**УТВЕРЖДАЮ**

**…**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ …**

**«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.**

# РЕГЛАМЕНТ

разработки и внедрения программных продуктов

в рамках проекта АСУ «Экспресс» НП

**Общие положения**

Настоящий регламент разработан в целях унификации и стандартизации программных средств, интерфейсов и библиотек, используемых при создании программных модулей по проекту АСУ «Экспресс» НП.

Разработка должна осуществляться с использованием принятой архитектуры программного обеспечения, позволяющей упростить процессы взаимодействия между разработчиками и повторно использовать разработанные компоненты и шаблоны. При этом должен реализовываться подход разделения программно-аппаратной (внутренней) части системы (BackEnd) и клиентской стороны (FrontEnd), обеспечивающей реализацию пользовательского интерфейса.

Серверная инфраструктура АСУ «Экспресс» НП должна предусматривать использование отказоустойчивой архитектуры, состоящей из серверов баз данных и серверов приложений.

В соответствии с действующими рекомендациями Главного вычислительного центра (ГВЦ) ОАО «РЖД» в качестве основной системы управления базами данных, используемой разработчиками, должна выступать Postgres Pro Enterprise. В целях повышения производительности и устойчивости работы ПО, должно быть предусмотрено использование технологий кластеризации, партиционирования, шардирования и каскадной репликации данных.

Предлагаемая ГВЦ архитектура серверов приложений включает: веб-сервер и обратный прокси-сервер NGINX и 3 сервера WildFly. Рекомендуемые в настоящее время версии операционной системы и программного обеспечения серверов приведены в Приложении 1.

В целях повышения отказоустойчивости и надёжности система должна предусматривать выполнение репликации данных, а также автоматическое создание резервных копий как используемых баз данных, так и программного обеспечения серверов приложений. При этом должны быть предусмотрены механизмы мониторинга (рекомендации в Приложении 2) нагрузки и работоспособности серверов баз данных и приложений, включая метрики по отдельным модулям и приложениям.

Взаимодействие между модулями системы должно осуществляться через следующие механизмы:

* веб-сервисы на основе XML;
* веб-сервисы на основе RESTful;
* брокеров сообщений по схеме «издатель-подписчик».

Рекомендации по используемым протоколам и механизмам реализации обмена приведены в Приложении 3.

Управление версиями программного обеспечения системы должно осуществляться с использованием протоколов git, на базе внутреннего сервера gitlab. Использование внешних хранилищ исходных кодов не допускается.

В целях сохранения и развития созданных модулей системы основной платформой Backend разработок принимается Java Platform, Enterprise Edition, а спецификацией, используемой для построения веб-приложений принимается JavaServer Faces (JSF). Сборка проектов должна осуществляться в фреймворке автоматизации сборки Apache Maven. Версии программных средств разработки и развертывания приведены в Приложении 4. Версии допустимых к использованию в разработках основных библиотек и компонентов представлены в Приложении 5, дополнительных в Приложении 6.

Базовой концепцией построения модулей и приложений системы должная являться схема разделения данных приложения и управляющей логики на три компонента: модель, представление и контроллер - Model-View-Controller (MVC).

Для реализации представлений должны использоваться следующие технологии:

* HTML;
* CSS;
* JavaScript;
* CSS фреймворк Bootstrap;
* JavaScript фреймворк jQuery;
* Primefaces.

Версии используемых для представлений языков и технологий приведены в Приложении 7. Допустимые к использованию библиотеки и файлы представлений приведены в Приложении 8.

Для повышения скорости реализации в качестве перспективного фреймворка разработки приложений может быть использован Spring Framework. Версии допустимых основных и дополнительных модулей данного фреймворка приведены в Приложениях 9 и 10 соответственно.

При разработке представлений также допустимо использование основных и дополнительных перспективных библиотек и компонентов, приведенных в Приложениях 11 и 12 соответственно.

В качестве основной среды разработки модулей и приложений системы рекомендуется использование IDE Netbeans. Краткая инструкция по созданию проекта в данной среде приведена в Приложении 13.

В приложения и веб-сервисы должна встраиваться библиотека проверки прав доступа на доступность кода задачи и наличие прав на чтение и на редактирование. Дополнительные проверки агентов, перевозчиков, дорог, филиалов должны быть описаны в алгоритме работы приложения или веб-сервиса. Описание работы с библиотеками проверки прав доступа для приложений приведено в Приложении 14, для веб-сервисов – в Приложении 15.

При внесении (редактировании) БД приложение или веб-сервис должно вызывать библиотеку логирования внесения изменений в БД. Описание работы с библиотекой логирования приведено в Приложении 16.

В веб-приложении данные о номере версии, технологе и контактах должны формироваться посредством подключения библиотеки информации о приложении. Описание работы с библиотекой приведено в Приложении 17

Для получения данных при наполнении элементов входных параметров (меню) должен использоваться банк микросервисов. Описание работы с банком микросервисов приведено в Приложении 18.

При переводе валют должна использоваться функция для получения курса валют. Описание работы с функцией приведено в Приложении 19.

# Приложение 1 Рекомендации по версиям серверного программного обеспечения

1. Используемая операционная система CentOS 7.5, 7.6, 7.9

Перспективная операционная система РЕД ОС.

Рекомендуемый дистрибутив: redos-MUROM-7.3-20210412-Everything-x86\_64-DVD1.iso

1. Сервер приложений WildFly (версии 14-21)
2. Веб-сервер NGINX версии не ниже \_\_\_ (требования будут уточнены позже)
3. Система управления базами данных PostgreSQL 12.5 (в ЦДСПП PostgreSQL 11.6)

Перспективная система управления базами данных Postgres Pro Enterprise версии \_\_\_\_

# Приложение 2 Рекомендации по средствам мониторинга серверов и приложений

(требования будут уточнены позже)

1. Агенты мониторинга Tivoli версии \_\_\_
2. Система мониторинга Prometheus версии \_\_\_
3. Отображение данных мониторинга с использованием приложения Grafana, версии \_\_\_

# Приложение 3 Рекомендации по протоколам и средствам обмена сообщениями между компонентами и модулями системы

1. JAX-RS, JavaTM API for XML-Based Web Services (JAX-WS) версии 2.0
2. Apache Kafka версии \_\_\_ (требования будут уточнены позже)
3. ActiveMQ Artemis версии \_\_\_ (требования будут уточнены позже)

# Приложение 4 Рекомендации по версиям используемых средств разработки и сборки

1. Java Platform, Enterprise Edition, версии 8
2. JavaServer Faces (JSF) версии не ниже 2.0
3. Сборщик проектов Maven (3.6.3-3.8.х)

