

Технологическая база

1. IntelliJIdea Ultimate
2. Jrebel
3. maven
4. git (stash) -
5. JSF, PrimeFaces, PrettyFaces

логины - басовский емейл без @bas.kz

Jira: <https://rnd.bas.kz:8444/jira/secure/Dashboard.jspa>

Stash: <https://rnd.bas.kz:8444/stash/projects>

для Idea+JBoss

vmOptions= -Xms64M -Xmx512M -XX:MaxPermSize=256M

для Git перед инициализацией репозитория запустить:

git config --global http.sslVerify false

Правила работы с GIT

1. Add
2. Commit
3. Pull
- 3.1 Разрешить конфликты
4. Push

1. Если происходит конфликт с файлом, который к Вашей подсистеме не относится, то его необходимо целиком заменить в вашей локальной копии, вызвав в окне редактора конфликтов TortoiseGit в левой панели “Use this whole file”.

2. Если конфликт с Вашим файлом, то производите разрешение конфликтов в штатном режиме.

Вопросы:

1. Процесс авторизаций и аутентификаций
 - Предварительная загрузка статичных данных
 - Фильтр запросов по адресу (url)
 - Аутентификация пользователя в системе
 - Возврат текущего авторизованного пользователя (UserModel)
 - Чтение модели UserModel (все свойства, в том числе роли и права)

- Виды ошибок
- 2. Навигация в системе
 - Добавление нового адреса страницы
 - Переход по указанному адресу
- 3. Логирование событий системы
 - Типы событий в системе
 - Добавление событий по типу
- 4. Хлебные крошки (Breadcrumbs)
 - Компонент
 - Запись в историю переходов
- 5. Компоненты PrimeFaces 5.0
- 6. Идентификаторы сущностей

Общий чат

host: sop.bas.kz

domain: sop.bas.kz

Зарегистрируйтесь на сервере, примерные логины представлены ниже

Адилбеков Ернар	e.adilbekov
Гребенщиков Евгений	e.grebenschikov
Мерзентай Ергнат	yerganat.m
Ни Юрий	y.nee
Букша Вероника	v.buksha
Байдосов Самат	s.baidossov
Щупакова Алена	a.shupakova
Сыздыков Аслан	a.syzdykov
Гуч А.	a.guch

Пример заведения пользователя “AAA” в ICQ мессенджере Pidgin.

Приложение можете использовать любое, которое поддерживает Xmpp протокол(ICQ, Pidgin, и проч)

PS: пароль обязателен)

Изменить учётную запись

Основные

Дополнительно

Прокси

Параметры входа

Протокол:

XMPP

Имя пользователя:

aaa

Домен:

sop.bas.kz

Ресурс:

sop.bas.kz

Пароль:


☐ Запомнить пароль

Параметры пользователя

Локальный псевдоним:

☒ Уведомления о новой почте

☐ Использовать этот значок собеседника для этой учётной записи:

