.prilForms = prilForms;*

*}*

*/\*\**

*\* Take this filter out of service.*

*\*/*

*public void destroy() {*

*this.encoding = null;*

*this.filterConfig = null;*

*}*

*/\*\**

*\* Select and set (if specified) the character encoding to be used to*

*\* interpret request parameters for this request.*

*\**

*\* @param request The servlet request we are processing*

*\* @param result The servlet response we are creating*

*\* @param chain The filter chain we are processing*

*\* @exception IOException if an input/output error occurs*

*\* @exception ServletException if a servlet error occurs*

*\*/*

*public void doFilter(ServletRequest request, ServletResponse response,*

*FilterChain chain)*

*throws IOException, ServletException {*

*HttpServletResponse httpRes = (HttpServletResponse) response;*

*HttpServletRequest httpReq = (HttpServletRequest) request;*

*httpRes.setHeader("Cache-Control", "no-cache, no-store, must-revalidate"); // HTTP 1.1.*

*httpRes.setHeader("Pragma", "no-cache"); // HTTP 1.0.*

*httpRes.setDateHeader("Expires", 0); // Proxies.*

*UserCheckRemote ejbUserCheck = null;*

*try {*

*final InitialContext ic = new InitialContext();*

*ejbUserCheck = (UserCheckRemote) ic.lookup("java:global/UCheck-1.0/UserCheck!com.vniizht.ucheck.UserCheckRemote");*

*// Раскладываем адрес на составляющие*

*String[] list = httpReq.getRequestURI().split("/");*

*// Извлекаем наименование страницы*

*String page = "";*

*if (list[list.length - 1].indexOf(".xhtml") > 0) {*

*page = list[list.length - 1];*

*}*

*//*

*if (!page.equalsIgnoreCase("info.xhtml")) {*

*ejbUserCheck.setUser(httpReq.getRemoteUser());*

*ejbUserCheck.setIp(httpReq.getRemoteAddr());*

*ejbUserCheck.setTaskCode("NSI");*

*ejbUserCheck.setStatTaskCode("NSI");*

*if(prilForms != null) {*

*ejbUserCheck.setIdUser(prilForms.getIduser());*

*prilForms.setNameuser(httpReq.getRemoteUser());*

*}*

*//ejbUserCheck.checkMenu();*

*if(!ejbUserCheck.check()) {*

*returnError(httpReq, httpRes, new Exception(ejbUserCheck.getMessage()));*

*return;*

*}*

*if(prilForms != null) {*

*prilForms.setPod(ejbUserCheck.getUserPod());*

*prilForms.setIduser(ejbUserCheck.getIdUser());*

*}*

*}*

*// Conditionally select and set the character encoding to be used*

*if (ignore || (request.getCharacterEncoding() == null)) {*

*String encoding = selectEncoding(request);*

*if (encoding != null) {*

*request.setCharacterEncoding(encoding);*

*}*

*}*

*response.setContentType("text/html; charset=UTF-8");*

*response.setCharacterEncoding("UTF-8");*

*// Pass control on to the next filter*

*chain.doFilter(request, response);*

*} catch (Exception ex) {*

*RequestDispatcher rd = request.getRequestDispatcher("/error.jsp");*

*request.setAttribute("error", ex);*

*rd.forward(request, response);*

***} finally {***

***// Добавляем удаление подключенного ejb***

***ejbUserCheck.remove();***

***}***

*}*

*/\*\**

*\* Place this filter into service.*

*\**

*\* @param filterConfig The filter configuration object*

*\*/*

*public void init(FilterConfig filterConfig) throws ServletException {*

*this.filterConfig = filterConfig;*

*this.encoding = filterConfig.getInitParameter("encoding");*

*String value = filterConfig.getInitParameter("ignore");*

*if (value == null) {*

*this.ignore = true;*

*} else if (value.equalsIgnoreCase("true")) {*

*this.ignore = true;*

*} else if (value.equalsIgnoreCase("yes")) {*

*this.ignore = true;*

*} else {*

*this.ignore = false;*

*}*

*}*

*/\*\**

*\* Select an appropriate character encoding to be used, based on the*

*\* characteristics of the current request and/or filter initialization*

*\* parameters. If no character encoding should be set, return*

*\* <code>null</code>. \* The default implementation unconditionally returns*

*\* the value configured by the <strong>encoding</strong> initialization*

*\* parameter for this filter.*

*\**

*\* @param request The servlet request we are processing*

*\*/*

*protected String selectEncoding(ServletRequest request) {*

*return (this.encoding);*

*}*

*private void returnError(HttpServletRequest request, HttpServletResponse response, Exception ex) {*

*try {*

*RequestDispatcher rd = request.getRequestDispatcher("/prohibited.jsp");*

*request.setAttribute("prohibited", ex);*

*rd.forward(request, response);*

*return;*

*} catch (Exception e) {*

*}*

*}*

*}*

Для подсчета времени работы в программе в отдельном бине, у меня это PrilForms, нужно сохранять переменную userid:

*public class PrilForms implements Serializable {*

*private Integer iduser = 0;*

*public Integer getIduser() {*

*return iduser;*

*}*

*public void setIduser(Integer iduser) {*

*this.iduser = iduser;*

*}*

*}*

web.xml

<security-constraint>

<display-name>Ограничение1</display-name>

<web-resource-collection>

<web-resource-name>all</web-resource-name>

<description>secure all content</description>

<url-pattern>/\*</url-pattern>

</web-resource-collection>

<auth-constraint>

<description>auth\_constraint</description>

<role-name>authentificated\_user</role-name>

</auth-constraint>

</security-constraint>

<login-config>

<auth-method>BASIC</auth-method>

<realm-name>UnderSecDomain</realm-name>

</login-config>

<security-role>

<description/>

<role-name>authentificated\_user</role-name>

</security-role>

jboss-web.xml

<?xml version="1.0" encoding="UTF-8"?>

<jboss-web version="8.0" xmlns="http://www.jboss.com/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.jboss.com/xml/ns/javaee http://www.jboss.org/schema/jbossas/jboss-web\_8\_0.xsd">

<context-root>/nsigos-web</context-root>

<security-domain>UnderSecDomain</security-domain>

</jboss-web>

# **Описание методов**

1. Проверка роли, если главный администратором – то true, иначе false

boolean isSysAdmin() throws Exception;

1. Версия программы

String getVersion();

1. Запись логина пользователя

void setUser(String u);

1. Запись кода pud.oarm.id

void setIdUser(Integer id);

1. Выдача кода записи pud.oarm.id

Integer getIdUser();

1. Сохранение ip пользователя

void setIp(String i) throws Exception;

1. Сохранение кода задачи пользователя

void setTaskCode(String t);

1. Сохранение кода задачи для статистики

void setStatTaskCode(String s);

1. Основной метод проверки доступа. Возвращает true, если доступ разрешен.

boolean check() throws Exception;

1. Дополнительный метод проверки доступа, без записи в логи. Возвращает true, если доступ разрешен.

boolean checkpart() throws Exception;

1. Основной метод проверки доступа при формировании меню. Возвращает true, если доступ разрешен.

boolean checkMenu() throws Exception;

1. Сохранение времени работы пользователя в системе и запись о выходе из системы

boolean SaveTimeExit() throws Exception;

1. Проверка на блокировку пользователя

boolean checkUserLock() throws Exception;

1. Проверка на блокировку задачи

boolean checkTaskLock() throws Exception;

1. Состояние базы (в разработке)

int getDbState() throws Exception;

1. Сообщение о состоянии базы (в разработке)

String getDbStateMessage() throws Exception;

1. Возвращает меню пользователя

ArrayList getUserMenu() throws Exception;

1. Возвращает список ролей пользователя

String getUserRole() throws Exception;

1. Возвращает список программ пользователя

String getUserPod() throws Exception;

1. Возвращает код дороги, к которой принадлежит пользователь

String getUserDor() throws Exception;

1. Возвращает государство пользователя

String getUserGos() throws Exception;

1. Возвращает OWNERPLASE

String getLocation() throws Exception;

1. Возвращает код перевозчика, к которому принадлежит пользователь

int getSkp() throws Exception;

1. Возвращает код агента, к которому принадлежит пользователь

int getAgent() throws Exception;

1. Возвращает код асупв (на данный момент типы пользователей)

int getAsupv() throws Exception;

1. Возвращает код подразделения, к которому принадлежит пользователь

String getUserPodr() throws Exception;

1. Возвращает фамилию пользователя

String getUserFamiliya() throws Exception;

1. Возвращает имя пользователя

String getUserImya() throws Exception;

1. Возвращает отчество пользователя

String getUserOtchestvo() throws Exception;

1. Возвращает должность пользователя

String getUserDolzhn() throws Exception;

1. Возвращает почту пользователя

String getUserEmail() throws Exception;

1. Возвращает код филиала, к которому принадлежит пользователь

String getUserFilial() throws Exception;

1. Возвращает код субагента, к которому принадлежит пользователь

int getUserSubagn() throws Exception;

1. Возвращает код ЕЛС пользователя

String getUserKodels() throws Exception;

1. Не используется на данный момент

int getPermission() throws Exception;

1. Возвращает сообщение usercheck

String getMessage();

1. Не используется на данный момент

String getMessageUser();

1. Право на чтение: 1 – есть, 0 - нет

int getRead() throws Exception;

1. Право на добавление: 1 – есть, 0 - нет

int getCreate() throws Exception;

1. Право на изменение: 1 – есть, 0 - нет

int getUpdate() throws Exception;

1. Право на удаление: 1 – есть, 0 - нет

int getDelete() throws Exception;

1. Государство АБД (в разработке на данный момент)

String getGosAbd() throws Exception;

1. Метод очистки всех введенных данных

void remove();

1. Метод определения прав: 1 – есть, 0 - нет, param может принимать значения:

read - чтение

create - создание

update - изменение

delete - удаление

load - загрузка

download – выгрузка

linpr – код линейного предприятия (добавлено 2021-08-02)

privilege – привилегия (добавлено 2021-10-01)

moderator – модератор приложения (добавлено 2021-10-01)

int getParamI(String param) throws Exception;

1. Метод для выдачи строковых данных по параметру. Param может принимать значения:

messageUZ – выдача сообщения о сроке действия учетной записи

jadm – ж/д администрация, \* - ЦСЖТ, код – конкретный код ж/д администрации. (добавлено 2021-08-02)

String getParamS(String param) throws Exception;

1. Выдача данных массивов по параметру (в разработке)

ArrayList getParamA(String param) throws Exception;

1. Выдача данных логических типов по параметру (в разработке)

Boolean getParamB(String param) throws Exception;

1. Возвращает список элементов по параметру. Param может принимать значения:

listodMap - возвращает список доступных кодов приложений

HashMap getParamHashMap(String param) throws Exception