# Приложение 5 Допустимые к использованию при разработке приложений основные библиотеки

Библиотеки, интегрированные в WildFly

|  |  |  |
| --- | --- | --- |
| **Название библиотеки** | **версия** | **описание** |
| commons-compress | 1.19 |  |
| commons-collections4 | 4.4, 4.3 |  |
| commons-lang3 | 3.9 |  |
| commons-net | 3.6 |  |
| itext | 2.1.7 | Формирование файла rtf, pdf |
| primefaces | 6.2, 7.0, 8.0 | Дополнительная библиотека с расширенным функционалом для jsf |
| poi | 3.11, 4.1.2 (4.1.0, 5.0.0) | Формирование файла excel |
| poi-ooxml-schemas | 3.11, 4.1.2 | Формирование файла excel |
| poi-ooxml | 3.11, 4.1.2 | Формирование файла excel |
| PBKDF2 | 1.1.0 | Шифрование |
| postgresql | 42.2.18 | Подключение к базе |
| xmlbeans | 2.6.0, 3.1.0 | Формирование файла excel |

Библиотеки, требующие отдельной установки на сервер WildFly

|  |  |  |
| --- | --- | --- |
| **Название библиотеки** | **версия** | **описание** |
| maven-compiler-plugin | 3.1 | Сборщик maven |
| maven-war-plugin | 2.3 | Сборщик maven |
| maven-dependency-plugin | 2.6 | Сборщик maven |
| maven-install-plugin | 2.5.2 | Сохранение в локальном репозитории своих библиотек |
| Omnifaces | 1.7 |  |
| Javax.mail | 1.5.0 | это Java API предназначенное для получения и отправки электронной почты с использованием протоколов SMTP, POP3 и IMAP |
| Javaee-api | 7.0 |  |
| FastInfoset | 1.2.15, 1.13 | это хранение XML данных в бинарном файле |
| Commons-codec | 1.2.15, 1.15 | Кодирование/декодирование данных |
| Commons-math3 | 3.6.1 | Математические методы |
| Commons-logging | 1.2 | это утилита ведения журнала на основе Java и модель программирования для ведения журнала и других наборов инструментов. Он предоставляет API, реализации журналов и реализации оболочек поверх некоторых других инструментов |
| Hibernate-jpamodelgen | 5.4.4.Final, 5.4.12.Final | Использование jpa для связи с БД |
| Istack-commons-runtime | 3.0.7 |  |
| Javax.activatin-api | 1.2.0 | The JavaBeans(TM) Activation Framework is used by the JavaMail(TM) API to manage MIME data. |
| Jaxb-api | 2.3.1 | Java API для маршалинга объекта в XML и восстановления объекта из XML файла. Изначально JAXB был отдельным проектом, но своей простотой и удобством быстро завоевал популярность Java разработчиков. |
| Jaxb-runtime | 2.3.1 | Java Architecture for XML Binding. Специальный инструмент для маршалинга и анмаршалинга объектов. Часто используется в веб сервисах для представления объекта в виде XML схемы и передачи ее по сети |
| Jboss-logging | 3.3.2 | Ведение журнала логирования в Jboss |
| Jcl-over-slf4j | 1.7.30 | SLF4J (Simple Logging Facade for Java) — библиотека для протоколирования, ставящая своей целью предоставить максимально простой, но при этом мощный фасад для различных систем протоколирования на Java |
| slf4j-api | 1.7.30 | SLF4J (Simple Logging Facade for Java) — библиотека для протоколирования, ставящая своей целью предоставить максимально простой, но при этом мощный фасад для различных систем протоколирования на Java |
| SparseBitSet | 1.2 | An efficient sparse bitset implementation for Java |
| Stax-ex | 1.8 |  |
| Txw2 | 2.3.1 | TXW is a library that allows you to write XML documents. |

# Приложение 6 Допустимые к использованию при разработке приложений дополнительные библиотеки

|  |  |  |
| --- | --- | --- |
| **Название библиотеки** | **версия** | **описание** |
| activation | 1.1 | это стандартное расширение платформы Java, которое позволяет вам использовать преимущества стандартных сервисов для: определения типа произвольного фрагмента данных; инкапсуляции доступа к нему; обнаружения операций, доступных для него; и создания экземпляра соответствующего компонента для выполнения операции |
| Curvesapi | 1.04 | Реализация различных математических кривых, которые определяют себя по набору контрольных точек. API написан на Java. Поддерживаются следующие кривые: Безье, B-сплайн, Кардинальный сплайн, сплайн Catmull-Rom, Лагранж, Натуральный кубический сплайн и NURBS. |
| Jackson-annotations | 2.8.0, 2.9.0, 2.13.1 |  |
| Jackson-core | 2.8.7, 2.9.7, 2.9.9, 2.13.1 |  |
| Jackson-databind | 2.9.7, 2.9.9, 2.13.1 |  |
| Javax.json | 1.1.2 |  |
| Javax.json-api | 1.0 |  |
| Javax.json.bind-api | 1.0 |  |
| Json | 20180813 |  |
| Log4j | 1.2.12 | Логирование хода выполнения программы в отдельный файл |
| Resteasy-json-binding-provider | 4.0.0.Final |  |
| Yasson | 1.0.3 |  |

# Приложение 7 Рекомендации по языкам, технологиям разработки представлений

1. HTML версии 5

2. CSS версии не ниже 3

3. JavaScript версии не ниже 1.7

4. фреймворк Bootstrap, версии 5

5. JavaScript фреймворк jQuery версии 3.х

6. Primefaces, версии не ниже 6.2

# Приложение 8 Допустимые к использованию библиотеки и файлы представлений

js-файлы:

1. Bootstrap версии
2. jQueryUI версии
3. jQueryTable версии
4. popper.min.js версии
5. bootstrap.min.js версии

css-файлы:

1. Bootstrap версии
2. font-awesome.min.css версии
3. feather.css версии

# Приложение 9 Допустимые к использованию основные модули фреймворка Spring

|  |  |  |
| --- | --- | --- |
| **Название модуля** | **Версия** | **Описание** |
| Spring Boot | 2.7.1 | Предназначен для упрощения создания приложений на основе Spring. Позволяет наиболее простым способом создать web-приложение, требуя от разработчиков минимум усилий по его настройке и написанию кода.   * Создание полноценных Spring приложений * Встроенный Tomcat или Jetty (не требуется установки WAR файлов) * Обеспечивает 'начальные' POMs для упрощения вашей Maven конфигурации * Автоматическая конфигурирация Spring когда это возможно * Обеспечивает такими возможностями, как метрики, мониторинг состояниями и расширенная конфигурация * Абсолютно без генерации кода и без написания XML конфигурации |
| Spring Framework | 5.3.22 | фреймворк предоставляет комплексную модель программирования и конфигурации для современных корпоративных приложений на основе Java на любой платформе развертывания.  Ключевым элементом Spring является инфраструктурная поддержка на уровне приложений: Spring фокусируется на «подключении» корпоративных приложений, чтобы команды могли сосредоточиться на бизнес-логике на уровне приложений без ненужных привязок к конкретным средам развертывания   * Внедрение зависимости * Аспектно-ориентированное программирование, включая декларативное управление транзакциями * Создание Spring MVC web-приложений и RESTful web-сервисов * Начальная поддержка JDBC, JPA, JMS |