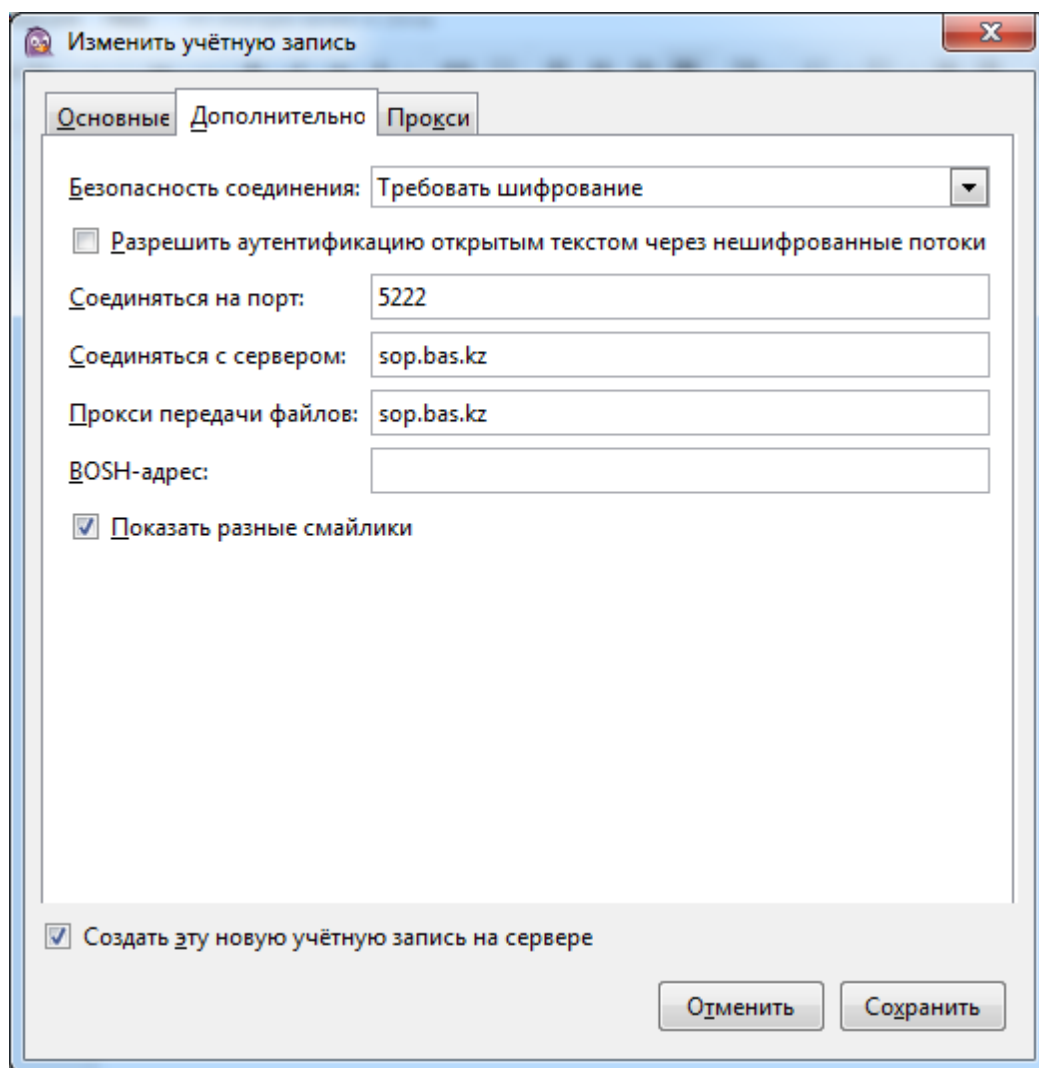


Удалить

☒ Создать эту новую учётную запись на сервере

Отменить

Сохранить



1. Процесс авторизации и аутентификации

Предварительная загрузка статических данных

SecurityHandler – регистрируется в конфигурационном файле. Контролирует слушаая определенные адреса браузера, которые указаны в фильтре:

```
<listener>  
  <listener-class>kz.bas.sop.servlet.SecurityHandler</listener-class>  
</listener>
```

Первая инициализация конфигурационного файла security-config.xml, где указаны, все пути в системе (подсистеме) с правами доступа.

```
@Override
public void init(FilterConfig config) throws ServletException {
    try {
        securityConfig = readConfig(config.getServletContext().getRealPath(SEcurity_CONFIG_FILE));
        String loginPage = "/" + StringUtils.defaultString(securityConfig.getLoginPage(), DEFAULT_LOGIN_PAGE);
        unsecuredUrls.add(loginPage);
        unsecuredUrls.addAll(securityConfig.getUnsecuredUrls());
        redirectURL = config.getServletContext().getContextPath() + loginPage;
        log.info("### SecurityConfig is successfully loaded.");
        //log.info("### " + securityConfig);
        config.getServletContext().setAttribute(SEcurity_CONFIG, securityConfig);
    } catch (FileNotFoundException e) {
        log.error(String.format("### %s loading error", SEcurity_CONFIG_FILE), e);
        e.printStackTrace();
    }

    //log.info("### SECURITY FILTER UNSECURED URLS : " + Arrays.toString(unsecuredUrls.toArray()) + " " + redirectURL);
    config.getServletContext().setAttribute(ACTIVE_SESSIONS, activeSessions);
}
```

Фильтр запросов по адресу (url)

При переходе на любой адрес (url), система фильтрует адреса по тем правилам, которые указаны в конфигурационном файле. При отработке методов класса SecurityHandler, мы получаем ответ:

- надо перенаправляет на страницу **логин**, если не авторизован
- показать ошибки при вторичном входе в систему, если есть.
- успешный переход по указанному адресу, если адрес существует и доступен текущему пользователю