# Приложение 10 Допустимые к использованию дополнительные модули фреймворка Spring

|  |  |  |
| --- | --- | --- |
| **Название модуля** | **Версия** | **Описание** |
| Spring Data | 2.7.1 | дополнительный удобный механизм для взаимодействия с сущностями базы данных, организации их в репозитории, извлечение данных, изменение, в каких то случаях для этого будет достаточно объявить интерфейс и метод в нем, без имплементации |
| Spring Web Services | 3.1.1 | продукт Spring-сообщества, основной целью которого является создание документоориентированных web-сервисов. Spring Web Services нацелен на облегчение разработки SOAP-сервиса методом [Сontract-Аirst](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%BB%D1%83%D0%B6%D0%B1%D0%B0#.D0.9C.D0.B5.D1.82.D0.BE.D0.B4.D1.8B_.D1.80.D0.B0.D0.B7.D1.80.D0.B0.D0.B1.D0.BE.D1.82.D0.BA.D0.B8), позволяя создавать легко изменяемые web-сервисы путем манипулирования XML-настройками. Продукт основан на Spring, чтопозволяет вам изпользовать такие возможности, как DI, и это будет неотъемлемой частью вашего web-сервиса.  Использовать Spring-WS можно в разных целях, но в основном потому, что находят в других SOAP-инструментах недостатки, когда дело доходит до "best practices" при реализации SOAP-сервиса. Spring-WS делает разработку "best practices" более легким. Он включает такие методологии, как [WS-I basic profile](https://ws-i.org/), [Сontract-Аirst](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%BB%D1%83%D0%B6%D0%B1%D0%B0#.D0.9C.D0.B5.D1.82.D0.BE.D0.B4.D1.8B_.D1.80.D0.B0.D0.B7.D1.80.D0.B0.D0.B1.D0.BE.D1.82.D0.BA.D0.B8) и имеет слабую связь между "договором" и реализацией. |
| Spring Security | 5.7.2 | фреймворк, который сфокусирован на обеспечение как аутентификации, так и авторизации в Java-приложениях. Как и все Spring проекты, настоящая сила Spring Security в том, что он может быть легко дополнен нужным функционалом.   * Комплексная и расширяемая поддержка как аутентификации, так и авторизации * Защита от атак типа фиксация сессии, [кликджекинг](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%BA%D0%B4%D0%B6%D0%B5%D0%BA%D0%B8%D0%BD%D0%B3), [межсайтовая подделка запроса](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B6%D1%81%D0%B0%D0%B9%D1%82%D0%BE%D0%B2%D0%B0%D1%8F_%D0%BF%D0%BE%D0%B4%D0%B4%D0%B5%D0%BB%D0%BA%D0%B0_%D0%B7%D0%B0%D0%BF%D1%80%D0%BE%D1%81%D0%B0) и др. * Интеграция с Servlet API * Интеграция с Spring Web MVC при необходимости |
| Spring AMQP | 2.4.6 | Проект состоит из двух частей: spring-amqp как базовая абстракция и spring-rabbit как реализация [RabbitMQ](https://www.rabbitmq.com/).   * Listener container для асинхронной обработки входящих сообщений * RabbitTemplate для отправки и получения сообщений * RabbitAdmin для автоматического описания запросов, обмена и связывания |
| Spring Integration | 5.5.12 | Этот фреймворк расширяет модель программирования в Spring, благодаря поддержке [Enterprise Integration Patterns](https://www.eaipatterns.com/). Spring Integration включает в себя легковесный обмен сообщениями между Spring-приложениями через событийно-управляемые адаптеры. Эти адаптеры обеспечивают высокоуровневую абстракцию над возможностями Spring, такими как удаленное взаимодействие, обмен сообщениями и выполнение по расписанию. Первичной целью Spring Integration является предоставление простой модели создания интеграционных бизнес-решений, сохраняя разделение задач, что необходимо для написания поддерживаемого и тестируемого кода. |
| Spring Web Flow | 2.5.1 | фреймворк основан на Spring MVC и позволяет реализовать "потоки" web-приложений. Поток представляет собой последовательность шагов, которые необходимо совешить для выполнения поставленной бизнес-задачи. Он состоит из множества НТТР-запросов, имеющими состояния, оперирующие данными по транзакции, повторно используемые, которые могут быть как короткосрочными, так и долгоживущими. |
| Spring Batch | 4.3.6 | фреймворк предоставляет повторно используемые функции, которые необходимы в процессе обработки большого числа записей, включая логгирование/трассировку, управление транзакциями, обработка статистики, перезапуск и пропуск задач и управление ресурсами. Он также предоставляет продвинутые сервисы и возможности, которые включают в себя высокопроизводительные задачи обработки данных чрезвычайно большого объема через техники оптимизации и разбиения. Как простые, так и сложные объемные пакетные задачи могут использовать фреймворк, который использует хорошо масштабируемый способ обработки значительных объемов информации. |

# Приложение 11 Допустимые к использованию основные перспективные библиотеки и компоненты разработки представлений

|  |  |  |
| --- | --- | --- |
| **Название модуля** | **Версия** | **Описание** |
| React | actual | JavaScript-библиотека для создания пользовательских интерфейсов |
| Mobx | actual | автономная JavaScript библиотека, для управления фронтенд-состоянием приложения. Обеспечивает консистентность и согласованность внутреннего состояния фронтенд-приложения, предоставляя удобные инструменты для его изменения. |
| Material UI | actual | это набор компонентов React, который реализует Google Material Design (material-ui недавно выпустили v1 библиотеки). Эти компоненты работают изолированно, это означает, что они являются само-поддерживающими и вводят только те стили, которые они должны отображать |
| React Bootstrap | actual | это библиотека React-компонентов, подходящих для повторного использования, которая реализует возможности популярного шаблона Bootstrap от Twitter. |

# Приложение 12 Допустимые к использованию дополнительные перспективные библиотеки и компоненты разработки представлений

|  |  |  |
| --- | --- | --- |
| **Название модуля** | **Версия** | **Описание** |
| Jest | 28.1 | это фреймворк для тестирования JavaScript, разработанный для обеспечения уверенности в правильной работе любого JavaScript кода. Он позволяет вам писать тесты с приемлемым, знакомым и функциональным API, и быстро достигать желаемых результатов. |
| Typescript |  | язык программирования для веб-разработки, основанный на JavaScript |

# Приложение 13 Инструкция по созданию проекта в среде Netbeans

1. Подключить явным образом maven в IDE Netbeans:
   1. Открыть IDE Netbeans
   2. Перейти в меню IDE Сервис->Параметры->закладка maven
   3. Выбрать пункт домашняя страница maven
   4. Указать явным образом компановщик maven
2. Создать в IDE проект:

Для версии IDE 8.2:

* 1. Maven -> Веб приложение (Шаблон Maven проекта веб-приложения, созданный при помощи плагина Maven Archetype)
  2. Уточнение выбрать сервер wildfly с версией Java EE – Java EE 7 Web
  3. Выбрать собственный проект -> Новый -> Другое… -> Веб-службы -> Веб-службы RESTful на основе шаблонов (Создает веб-службы RESTful с помощью одного из следующих шаблонов: единичный, контейнер-элемент, контейнер-элемент с клиентским управлением) -> Простой корневой ресурс

Для версии IDE с 12:

* 1. Maven -> Веб приложение (Шаблон Maven проекта веб-приложения, созданный при помощи плагина Maven Archetype)
  2. Уточнение выбрать сервер wildfly с версией Java EE – Java EE 8 Web

# Приложение 14 Описание работы с библиотекой проверки прав доступа, встраиваемой в веб-приложения (usercheck)

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

emailMap – возвращает список emailов по переданному в программу коду задачи

HashMap getParamHashMap(String param) throws Exception

Пример получения списка с почтой:

HashMap emailM = getParamHashMap("emailMap");

# Приложение 15 Описание работы с библиотекой проверки прав доступа, встраиваемой в веб-сервисы (fcheck)

**Варианты функций:**

1. HashMap check(String login, String task, String stattask, String ip) throws Exception;
2. HashMap checkpass(String login, String pass, String task, String stattask, String ip) throws Exception;
3. HashMap checklight(String login, String task, String stattask, String ip) throws Exception;
4. HashMap checkpasslight(String login, String pass, String task, String stattask, String ip) throws Exception;
5. boolean SaveTimeExit(Integer id,Integer sizef) throws Exception;
6. boolean SaveTimeExitFile(Integer userid, Integer sizef, String namefile, String typefile) throws Exception;
7. boolean SaveTimeExitLoad(Integer userid, Integer sizef, String namefile, String typefile) throws Exception;

**Подключение FCheck**

ejbUserCheck = (FCheckRemote) ic.lookup("java:global/FCheck-1.0/FCheck!com.vniizht.fcheck.FCheckRemote");

Используется команда:

3. HashMap checklight(String login, String task, String stattask, String ip)

Пример использования:

HashMap nh = ejbUserCheck.checklight(httpReq.getRemoteUser(),"TBR", "TBR", httpReq.getRemoteAddr());

HashMap содержит:

**"result"** – результат запроса true успешно прошел, false вход не успешный, причина неуспешного входа содержаться в "message". Если от fcheck пришел false, то сервис должен вернуть сообщение об ошибке, тому кто его вызвал. В каком виде и какие ошибки, это должно обговариваться с технологами по задаче.

**result = (boolean)nh.get("result");**

**"message"** – сообщение об успешном входе, или содержит причину неуспешного входа

**message = (String)nh.get("message");**

**"userid"** – если вход успешный, содержит id из таблицы PUD.OARM, иначе 0

**userid = (Integer)nh.get("userid");**

**"armgroup"** – содержит информацию о пользователе

**armgroup = (HashMap)nh.get("armgroup");**

Коды для получения информации о пользователе, следующие:

private final static String[] armgroup = {"LOCK","KOD\_ASUPV","GROUPCODE", "OWNER", "OWNERPLASE", "SKP", "AGENT", "KOD\_PODR", "FAMILIYA", "IMYA", "OTCHESTVO", "DOLZHN", "EMAIL", "PHONE", "FILIAL", "SUBAGN", "KODELS","MSG"};

Пример получение информации:

int skp = (Integer.valueOf((Short) armgroupRow.get("SKP"))).intValue();

int agent = (Integer.valueOf((Short) armgroupRow.get("AGENT"))).intValue();

String groupFil = String.valueOf((Character) armgroupRow.get("FILIAL"));

int groupSuba = Integer.valueOf((Short) armgroupRow.get("SUBAGN"));

String groupDor = String.valueOf((Character) armgroupRow.get("OWNER"));

if (groupDor.equals("@") || groupDor.equals("\*")) {

}

Список сообщений **"message"**:

|  |  |  |
| --- | --- | --- |
| № | message | kod |
| 1 | Необходимо задать имя рабочей группы (пользователя)! |  |
| 2 | Пользователь заблокирован! |  |
| 3 | Пользователь не зарегистрирован! |  |
| 4 | Необходимо задать название задачи! |  |
| 5 | Задача ‘task’ заблокирована! |  |
| 6 | Задача с кодом ‘task’ не зарегистрирована! |  |
| 7 | У пользователя ‘login’ отсутствует разрешение запускать приложения с ip-адреса ‘ip’! |  |
| 8 | У пользователя ‘login’ отсутствует разрешение на выполнение задачи с кодом ‘task’! |  |
| 9 | Укажите пароль! |  |
| 10 | Неверный пароль! |  |
| 11 | Ошибка добавления записи в PUD.OARM: |  |
| 12 | Доступ пользователя ‘login’ к задаче с кодом ‘task’ разрешен. | ​ |
| 13 | FCheck: Пустой IP-адрес! |  |
| 14 | FCheck: Ошибочный формат IP-адреса! |  |

**Передача данных в OARM для ведения журналов аудита.**

**При успешном завершении нужно вызвать функцию (устарело)**

5.boolean SaveTimeExit(Integer id,Integer sizef) throws Exception;

Веб-сервис должен передавать следующие данные в OARM:

1. userid – id из таблицы OARM полученное из fcheck
2. sizef – объем передаваемых данных в байтах

boolean result = SaveTimeExit(Userid, sizef);

Результат: true – успешная запись данных и сохранение времени работы в OARM

false – ошибка при попытке сохранить данные

**Если происходила выгрузка файла, то нужно использовать функцию (новое):**

6.boolean SaveTimeExitFile(Integer userid, Integer sizef, String namefile, String typefile) throws Exception;

Веб-сервис должен передавать следующие данные в OARM:

1. userid – id из таблицы OARM полученное из fcheck
2. sizef – объем передаваемых данных в байтах
3. namefile – название файла (не более 50 символов)
4. typefile – тип файла (не более 5 символов)

boolean result = SaveTimeExit(Userid, sizef);

Результат: true – успешная запись данных и сохранение времени работы в OARM

false – ошибка при попытке сохранить данные

**Если происходила выгрузка файла, то нужно использовать функцию (новое):**

7.boolean SaveTimeExitFile(Integer userid, Integer sizef, String namefile, String typefile) throws Exception;

Веб-сервис должен передавать следующие данные в OARM:

1. userid – id из таблицы OARM полученное из fcheck
2. sizef – объем передаваемых данных в байтах
3. namefile – название файла (не более 50 символов)
4. typefile – тип файла (не более 5 символов)

boolean result = SaveTimeExit(Userid, sizef);

Результат: true – успешная запись данных и сохранение времени работы в OARM

false – ошибка при попытке сохранить данные

При использовании команды

1.HashMap check(String login, String task, String stattask, String ip);

Помимо вышеуказанных переменных. Можно получить список входящих в код программ:

**"taskMap"** – содержит список кодов задач, входящих в данную задачу, на которые имеет право пользователь:

**taskMap = (HashMap)nh.get("taskMap");**

**Если происходила загрузка файла, то нужно использовать функцию (новое):**

8.boolean SaveTimeExitLoad(Integer userid, Integer sizef, String namefile, String typefile) throws Exception;

Веб-сервис должен передавать следующие данные в OARM:

1. userid – id из таблицы OARM полученное из fcheck
2. sizef – объем закаченных данных в байтах
3. namefile – название файла (не более 50 символов)
4. typefile – тип файла (не более 5 символов)

boolean result = SaveTimeExit(Userid, sizef);

Результат: true – успешная запись данных и сохранение времени работы в OARM

false – ошибка при попытке сохранить данные

При использовании команды

1.HashMap check(String login, String task, String stattask, String ip);

Помимо вышеуказанных переменных. Можно получить список входящих в код программ:

**"taskMap"** – содержит список кодов задач, входящих в данную задачу, на которые имеет право пользователь:

**taskMap = (HashMap)nh.get("taskMap");**

# Приложение 16 Описание работы с библиотекой логирования (nglog)

**Варианты функций:**

1. HashMap log(String e\_class, String e\_text, String s\_ip, String system\_n, String proc\_n, Integer e\_code, Integer parent\_id) throws Exception;
2. HashMap logHashMap(HashMap list) throws Exception;
3. HashMap logprilHashMap(HashMap list) throws Exception;
4. HashMap log(String e\_class, String e\_text, String s\_ip, String system\_n, String proc\_n, Integer e\_code, Integer parent\_id, BigInteger proc\_id) throws Exception;
5. HashMap startProcessesHashMap(HashMap list) throws Exception;
6. HashMap endProcessesHashMap(HashMap list) throws Exception;
7. HashMap event(HashMap list) throws Exception;
8. HashMap eventlog(HashMap list) throws Exception;

**Подключение NGlog**

В pom.xml добавляется

<dependency>

<groupId>NGlog</groupId>

<artifactId>nglog</artifactId>

<version>1.0</version>

<type>jar</type>

<classifier>cient</classifier>

</dependency>

На зависимость ставится nglog-1.0-client.jar

ejbLog = (NGlogRemote) ic.lookup("java:global/nglog-1.0/NGlog!org.vniizht.nglog.NGlogRemote");

1. **Запись в таблицу nglog.log команда:**

HashMap log(String e\_class, String e\_text, String s\_ip, String system\_n, String proc\_n, Integer e\_code, Integer parent\_id);

Пример использования:

HashMap nglog = ejbLog. log("Е", "Ошибка при записи в таблицу", "172.10.10.10", "servicecheck", "passworddatak", 0, 0);

HashMap содержит:

**"result"** – результат запроса true успешная запись в лог, false - не успешная запись в лог, причина неуспешной записи содержаться в "message".

**result = (boolean)** **nglog.get("result");**

**"message"** – сообщение об успешном входе, или содержит причину неуспешного входа

**message = (String)** **nglog.get("message");**

**"id"** – если вход успешный, содержит id из таблицы nglog.log, иначе 0

**id = (Integer)** **nglog.get("id");**

Список сообщений **"message"**:

|  |  |  |
| --- | --- | --- |
| № | message | kod |
| 1 | Ошибка добавления записи в nglog.log: |  |

1. **Запись в таблицу nglog.log (выборочные поля) команда**

HashMap logHashMap(HashMap list) ;

private final static String[] logtable = {"E\_CLASS","PARENT\_ID","S\_IP", "C\_IP", "USER\_ID",

"SYSTEM\_N", "PROC\_N", "E\_CODE", "E\_TEXT", "TO\_MON", "E\_PRIM", "PROC\_ID"};

HashMap tab = new HashMap();

tab.put(logtable[0], "Е");

tab.put(logtable[8], "Ошибка при записи в таблицу");

tab.put(logtable[5], "servicecheck");

tab.put(logtable[6], "passworddatak ");

tab.put(logtable[2], request.getRemoteAddr());

tab.put(logtable[11], **proc\_id**); //в 5 пункте описывается его получение

nglog = ejbLog.logHashMap(tab);

HashMap содержит: "result", "message", "id".

1. **Запись в таблицу nglog.logpril команда:**

HashMap logprilHashMap(HashMap list) ;

private final static String[] logpriltable = {"zadcode", "result", "kodotv"};

private final static String[] priltyped = {"String", "Integer", "Integer"};

HashMap tab = new HashMap();

tab.put(logpriltable [0], "NSI");

tab.put(logpriltable [1], 1);

tab.put(logpriltable [2], 200);

nglog = ejbLog.logprilHashMap(tab);

HashMap содержит: "result", "message", "id".

1. **Запись в таблицу nglog.log команда:**

HashMap log(String e\_class, String e\_text, String s\_ip, String system\_n, String proc\_n, Integer e\_code, Integer parent\_id, BigInteger proc\_id);

Пример использования:

HashMap nglog = ejbLog.log("Е", "Ошибка при записи в таблицу", "172.10.10.10", "servicecheck", "passworddatak", 0, 0,1223);

HashMap содержит: "result", "message", "id".

1. **Запись в таблицу nglog.processes – создание процесса команда:**

HashMap startProcessesHashMap(HashMap list);

private final static String[] processestable = {"system\_n", "proc\_n", "s\_ip", "statusr"};

private final static String[] processestyped = {"String", "String", "String", "String"};

HashMap tab = new HashMap();

tab.put(processestable [0], "servicecheck");

tab.put(processestable [1], "passworddatak");

tab.put(processestable [2], "172.10.10.10");

tab.put(processestable [3], "R");

nglog = ejbLog. startProcessesHashMap(tab);

HashMap содержит: "result", "message", "proc\_id".

**proc\_id** содержит номер процесса его нужно сохранить, и использовать при записи данных в таблицу nglog.log, так же при окончании процесса его нужно передать в функцию endProcessesHashMap

**BigInteger proc\_id = (BigInteger)** **nglog.get("proc\_id");**

1. **Изменение таблицы nglog.processes – закрытие процесса и подсчет времени работы команда:**

HashMap endProcessesHashMap(HashMap list);

HashMap tab = new HashMap();

tab.put(processestable [0], "servicecheck");

tab.put(processestable [1], "passworddatak");

tab.put(processestable [2], "172.10.10.10");

Если процедура завершилась успешно, то передаем в статусе О(лат)

tab.put(processestable [3], "O");

tab.put("proc\_id", proc\_id);

Если процедура завершилась не успешно, то передаем E(лат)

tab.put(processestable [3], "E");

tab.put("proc\_id", proc\_id);

nglog = ejbLog. endProcessesHashMap(tab);

HashMap содержит: "result", "message".

1. **Запись в таблицу nglog.event (выборочные поля) команда**

HashMap event(HashMap list);

private final static String[] eventtable = {"shema", "tablename", "maxid", "username", "userip", "datver", "nver", "datform"};

private final static String[] eventtyped = {"String", "String", "BigInteger", "String", "String", "Date", "String", "Date"};

HashMap tab = new HashMap();

tab.put(eventtable[0], "unp");

tab.put(eventtable[1], "train");

tab.put(eventtable[2], 10);

tab.put(eventtable[3], request.getRemoteUser());

tab.put(eventtable[4], request.getRemoteAddr());

tab.put(eventtable[6], "1");

nglog = ejbLog.event(tab);

HashMap содержит: "result", "message", "id".