Первая проверка пользователя при переходе по любому адресу! При проверке вызываются методы с внешних классов (SecurityServlet). SecurityServlet отвечает за аутентификацию и авторизацию пользователя, в том числе добавление cookie, session.

```
@Override
public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain) throws IOException, ServletException {
    HttpServletRequest httpRequest = (HttpServletRequest) request;
    HttpServletResponse httpResponse = (HttpServletResponse) response;
    String path = httpRequest.getServletPath();

    if(httpRequest.getDispatcherType() == DispatcherType.REQUEST && path.matches(".*\\.xhtml")) {
        if(!path.matches(".*\\.java\\.faces\\.resource\\..*")) {
            //log.info("UNMAPPED URL: " + httpRequest.getDispatcherType() + " " + path);
            httpResponse.sendError(HttpServletResponse.SC_FORBIDDEN, "Данный ресурс запрещен");
        } else {
            chain.doFilter(request, response);
        }
    } else {
        HttpSession session = httpRequest.getSession(false);
        if(!unsecuredUrls.contains(path)) {
            if(session != null) {
                Map<String, HttpSession> activeSessions = (Map<String, HttpSession>) request.getServletContext().getAttribute(ACTIVE_SESSIONS);
                if(activeSessions.containsKey(session.getId())) {
                    if(httpRequest.getDispatcherType() == DispatcherType.ERROR && httpResponse.getStatus() != HttpServletResponse.SC_OK) {
                        chain.doFilter(request, response);
                    } else {
                        int httpStatus = isAllowedResource(httpRequest);
                        if(httpStatus == HttpServletResponse.SC_OK) {
                            if(httpResponse.getStatus() != HttpServletResponse.SC_NOT_FOUND &&
                                httpResponse.getStatus() != HttpServletResponse.SC_FORBIDDEN &&
                                httpResponse.getStatus() != HttpServletResponse.SC_UNAUTHORIZED) {
                                processPageHistory((HttpServletRequest) request);
                            }
                            chain.doFilter(request, response);
                        } else {
                            //TODO add message
                            //request.removeAttribute("prettyContext");
                            httpResponse.sendError(httpStatus);
                        }
                    }
                } else {
                    log.info(String.format("KICK USER: sessionID - %s", session.getId()));
                    session.invalidate();
                    SecurityServlet.clearCookies(httpRequest, httpResponse);
                    request.setAttribute("loginError", "Ваша сессия была прервана Администратором системы.");
                    request.getRequestDispatcher("/login.jsp").forward(request, response);
                }
            } else {
                //log.info("### FILTER REDIRECT NO SESSION ");
                if(httpRequest.getDispatcherType() == DispatcherType.FORWARD) {
```

Аутентификация пользователя в системе

После выполнения метода **login**:

- если пользователь не авторизован, происходит авторизация пользователя по условиям, которые указаны в СТПО. При успешной авторизации, в cookie записывается логин пользователя, а в сессию модель пользователя (*UserModel*) которая получена с БД.
- если пользователь уже авторизован идет проверка жизни сессий текущего контекста.

```
private boolean login(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    String login = request.getParameter("login");
    String password = request.getParameter("password");
    if(login != null && password != null) {
        if(dataSource != null) {
            UserRoleDatabaseUtil util = null;
            UserModel userModel = null;
            try {
                PasswordToHash hash = new PasswordToHash();
                String hashedPassword = hash.passwordToHash(password);
                util = new UserRoleDatabaseUtil(dataSource.getConnection(), Lang.getDefault().name().toLowerCase());
                userModel = util.login(login, hashedPassword);
                util.close();
                if(userModel != null) {
                    if(!isActiveUser(userModel.getLogin())) {
                        if(userModel.getIP() != null && !userModel.getIP().equalsIgnoreCase(request.getRemoteAddr())) {
                            throw new AuthorizationException(AuthorizationException.Type.INVALID_IP_ADDRESS);
                        }
                    }
                    HttpSession session = request.getSession(true);
                    setUserProps(userModel, request);
                    loadUserRoles(userModel);
                    addCookie(request, response, USER_LOGIN_COOKIE, userModel.getLogin(), -1);
                    session.setAttribute(SecurityHandler.USER_SESSION_BEAN, new SessionBean(userModel));
                    session.setAttribute(IropDictionaryState.IROP_DIC_STATE, new IropDictionaryState());
                    session.setAttribute(PageHistory.PAGE_QUEUE, new PageHistory(10));
                    return true;
                } else throw new AuthorizationException(AuthorizationException.Type.USER_ALREADY_LOGGED_IN, userModel.getId());
            } else request.setAttribute("loginError", "Вы ввели неверные данные ('логин'/'пароль').");
        } catch (SQLException e) {
            e.printStackTrace();
            request.setAttribute("loginError", "Ошибка БД: " + e.getMessage());
        } catch (AuthorizationException e) {
            if(e.getType().getEventId() != null) {
                logEvent(e.getUserId(), new Event(e.getType().getEventId(), login), request.getRemoteAddr());
            }
        }

        if(e.getType() == AuthorizationException.Type.WRONG_PASSWORD) {
```

Возврат текущего авторизованного пользователя (*UserModel*)

```
@Named("securityManager")
@SessionScoped
public class SecurityManager implements Serializable {
    private SessionBean sessionBean;

    @PostConstruct
    public void init() {
        sessionBean = (SessionBean) FacesContext.getCurrentInstance().getExternalContext().getSessionMap().get(Security
    }

    public SessionBean getSessionBean() { return sessionBean; }

    public boolean hasRole(String role) { return sessionBean != null && sessionBean.getAllRoles().containsKey(role); }
```

Чтение модели *UserModel* (все свойства, в том числе роли и права)

- Получение текущего пользователя, в контроллер классах:

```
if(securityManager.getSessionBean() != null && securityManager.getSessionBean().getUser() != null)
```