1. **Запись в таблицу nglog.eventlog (выборочные поля) команда**

HashMap eventlog(HashMap list);

private final static String[] logeventtable = {"username", "userip", "schemaname", "tablename", "eventoper", "zadcode", "idz", "note", "vid"};

private final static String[] logeventtyped = {"String", "String", "String", "String", "String", "String", "Integer", "String", "Integer"};

HashMap tab = new HashMap();

tab.put(logeventtable[0], request.getRemoteUser());

tab.put(logeventtable[1], request.getRemoteAddr());

tab.put(logeventtable[2], "unp");

tab.put(logeventtable[3], "train");

tab.put(logeventtable[4], "I");

tab.put(logeventtable[5], "unp\_mrt");

tab.put(logeventtable[6], 10);

tab.put(logeventtable[8], 10);

nglog = ejbLog.eventlog(tab);

HashMap содержит: "result", "message", "id".

# Приложение 17 Описание работы с библиотекой информации о приложении (prilinfo)

1. Состоит из двух библиотек prilinfo-1.0.jar, prilinfo-1.0-client.jar
2. prilinfo-1.0.jar деплоится на WildFly. На сервере приложений должен быть JNDI java/CheckDS
3. prilinfo-1.0-client.jar ставится в приложение
   1. Копируется сама библиотека в папку ../имяприложения/имя приложения-web/lib
   2. В pom.xml добавляется зависимость

<dependency>

<groupId>prilinfo</groupId>

<artifactId>prilinfo</artifactId>

<version>1.0</version>

<type>jar</type>

</dependency>

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

1. Пример заполнения нужных полей :

public class GosForms implements Serializable {

private String tmessage = "";

private String tzadname = ""; // название

private String tcomstr = ""; // путь

private String tversion = ""; // версия

private String tdatan = ""; // дата выхода

private String tfio = ""; // имя разработчика

private String ttel = ""; //телефон разработчика

private String temail = ""; //почта разработчика

private String thelpstr = ""; // путь к помощи

private String zadnamev = ""; //название приложения верхнего ур.

private String comstrv = ""; // путь к приложению верхнего ур.

private String comment = ""; // сообщение о ЕСПП

private String[] pril\_tables; // список таблиц

public String getTmessage() {

return tmessage;

}

public void setTmessage(String tmessage) {

this.tmessage = tmessage;

}

public String getTzadname() {

return tzadname;

}

public void setTzadname(String tzadname) {

this.tzadname = tzadname;

}

public String getTcomstr() {

return tcomstr;

}

public void setTcomstr(String tcomstr) {

this.tcomstr = tcomstr;

}

public String getTversion() {

return tversion;

}

public void setTversion(String tversion) {

this.tversion = tversion;

}

public String getTdatan() {

return tdatan;

}

public void setTdatan(String tdatan) {

this.tdatan = tdatan;

}

public String getTfio() {

return tfio;

}

public void setTfio(String tfio) {

this.tfio = tfio;

}

public String getTtel() {

return ttel;

}

public void setTtel(String ttel) {

this.ttel = ttel;

}

public String getTemail() {

return temail;

}

public void setTemail(String temail) {

this.temail = temail;

}

public String getThelpstr() {

return thelpstr;

}

public void setThelpstr(String thelpstr) {

this.thelpstr = thelpstr;

}

public String getZadnamev() {

return zadnamev;

}

public void setZadnamev(String zadnamev) {

this.zadnamev = zadnamev;

}

public String getComstrv() {

return comstrv;

}

public void setComstrv(String comstrv) {

this.comstrv = comstrv;

}

public String[] getPril\_tables() {

return pril\_tables;

}

public void setPril\_tables(String[] pril\_tables) {

this.pril\_tables = pril\_tables;

}

public String getComment() {

return comment;

}

public void setComment(String comment) {

this.comment = comment;

}

@PostConstruct

public void postConstruct() {

// HttpServletRequest httpReq = (HttpServletRequest) FacesContext.getCurrentInstance().getExternalContext().getRequest();

PrilInfoRemote ejbPI = null;

try {

final InitialContext ic = new InitialContext();

ejbPI = (PrilInfoRemote) ic.lookup("global/prilinfo-1.0/PrilInfo!org.vniizht.prilinfo.PrilInfoRemote");

HashMap prinfo = ejbPI.info("GOSK"); // указываете код своего приложения

this.setTdatan((String)prinfo.get("datan"));

this.setTemail((String)prinfo.get("email"));

this.setTfio((String)prinfo.get("fio"));

this.setThelpstr((String)prinfo.get("helpstr"));

this.setTtel((String)prinfo.get("tel"));

this.setTversion(((String)prinfo.get("version"))+".20191128");

this.setTzadname((String)prinfo.get("zadname"));

this.setTcomstr((String)prinfo.get("comstr"));

this.setZadnamev((String)prinfo.get("zadnameV"));

this.setComstrv((String)prinfo.get("comstrV"));

this.setPril\_tables((String[])prinfo.get("pril\_tables"));

this.setComment((String)prinfo.get("comment"));

}catch (Exception ex) {

} finally {

}

}

}

1. **Вывод этих полей на страничке:**

~~<div class="page-header">~~

~~<nav class="navbar pl-0 px-0 pb-0 py-0">~~

~~<div class="row align-items-end">~~

~~<div class="col-lg-4">~~

~~<div class="page-header-breadcrumb">~~

~~<ul>~~

~~<li class="breadcrumb-item">~~

~~<a class="dropdown-toggle" data-toggle="dropdown" href="#"><i class="feather icon-info"></i></a>~~

~~<ul class="dropdown-menu dropdown-menu-custom">~~

~~<p><strong>Версия программы:</strong> #{GosForms.tversion}<br /><strong>Дата выхода:</strong> #{GosForms.tdatan} г.<br /><strong>Технолог:</strong> #{GosForms.tfio.trim()}<br /><strong>Тел.:</strong> #{GosForms.ttel.trim()}<br /><strong>Эл. адрес:</strong> #{GosForms.temail.trim()}</p>~~

~~</ul>~~

~~</li>~~

~~<li class="breadcrumb-item" data-toggle="modal" data-target="#myModal">~~

~~<a href="#" title="Помощь">~~

~~<i class="feather icon-help-circle"></i>~~

~~</a>~~

~~</li>~~

~~<li class="breadcrumb-item">~~

~~<p:commandLink id="clearButtonP2" action="#{goskController.clearp}" update="select\_date, messages , bodytable" onclick="$('input[id$=filter]').val('')" onsuccess="$(function() {roles(document.getElementsByClassName('ui-datatable-scrollable-body')[0].offsetTop + 25);});" >~~

~~<i class="feather icon-refresh-ccw"></i>~~

~~</p:commandLink>~~

~~</li>~~

~~<li class="breadcrumb-item">~~

~~<a class="dropdown-toggle" data-toggle="dropdown" href="#" title="Загрузить"><i class="feather icon-download glicon"></i></a>~~