- Получение и проверка текущего пользователя, в представлениях:

```
<c:choose>
  <c:when test="#{usersController.hasRole('CreateUsers'))">
    <p:commandLink id="addUser" title="#{msg['kz.bas.sop.administration.user.create.button']}" ajax="false" action="pretty:kz.b
      <i class="icon-plus" />
    </p:commandLink>
  </c:when>
  <c:otherwise>
    <p:commandLink id="addUser" title="#{msg['kz.bas.sop.administration.user.create.button']}" ajax="false" styleClass="btn btn
      <i class="icon-plus" />
    </p:commandLink>
  </c:otherwise>
</c:choose>
```

Виды ошибок

```
WRONG_LOGIN(1, "Попытка входа в систему с неправильным логином"),
WRONG_PASSWORD(2, "Попытка входа в систему с ошибочным паролем"),
USER_ALREADY_LOGGED_IN(3, "Попытка входа в систему под логином и паролем пользователя, который уже в системе"),
DELETED_USER(4, "Попытка входа в систему удаленного пользователя"),
BLOCKED_USER(null, "Попытка входа в систему заблокированного пользователя"),
INVALID_IP_ADDRESS(null, "Попытка входа в систему с рабочей станции с неверным IP-адресом");
```

Навигация в системе

Добавление нового адреса страницы

```
<url-mapping id="kz.bas.sop.administration.users.confirm" parentId="kz.bas.sop.administration.userrole">
  <pattern value="/confirm" />
  <view-id value="/faces/administration/user/confirm.xhtml" />
  <action inheritable="true">#{menuSelector.pushMenuItem('kz.bas.sop.administration.users.confirm','kz.
</url-mapping>
```

Переход по указанному адресу

```
<p:pretty:link mappingId="kz.bas.sop.administration.users.confirm">#{msg['kz.bas.sop.administration.users.confirm']}</p:pretty:link>
```

Логирование событий системы (EventLogger)

Вызов бина EventLogger:

```
@Inject
private EventLogger eventLogger;
```

Типы событий в системе

.....

Добавление событий по типу

Добавление событий осуществляется с помощью метода logEvent в классе EventLogger, который имеет несколько перегрузок.

Хлебные крошки (Breadcrumbs)

Компонент (org.primefaces.component.breadcrumb.BreadCrumb)

.....

Запись в историю переходов

Как говорили в начале, каждый запрос по адресу проходит через фильтр запросов. После всей проверки (успешной), адрес указанной страницы записывается. За записи историй переходов, отвечает **kz.bas.sop.beans.core.PageHistoryController**.

Вот так выглядит запись в историю в методе **doFilter (SecurityHandler)**:

```
processPageHistory( (HttpServletRequest) request );
```

```
private void processPageHistory(HttpServletRequest request) {  
    if(!isAjax(request) && request.getParameter(PageHistory.PAGE_NAVIGATION_REQUEST) == null) {  
        HttpSession session = request.getSession();  
        PageHistory pageHistory = (PageHistory)session.getAttribute(PageHistory.PAGE_QUEUE);  
        if(pageHistory != null) {  
            pageHistory.addHistory(getRequestURI(request));  
        }  
    }  
}
```

```
public void addHistory(String url) {  
    if(historyNavigation) {  
        historyNavigation = false;  
        return;  
    }  
  
    if(isException(url)) {  
        return;  
    }  
  
    if(history.size() > 0) {  
        if(currentIndex == history.size() - 1) {  
            if(history.size() == maxSize) {  
                history.removeFirst();  
                currentIndex--;  
            }  
        } else {  
            int last = history.size() - 1;  
  
            try {  
                while(last != currentIndex) {  
                    history.removeLast();  
                    last--;  
                }  
            } catch (NoSuchElementException e) {  
            }  
        }  
  
        if(url.equalsIgnoreCase(history.peekLast())) {  
            return;  
        }  
  
        currentIndex++;  
    }  
  
    history.offerLast(url);  
}
```


Проектные сервисы

Коллеги Вам предоставлен доступ к проектным сервисам
логины - басовский email без @bas.kz

Jira: <https://rnd.bas.kz:8444/jira/secure/Dashboard.jspa>

Stash: <https://rnd.bas.kz:8444/stash/projects>

Конфигурация Jboss

Коллеги, на нашем серваке (sop.bas.kz, 192.168.110.92) поднято 15 серверов приложений.
Настройка на них в приложенном файле.

Для доступа на рабочий стол данного сервера по RDP:

Host: 192.168.110.92

Login:Administrator

pwd:Zaq1xsw2

(Заходить по крайней необходимости)

Настройте свои IDE на использование своих серверов согласно вверенным Вам подсистемам.
На них сейчас задеплоены исходники без VVIS и PIKIP.

Структура серверов:

X.DEV - сервера для разработки для локальных приложений (смотрят на БД SOPDEV)

REM.DEV - внешний сервер VVIS для разработки (смотрят на БД SOPRDEV)

1.TEST - сервер для показа (смотрят на БД SOPTEST)

REM.TEST - внешний сервер VVIS для показа (смотрят на БД SOPRTEST)

Для всех серверов учетка на Системного администратора:

system_adm/123456

Также настроены БД.

1. Для каждого экземпляра созданы одинаковые пользователи

sys/manager

system/manager

dbsnmp/manager

sysman/manager

ADMIN_SOP/welcome1

2. Tnsnames

#Development Database

SOPDEV =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = sop.bas.kz)(PORT = 1522))

(CONNECT_DATA =

(SERVER = DEDICATED)

(SERVICE_NAME = SOPDEV)

)

)

#Development Remote VVIS Database

SOPRDEV =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = sop.bas.kz)(PORT = 1522))

(CONNECT_DATA =

```

    (SERVER = DEDICATED)
    (SERVICE_NAME = SOPRDEV)
  )
)

```

#Test Database

```

SOPTTEST =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = sop.bas.kz)(PORT = 1522))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = SOPTTEST)
    )
  )
)

```

#Test Remote VVIS Database

```

SOPRTEST =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = sop.bas.kz)(PORT = 1522))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = SOPRTEST)
    )
  )
)

```

3. На сервере ORACLE_HOME=C:\oracle\product\11.2.0

```

*****
*****
*****

```

DEV

- «Администрирование»

- «ИРОП»

<http://192.168.110.92:8081/sop/login.jsp>

Jrebel_port=12345

Server Folder:

C:\JBoss\SOP\1.DEV

Настройки сервера

```

<socket-binding-group name="standard-sockets" default-interface="public"
port-offset="${jboss.socket.binding.port-offset:0}">
  <socket-binding name="management-native" interface="management"
port="${jboss.management.native.port:7999}"/>
  <socket-binding name="management-http" interface="management"
port="${jboss.management.http.port:7990}"/>
  <socket-binding name="management-https" interface="management"
port="${jboss.management.https.port:7443}"/>
  <socket-binding name="ajp" port="7009"/>
  <socket-binding name="http" port="8081"/>
  <socket-binding name="https" port="7443"/>
  <socket-binding name="osgi-http" interface="management" port="7090"/>
  <socket-binding name="remoting" port="7447"/>
  <socket-binding name="txn-recovery-environment" port="7712"/>
  <socket-binding name="txn-status-manager" port="7713"/>
  <outbound-socket-binding name="mail-smtp">
    <remote-destination host="localhost" port="25"/>
  </outbound-socket-binding>
</socket-binding-group>

```

```
    </outbound-socket-binding>  
</socket-binding-group>
```

- «КУ»

http://192.168.110.92:8082/sop/login.jsp
Jrebel_port=12346

Server Folder:
C:\JBoss\SOP\2.DEV

Настройки сервера

```
<socket-binding-group name="standard-sockets" default-interface="public"
port-offset="${jboss.socket.binding.port-offset:0}">
  <socket-binding name="management-native" interface="management"
port="${jboss.management.native.port:8000}"/>
  <socket-binding name="management-http" interface="management"
port="${jboss.management.http.port:7991}"/>
  <socket-binding name="management-https" interface="management"
port="${jboss.management.https.port:7544}"/>
  <socket-binding name="ajp" port="7010"/>
  <socket-binding name="http" port="8082"/>
  <socket-binding name="https" port="7544"/>
  <socket-binding name="osgi-http" interface="management" port="7091"/>
  <socket-binding name="remoting" port="7448"/>
  <socket-binding name="txn-recovery-environment" port="7714"/>
  <socket-binding name="txn-status-manager" port="7715"/>
  <outbound-socket-binding name="mail-smtp">
    <remote-destination host="localhost" port="26"/>
  </outbound-socket-binding>
</socket-binding-group>
```

- «ПКП»

http://192.168.110.92:8083/sop/login.jsp
Jrebel_port=12347

Server Folder:
C:\JBoss\SOP\3.DEV

Настройки сервера

```
<socket-binding-group name="standard-sockets" default-interface="public"
port-offset="${jboss.socket.binding.port-offset:0}">
  <socket-binding name="management-native" interface="management"
port="${jboss.management.native.port:8001}"/>
  <socket-binding name="management-http" interface="management"
port="${jboss.management.http.port:7992}"/>
  <socket-binding name="management-https" interface="management"
port="${jboss.management.https.port:7545}"/>
  <socket-binding name="ajp" port="7011"/>
  <socket-binding name="http" port="8083"/>
  <socket-binding name="https" port="7545"/>
  <socket-binding name="osgi-http" interface="management" port="7092"/>
  <socket-binding name="remoting" port="7449"/>
  <socket-binding name="txn-recovery-environment" port="7716"/>
  <socket-binding name="txn-status-manager" port="7717"/>
  <outbound-socket-binding name="mail-smtp">
    <remote-destination host="localhost" port="27"/>
  </outbound-socket-binding>
</socket-binding-group>
```