~~<ul class="dropdown-menu">~~

~~<li><h:commandLink id="exportButtonE" action="#{goskController.exportAll}"~~

~~value="Полный экспорт">~~

~~<i class="feather icon-file glicon" ></i>~~

~~</h:commandLink></li>~~

~~<li><h:commandLink id="exportButtonF" action="#{goskController.export}"~~

~~value="Экспорт страницы">~~

~~<i class="feather icon-file-text glicon" ></i>~~

~~</h:commandLink></li>~~

~~</ul>~~

~~</li>~~

~~<li class="breadcrumb-item">~~

~~<a href="#" onclick="toggleControlBar()">~~

~~<span title="Свернуть" >~~

~~<i class="feather icon-arrow-up glicon" onclick="document.getElementById('gosForm:up').click();" ></i>~~

~~</span>~~

~~<span class="hidden" title="Развернуть">~~

~~<i class="feather icon-arrow-down glicon" onclick="document.getElementById('gosForm:down').click();" ></i>~~

~~</span>~~

~~</a>~~

~~</li>~~

~~</ul>~~

~~<div class="col-sm-0 col-md-0 col-lg-0 border\_thin mt-0 px-0 " style="display: none">~~

~~<h:panelGroup layout="block" >~~

~~<p:commandLink id="up" action="#{GosForms.setCb(true)}" oncomplete="$(function() {roles(document.getElementsByClassName('ui-datatable-scrollable-body')[0].offsetTop + 20);});"/>~~

~~<p:commandLink id="down" action="#{GosForms.setCb(false)}" oncomplete="$(function() {roles(document.getElementsByClassName('ui-datatable-scrollable-body')[0].offsetTop + 20);});"/>~~

~~</h:panelGroup>~~

~~</div>~~

~~</div>~~

~~</div>~~

~~<div class="col-lg-8 navbar2">~~

~~<div class="page-header-breadcrumb">~~

~~<ul class="breadcrumb-title">~~

~~<li class="breadcrumb-item">~~

~~<a href="#{GosForms.comstrv}"> <i class="feather icon-home"></i> #{GosForms.zadnamev}</a>~~

~~</li>~~

~~<li class="breadcrumb-item">~~

~~<a href="#{GosForms.tcomstr}">#{GosForms.tzadname}</a>~~

~~</li>~~

~~</ul>~~

~~</div>~~

~~</div>~~

~~</div>~~

~~</nav>~~

~~</div>~~

~~<div class="modal" id="myModal">~~

~~<div class="modal-dialog" style="max-width: 95vw">~~

~~<div class="modal-content">~~

~~<!-- Modal Header -->~~

~~<div class="modal-header">~~

~~<h4 class="modal-title">Помощь</h4>~~

~~<button type="button" class="close" data-dismiss="modal">&times;</button>~~

~~</div>~~

~~<!-- Modal body -->~~

~~<div class="modal-body" style="max-height: 72vh">~~

~~<iframe src="#{GosForms.thelpstr}" class="" style="height: 62vh; width: 91vw;"/>~~

~~</div>~~

~~<!-- Modal footer -->~~

~~<div class="modal-footer">~~

~~<button type="button" class="btn btn-danger" data-dismiss="modal">Закрыть</button>~~

~~</div>~~

~~</div>~~

~~</div>~~

~~</div>~~

Новый вариант вывода вместе с сообщением о ЕСПП

<li class="breadcrumb-item-c">

<a class="dropdown-toggle" data-toggle="dropdown" href="#"><i class="feather icon-info" title="Информация о программе"></i></a>

<ul class="dropdown-menu dropdown-menu-custom">

<p><span style='margin-right: 17px;'><i class="fa fa-info fa-1x"></i></span><span>Версия программы: #{GosForms.tversion} </span><br /><span style='margin-right: 9px;'><i class="fa fa-calendar fa-1x"></i></span><span>Дата обновления: #{GosForms.tdatan} г. </span><br /><span style='margin-right: 5px;'><i class="fa fa-address-card-o fa-1x"></i></span><span>Технолог: #{GosForms.tfio.trim()} </span><br />

<span> #{GosForms.comment.trim()} </span>

</p>

</ul>

</li>

# Приложение 17 Описание работы с банком микросервисов

Таблица 1. Список доступных для получения данных (с параметрами):

|  |  |  |
| --- | --- | --- |
| Описание | Название списка | Параметры |
| Список перевозчиков | perList | data\*,  gos – числовой код государства (напр. Россия – 20),  skp – числовой код перевозчика (напр. ДОСС - 40) |
| Список направлений | dirList | без параметров |
| Список брендов поездов | brandList | data |
| Список дорог | dorList | data,  gos – числовой код государства (напр. Россия – 20) |
| Список тарифных планов | tplanList | data,  skp – числовой код перевозчика (напр. ДОСС - 40) |
| Список регионов | sfList | data |
| Список филиалов | filList | data |
| Список линейных предприятий | linprList | data |
| Список названий станций | stanList | data,  gos – числовой код государства (напр. Россия – 20),  dor (ввиде строки кода дороги char1, в строчку через запятую (пример: О,Г,Д) |
| Список агентов | agentList | data |
| Список пунктов продажи | ppList | data,  dor – буквенный код дороги char1 |
| Список ИВЦ | vcList | data |
| Список субагентов | subagentList | data |
| Список государств | gosList | data |

\*Параметр data является обязательным там, где он указан в списках параметров.

Остальные параметры необязательны, т.е. можно не указывать, соответственно фильтра по ним не будет. Все параметры передаются любым JSON-типом (строка\число).

data – дата в формате ГГГГ-ММ-ДД (напр. 2022-01-20)