- «БВИС» - local

http://192.168.110.92:8084/sop/login.jsp
Jrebel_port=12348

Server Folder:
C:\JBoss\SOP\4.DEV

Настройки сервера

```
<socket-binding-group name="standard-sockets" default-interface="public"
port-offset="${jboss.socket.binding.port-offset:0}">
  <socket-binding name="management-native" interface="management"
port="${jboss.management.native.port:8002}" />
  <socket-binding name="management-http" interface="management"
port="${jboss.management.http.port:7993}" />
  <socket-binding name="management-https" interface="management"
port="${jboss.management.https.port:7546}" />
  <socket-binding name="ajp" port="7012" />
  <socket-binding name="http" port="8084" />
  <socket-binding name="https" port="7546" />
  <socket-binding name="osgi-http" interface="management" port="7093" />
  <socket-binding name="remoting" port="7450" />
  <socket-binding name="txn-recovery-environment" port="7718" />
  <socket-binding name="txn-status-manager" port="7719" />
  <outbound-socket-binding name="mail-smtp">
    <remote-destination host="localhost" port="28" />
  </outbound-socket-binding>
</socket-binding-group>
```

- «БВИС» - remote

http://192.168.110.92:8085/sop/login.jsp
Jrebel_port=12349

Server Folder:
C:\JBoss\SOP\5.DEV

Настройки сервера

```
<socket-binding-group name="standard-sockets" default-interface="public"
port-offset="${jboss.socket.binding.port-offset:0}">
  <socket-binding name="management-native" interface="management"
port="${jboss.management.native.port:8003}" />
  <socket-binding name="management-http" interface="management"
port="${jboss.management.http.port:7994}" />
  <socket-binding name="management-https" interface="management"
port="${jboss.management.https.port:7547}" />
  <socket-binding name="ajp" port="7013" />
  <socket-binding name="http" port="8085" />
  <socket-binding name="https" port="7547" />
  <socket-binding name="osgi-http" interface="management" port="7094" />
  <socket-binding name="remoting" port="7451" />
  <socket-binding name="txn-recovery-environment" port="7720" />
  <socket-binding name="txn-status-manager" port="7721" />
  <outbound-socket-binding name="mail-smtp">
    <remote-destination host="localhost" port="29" />
  </outbound-socket-binding>
</socket-binding-group>
```

- «OK»

http://192.168.110.92:8086/sop/login.jsp
Jrebel_port=12350

Server Folder:
C:\JBoss\SOP\6.DEV
Настройки сервера

```
<socket-binding-group name="standard-sockets" default-interface="public"
port-offset="\${jboss.socket.binding.port-offset:0}">
  <socket-binding name="management-native" interface="management"
port="\${jboss.management.native.port:8004}"/>
  <socket-binding name="management-http" interface="management"
port="\${jboss.management.http.port:7995}"/>
  <socket-binding name="management-https" interface="management"
port="\${jboss.management.https.port:7548}"/>
  <socket-binding name="ajp" port="7012"/>
  <socket-binding name="http" port="8086"/>
  <socket-binding name="https" port="7548"/>
  <socket-binding name="osgi-http" interface="management" port="7095"/>
  <socket-binding name="remoting" port="7452"/>
  <socket-binding name="txn-recovery-environment" port="7722"/>
  <socket-binding name="txn-status-manager" port="7723"/>
  <outbound-socket-binding name="mail-smtp">
    <remote-destination host="localhost" port="30"/>
  </outbound-socket-binding>
</socket-binding-group>
```

- «ОРД»

http://192.168.110.92:8087/sop/login.jsp
Jrebel_port=12351

Server Folder:
C:\JBoss\SOP\7.DEV
Настройки сервера

```
<socket-binding-group name="standard-sockets" default-interface="public"
port-offset="\${jboss.socket.binding.port-offset:0}">
  <socket-binding name="management-native" interface="management"
port="\${jboss.management.native.port:8005}"/>
  <socket-binding name="management-http" interface="management"
port="\${jboss.management.http.port:7996}"/>
  <socket-binding name="management-https" interface="management"
port="\${jboss.management.https.port:7549}"/>
  <socket-binding name="ajp" port="7013"/>
  <socket-binding name="http" port="8087"/>
  <socket-binding name="https" port="7549"/>
  <socket-binding name="osgi-http" interface="management" port="7096"/>
  <socket-binding name="remoting" port="7453"/>
  <socket-binding name="txn-recovery-environment" port="7724"/>
  <socket-binding name="txn-status-manager" port="7725"/>
  <outbound-socket-binding name="mail-smtp">
    <remote-destination host="localhost" port="31"/>
  </outbound-socket-binding>
</socket-binding-group>
```

- «ОМ»

http://192.168.110.92:8088/sop/login.jsp
Jrebel_port=12352

Server Folder:
C:\JBoss\SOP\8.DEV
Настройки сервера

```
<socket-binding-group name="standard-sockets" default-interface="public"
port-offset="${jboss.socket.binding.port-offset:0}">
  <socket-binding name="management-native" interface="management"
port="${jboss.management.native.port:8006}"/>
  <socket-binding name="management-http" interface="management"
port="${jboss.management.http.port:7997}"/>
  <socket-binding name="management-https" interface="management"
port="${jboss.management.https.port:7550}"/>
  <socket-binding name="ajp" port="7014"/>
  <socket-binding name="http" port="8088"/>
  <socket-binding name="https" port="7550"/>
  <socket-binding name="osgi-http" interface="management" port="7097"/>
  <socket-binding name="remoting" port="7454"/>
  <socket-binding name="txn-recovery-environment" port="7726"/>
  <socket-binding name="txn-status-manager" port="7727"/>
  <outbound-socket-binding name="mail-smtp">
    <remote-destination host="localhost" port="32"/>
  </outbound-socket-binding>
</socket-binding-group>
```

- «ВЭД»

http://192.168.110.92:8089/sop/login.jsp
Jrebel_port=12353

Server Folder:
C:\JBoss\SOP\9.DEV
Настройки сервера

```
<socket-binding-group name="standard-sockets" default-interface="public"
port-offset="${jboss.socket.binding.port-offset:0}">
  <socket-binding name="management-native" interface="management"
port="${jboss.management.native.port:8007}"/>
  <socket-binding name="management-http" interface="management"
port="${jboss.management.http.port:7998}"/>
  <socket-binding name="management-https" interface="management"
port="${jboss.management.https.port:7551}"/>
  <socket-binding name="ajp" port="7015"/>
  <socket-binding name="http" port="8089"/>
  <socket-binding name="https" port="7551"/>
  <socket-binding name="osgi-http" interface="management" port="7098"/>
  <socket-binding name="remoting" port="7455"/>
  <socket-binding name="txn-recovery-environment" port="7728"/>
  <socket-binding name="txn-status-manager" port="7729"/>
  <outbound-socket-binding name="mail-smtp">
    <remote-destination host="localhost" port="33"/>
  </outbound-socket-binding>
</socket-binding-group>
```

- «OO»

http://192.168.110.92:8090/sop/login.jsp
Jrebel_port=12354

Server Folder:

C:\JBoss\SOP\10.DEV

Настройки сервера

```
<socket-binding-group name="standard-sockets" default-interface="public"
port-offset="${jboss.socket.binding.port-offset:0}">
  <socket-binding name="management-native" interface="management"
port="${jboss.management.native.port:8008}" />
  <socket-binding name="management-http" interface="management"
port="${jboss.management.http.port:8100}" />
  <socket-binding name="management-https" interface="management"
port="${jboss.management.https.port:7552}" />
  <socket-binding name="ajp" port="7016" />
  <socket-binding name="http" port="8090" />
  <socket-binding name="https" port="7552" />
  <socket-binding name="osgi-http" interface="management" port="7099" />
  <socket-binding name="remoting" port="7456" />
  <socket-binding name="txn-recovery-environment" port="7730" />
  <socket-binding name="txn-status-manager" port="7731" />
  <outbound-socket-binding name="mail-smtp">
    <remote-destination host="localhost" port="34" />
  </outbound-socket-binding>
</socket-binding-group>
```

- «ΦO»

http://192.168.110.92:8091/sop/login.jsp
Jrebel_port=12355

Server Folder:

C:\JBoss\SOP\11.DEV

Настройки сервера

```
<socket-binding-group name="standard-sockets" default-interface="public"
port-offset="${jboss.socket.binding.port-offset:0}">
  <socket-binding name="management-native" interface="management"
port="${jboss.management.native.port:8009}" />
  <socket-binding name="management-http" interface="management"
port="${jboss.management.http.port:8101}" />
  <socket-binding name="management-https" interface="management"
port="${jboss.management.https.port:7553}" />
  <socket-binding name="ajp" port="7017" />
  <socket-binding name="http" port="8091" />
  <socket-binding name="https" port="7553" />
  <socket-binding name="osgi-http" interface="management" port="7100" />
  <socket-binding name="remoting" port="7457" />
  <socket-binding name="txn-recovery-environment" port="7732" />
  <socket-binding name="txn-status-manager" port="7733" />
  <outbound-socket-binding name="mail-smtp">
    <remote-destination host="localhost" port="35" />
  </outbound-socket-binding>
</socket-binding-group>
```