Таблица 2. Условия выборки данных

| Название параметра | Название таблицы | Параметры выборки | Входные параметры | Условия выборки | Выходные параметры |
| --- | --- | --- | --- | --- | --- |
| Перевозчик | nsi.sobper | Начальная дата продажи (datanp)  Конечная дата продажи (datakp)  Начальная дата отправления (datano)  Конечная дата отправления (datako)  Код государства (gos) | Дата | 1) Дата > или = datanp и < или = datakp  2) Дата > или = datano и < или = datako | Название перевозчика (nazvp)  Сетевой код перевозчика (skp) |
| Направление | car.guidedir | нет | - |  | Код направления (dir)  Наименование направления  (dirname) |
| Бренд | nsi.tbr | Дата начала действия (datan)  Дата окончания действия (datak)  Начальная дата отправления (dataotpn) Конечная дата отправления (dataotpk) | Дата | 1) Дата > или = datan и < или = datak  2) Дата > или = dataotpn и < или = dataotpk | Бренд код Экспресс (brkod)  Название бренд-кода (рус) (brname) |
| Дорога | nsi.dork | Дата начала действия (d\_datan)  Дата конца действия (d\_datak)  Дата начала использования (d\_datani)  Дата конца использования (d\_dataki) | Дата | Дата > или = d\_datan и < или = d\_datak | Наименование дороги (d\_name)  Код дороги буква (d\_kod)  Трехзначный номер дороги (d\_nom3) |
| Тарифный план | nsi.tplan | Дата начала действия строки (datan)  Дата окончания действия строки (datak)  Код перевозчика (skp) | Дата | Дата > или = datan и < или = datak | 2 цифры - уточняющий код льготы или код тарифного плана (kodn)  Информация o тарифном плане (itxt) |
| Регион | nsi.sf | Дата начала использования (sf\_datani)  Дата конца использования (sf\_dataki) | Дата | Дата > или = sf\_datani и < или = sf\_dataki | Код субъекта (МВД) (sf\_kod2) Наименование субъекта (sf\_name) |
| Филиал | nsi.fil | Дата начала действия (fil\_datan)  Дата начала использования (fil\_datani) Дата окончания действия (fil\_datak) Дата окончания использования (fil\_dataki) | Дата | 1) Дата > или = fil\_datan и < или = fil\_datak  2) Дата > или = fil\_datani и < или = fil\_dataki | Шифр филиала (Код дороги из таблицы DOR)  (fil\_shifr) Мнемокод филиала (Сокращенное название из таблицы DOR) (fil\_mnkod) Название филиала (fil\_nazv) |
| Линейное предприятие | nsi.linpr | Дата начала действия строки (datan)  Дата окончания (datak) | Дата | Дата > или = datan и < или = datak | Код предприятия (kod)  Название предприятия (nazv) |
| Название станции | nsi.stanv | Дата начала действия (datand) Дата начала использования (datani) Дата отмены действия (datakd) Дата окончания использования (dataki)  Код государства административного (gos) Код дороги (dor) | Дата | 1) Дата > или = datand и < или = datakd  2) Дата > или = datani и < или = dataki | Код станции (stan)  Тип объекта (tipob)  Полное название станции (pnazv) Сокращенное название станции (snazv) |
| Агент | nsi.agent | Дата начала действия (datand)  Дата  окончания действия (datakd) | Дата | Дата > или = datand и < или = datakd | Код агента (agent) Название (nazv) |
| Название пункта продажи + Номер тепминала | nsi.term | Дата начала действия (term\_datan)  Дата окончания действия (term\_datao)  Шифр дороги (term\_sdor) | Дата | Дата > или = term\_datan и < или = term\_datao | Порядковый номер ПП (term\_npp)  Название пункта продажи (term\_pp)  Канальный номер ТМ (term\_ntm) |
| ИВЦ | nsi.vc | Дата начала действия (vc\_datan) Дата начала использования (vc\_datani) Дата конца действия (vc\_datak) Дата конца использования (vc\_dataki) | Дата | 1) Дата > или = vc\_datan и < или =vc\_datak  2) Дата > или = vc\_datani и < или = vc\_dataki | Код ВЦ (vc\_kodvc) Наименование ВЦ (vc\_name) Сокращенное наименование ВЦ (vc\_names) |
| Субагент | nsi.subagnng | Дата начала действия (datand)  Дата окончания действия (datakd) | Дата | Дата > или = datand и < или = datakd | Код субагента (subagent)  Название (nazv) |
| Государство | nsi.gosk | Дата начала действия (g\_datan) Дата начала использования (g\_datani) Дата конца действия (g\_datak) Дата конца использования (g\_dataki) | Дата | 1) Дата > или = g\_datan и < или =g\_datak  2) Дата > или = g\_datani и < или = g\_dataki | Код государства (g\_kod)  Наименование государства (g\_name)  Сокращенное наименование государства (g\_names) |

Запрос к банку осуществляется из JavaScript посредством POST-запроса по адресу: document.location.origin/servicebank/getdata

\*Если обращаться по прямому адресу или извне контура SecurityDomain сервера приложений, где установлен банк – необходимо будет в запросе прописать дополнительно заголовки для авторизации в контур (на данный момент у нас используется Basic авторизация), что совершенно излишне при обращении к банку из приложения, установленного на том же веб-сервере.

В теле запроса должен содержаться JSON с параметрами списков, которые необходимо получить.

Формат его следующий:

{

  "название списка 1":

[

    {

      "параметр 1": "значение",

      "параметр 2": "значение",

      "параметр 3": "значение", …. и т.д.

   }

  ],

  "название списка 2":

[

    {

      "параметр 1": "значение",

      "параметр 2": "значение",

      "параметр 3": "значение", …. и т.д.

    }

  ]

}

На данный момент в банке содержится 13 списков, можно получать один, несколько или все сразу.

После отправки запроса, если все нормально, в теле ответа приходит пакет запрошенных списков в виде JSON.

Формат его следующий:

{

    "список 1":

[

        {

            "поле1": "значение",

            "поле2": "значение", … и т.д.

        },

        {

            "поле1": "значение",

            "поле2": "значение", … и.т.д.

        }, …и т.д.

    ],

    "список 2":

[

        {

            "поле1": "значение",

            "поле2": "значение", … и т.д.

        },

        {

            "поле1": "значение",

            "поле2": "значение", … и.т.д.

        }, …и т.д.

    ], … и т.д.

}

Пример

Запрос:

{

  "perList": [

    {

      "data": "2022-07-01",

      "gos": "20"

    }

  ],

  "sfList": [

    {

      "data": "2022-07-01"

    }

  ]

}

Ответ:

{

    "perList": [

        {

            "skp": 64,

            "nazvp": "АВТОБУСЫ ЖДЯ   "

        },

        {

            "skp": 57,

            "nazvp": "АНО \"ЕТД\"      "

        },

---- // ----

        {

            "skp": 69,

            "nazvp": "ЮЖНАЯ ППК      "

        }

    ],

    "sfList": [

        {

            "sf\_kod2": "80",

            "sf\_name": "АГИНСКИЙ БУРЯТСКИЙ АВТОНОМНЫЙ ОКРУГ               "

        },

        {

            "sf\_kod2": "22",

            "sf\_name": "АЛТАЙСКИЙ КРАЙ                                    "

        },

----///----

        {

        {

            "sf\_kod2": "89",

            "sf\_name": "ЯМАЛО-НЕНЕЦКИЙ АВТОНОМНЫЙ ОКРУГ                   "

        },

        {

            "sf\_kod2": "76",

            "sf\_name": "ЯРОСЛАВСКАЯ ОБЛАСТЬ                               "

        }

    ]

}

# Приложение 18 Описание работы с функцией получения курса валют

При переводе валют должна использоваться функция для получения курса валют get\_kurs\_vv, расположенная в схеме car. Вызов функции выполняется следующим образом:

car.get\_kurs\_vv(

<kodgp character>,  -- код государства продажи

<kodgn character>,  -- код государства в который нужно перевести

<dat date> -- дата на которую нужно перевести валюту)

**ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ**

| Версия | Количество страниц | Дата | Автор | Описание изменений |
| --- | --- | --- | --- | --- |
| 1.0 | 12 | 22.07.2022 | Артюхина М.А. | Создание версии 1.0 |
| 1.1 | 43 | 05.09.2022 | Артюхина М.А. | Внесены изменения в Общие положения, Приложение 3. Добавлены Приложения 14-18 |
| 1.2 | 44 | 10.10.2022 | Ширман Е.П. | Добавлена информация о новом поле comment в приложении prilinfo отвечающим за выдачу сообщения “От ИВЦ все запросы - через ЕСПП” |