- «ПФД»

http://192.168.110.92:8092/sop/login.jsp
Jrebel_port=12356

Server Folder:
C:\JBoss\SOP\12.DEV
Настройки сервера

```
<socket-binding-group name="standard-sockets" default-interface="public"
port-offset="${jboss.socket.binding.port-offset:0}">
  <socket-binding name="management-native" interface="management"
port="${jboss.management.native.port:8010}"/>
  <socket-binding name="management-http" interface="management"
port="${jboss.management.http.port:8102}"/>
  <socket-binding name="management-https" interface="management"
port="${jboss.management.https.port:7554}"/>
  <socket-binding name="ajp" port="7018"/>
  <socket-binding name="http" port="8092"/>
  <socket-binding name="https" port="7554"/>
  <socket-binding name="osgi-http" interface="management" port="7101"/>
  <socket-binding name="remoting" port="7458"/>
  <socket-binding name="txn-recovery-environment" port="7734"/>
  <socket-binding name="txn-status-manager" port="7735"/>
  <outbound-socket-binding name="mail-smtp">
    <remote-destination host="localhost" port="36"/>
  </outbound-socket-binding>
</socket-binding-group>
```

DEV REMOTE

- "БВИС" - remote

http://192.168.110.92:8093/sop/login.jsp
Jrebel_port=12357

Server Folder:
C:\JBoss\SOP\REM.DEV

Настройки сервера

```
<socket-binding-group name="standard-sockets" default-interface="public"
port-offset="${jboss.socket.binding.port-offset:0}">
  <socket-binding name="management-native" interface="management"
port="${jboss.management.native.port:8011}"/>
  <socket-binding name="management-http" interface="management"
port="${jboss.management.http.port:8103}"/>
  <socket-binding name="management-https" interface="management"
port="${jboss.management.https.port:7555}"/>
  <socket-binding name="ajp" port="7019"/>
  <socket-binding name="http" port="8093"/>
  <socket-binding name="https" port="7555"/>
  <socket-binding name="osgi-http" interface="management" port="7102"/>
  <socket-binding name="remoting" port="7459"/>
  <socket-binding name="txn-recovery-environment" port="7736"/>
  <socket-binding name="txn-status-manager" port="7737"/>
  <outbound-socket-binding name="mail-smtp">
    <remote-destination host="localhost" port="37"/>
  </outbound-socket-binding>
</socket-binding-group>
```

TEST LOCAL (for Demonstration)

http://192.168.110.92:8094/sop/login.jsp

Server Folder:

C:\JBoss\SOP\1.TEST

Настройки сервера

```
<socket-binding-group name="standard-sockets" default-interface="public"
port-offset="${jboss.socket.binding.port-offset:0}">
  <socket-binding name="management-native" interface="management"
port="${jboss.management.native.port:8012}"/>
  <socket-binding name="management-http" interface="management"
port="${jboss.management.http.port:8104}"/>
  <socket-binding name="management-https" interface="management"
port="${jboss.management.https.port:7556}"/>
  <socket-binding name="ajp" port="7020"/>
  <socket-binding name="http" port="8094"/>
  <socket-binding name="https" port="7556"/>
  <socket-binding name="osgi-http" interface="management" port="7103"/>
  <socket-binding name="remoting" port="7460"/>
  <socket-binding name="txn-recovery-environment" port="7738"/>
  <socket-binding name="txn-status-manager" port="7739"/>
  <outbound-socket-binding name="mail-smtp">
    <remote-destination host="localhost" port="38"/>
  </outbound-socket-binding>
</socket-binding-group>
```

TEST REMOTE (for Demonstration)

http://192.168.110.92:8095/sop/login.jsp

Server Folder:

C:\JBoss\SOP\REM.TEST

Настройки сервера

```
<socket-binding-group name="standard-sockets" default-interface="public"
port-offset="${jboss.socket.binding.port-offset:0}">
  <socket-binding name="management-native" interface="management"
port="${jboss.management.native.port:8013}"/>
  <socket-binding name="management-http" interface="management"
port="${jboss.management.http.port:8105}"/>
  <socket-binding name="management-https" interface="management"
port="${jboss.management.https.port:7557}"/>
  <socket-binding name="ajp" port="7021"/>
  <socket-binding name="http" port="8095"/>
  <socket-binding name="https" port="7557"/>
  <socket-binding name="osgi-http" interface="management" port="7104"/>
  <socket-binding name="remoting" port="7461"/>
  <socket-binding name="txn-recovery-environment" port="7740"/>
  <socket-binding name="txn-status-manager" port="7741"/>
  <outbound-socket-binding name="mail-smtp">
    <remote-destination host="localhost" port="39"/>
  </outbound-socket-binding>
```

```
</socket-binding-group>
*****1*****
```

Полезные функции:

1. Узнать из контроллера идентификатор настройки в pretty-config:

`PrettyContext.getCurrentInstance().getCurrentMapping().getId()`

Идентификаторы сущностей

Формат:

XXYYYYYYYY -

XX - код региональной базы

YYYYYYYY - сиквенс, дополненный слева нулями до 8-ми символов.