

# RunaWFE. Клиент-оповещатель о поступивших задачах. Руководство разработчика.

---

## Версия 3.0

© 2004-2011, ЗАО “Руна”. RunaWFE является системой с открытым кодом и распространяется в соответствии с LGPL лицензией (<http://www.gnu.org/licenses/lgpl.html>).

## Структура проекта

- lib — дополнительные библиотеки
- resources — настройки приложения
- img — изображения
- src — исходные файлы

## Архитектура

Приложение работает с системой RunaWFE по двум «каналам» одновременно: RMI и HTTP. Обращения к системе по RMI и HTTP не зависимы друг от друга и используются для разных целей. RMI используется для получения информации, необходимой при отображении трея приложения и всплывающих подсказок. Общение по протоколу HTTP напрямую клиентом-оповещателем не используется, и производится во встроенном браузере для отображения web интерфейса системы RunaWFE (основная работа пользователя с системой RunaWFE происходит внутри встроенного браузера посредством web интерфейса).

Перед началом работы необходимо, что бы клиент-оповещатель аутентифицировался в системе RunaWFE. Аутентификация происходит как по протоколу HTTP, так и по RMI. До аутентификации клиента-оповещателя по RMI в трее будет отображаться «кирпич» и функциональность по оповещению о новых задачах будет недоступна. До аутентификации по протоколу HTTP web интерфейс системы RunaWFE будет недоступен. В то же время незавершенная аутентификация по RMI (HTTP) не мешает использованию web интерфейса (оповещения о новых заданиях).

Клиент-оповещатель периодически проверяет текущие задания пользователя (по протоколу RMI) и в случае изменения заданий пользователя сообщает пользователю через трей и всплывающие подсказки. По событию MouseOver в System tray так-же происходит проверка заданий и оповещение пользователя.

## Авторизация

Для добавления механизма авторизации клиента-оповещателя по протоколу RMI необходимо создать класс, реализующий интерфейс `ru.runa.notifier.auth.Authenticator` и добавить созданный класс в `LoginHelper`. В настоящее время реализовано 2 аутентификатора:

- `UserInfoAuthenticator (userinfo)` — для аутентификации в системе RunaWFE с использованием логина и пароля пользователя. Логин и пароль пользователя попросят ввести перед началом работы в сплывающем окне.
- `KerberosAuthenticator (kerberos)` — для аутентификации в системе RunaWFE с использованием протокола kerberos. Аутентификация происходит без дополнительных запросов у пользователя.

Механизм аутентификации по протоколу HTTP задается с помощью url, относительно корня web приложения системы RunaWFE. Доступные механизмы аутентификации по протоколу HTTP зависят от настроек системы RunaWFE и включают в себя:

- /login.do — для аутентификации с использованием логина и пароля пользователя. Для корректной работы необходимо, что бы аутентификация по RMI так-же использовала логин и пароль.
- /krblogin.do — для аутентификации по протоколу kerberos.
- /ntlmlogin.do — для аутентификации по протоколу ntlm

## Описание классов

### **ru.runa.notifier.PlatformLoader**

Стартовый класс приложения. При старте показывает сплеш (картинку) в другом Shell'е, после инициализации запускает другой Shell с поддержкой system tray.

### **ru.runa.notifier.GUI**

Главный класс графической оболочки приложения, создает Browser, вешает обработчики событий на 3 правые кнопки в верхнем углу экрана.

### **ru.runa.notifier.tray.SystemTray**

Обеспечивает поддержку в system tray. Содержит обработчики событий system tray.

### **ru.runa.notifier.tray.SystemTrayAlert**

Реализует всплывающее окно возле system tray. Содержит функции отрисовки этого окна. Вызывается при необходимости уведомить пользователя о новых задачах.

### **ru.runa.notifier.checker.TaskChecker**

Обеспечивает периодическую (TimerTask) и принудительную (SystemTray) проверку новых задач.

### **ru.runa.notifier.auth.LoginHelper**

Класс служит для аутентификации в системе RunaWFE по протоколу kerberos. В случае неуспешной аутентификации умеет выводить диалоговое окно.

### **ru.runa.notifier.auth.LoginModuleResources**

Служит контейнером для настроек kerberos\_module.properties

### **ru.runa.notifier.auth.LoginConfiguration**

Конфигурация для аутентификации

### **org.eclipse.swt.widgets.STrayItem**

Класс для обработки сообщений SWT. Создан для возможности реагирования на события MouseOver.

### **ru.runa.notifier.util.ExtendedThread**

Используется классом SystemTrayAlert

### **ru.runa.notifier.util.WidgetsManager**

Класс — помощник

### **ru.runa.notifier.util.ResourcesManager**

Класс для загрузки и хранения значений application.properties

### **ru.runa.notifier.util.LayoutsManager**

Класс — помощник, используется классом SystemTrayAlert

### **ru.runa.notifier.util.ImageManager**

Класс — менеджер по работе с картинками (загрузка, выгрузка)

## Настройка приложения

### **kerberos\_module.properties**

appName=com.sun.security.jgss.initiate

moduleClassName=com.sun.security.auth.module.Krb5LoginModule

useTicketCache=true

doNotPrompt=true

debug=false

serverPrincipal=WFServer

appName — название логин-модуля

moduleClassName — класс логин-модуля

useTicketCache — использовать ли кеш для билетов Kerberos

doNotPrompt — не использовать пользовательский ввод в любой ситуации

debug — дополнительное логирование при аутентификации

serverPrincipal — логин с серверной стороны (конфигурируется в АД)

### **application.properties**

Все русскоязычные значения задаются в кодировке ANSI

application.name, popup.newtasks, user.name, user.password, login.message,

retry.message, error.login, error.internal, popup.no.tasks, popup.tease,

popup.error, menu.open, menu.exit — локализованные сообщения пользователя

server.url - URL сервера

(Пример: http://localhost:8080/wfe)

login.relative.url — начальная страница при запуске системы (аутентификация)

(Примеры:

/login.do — форма для ввода логина и пароля;

/ntlmlogin.do — аутентификация по NTLM;

/krblogin.do - аутентификация по Kerberos;

)

start.relative.url — страница с задачами

(Пример: /manage\_tasks.do?tabForwardName=manage\_tasks)

check.tasks.timeout — Таймаут проверки новых задач, мс

authentication.type — задает тип аутентификации по RMI (возможные значения userinput, kerberos)

infopath.submit.button.name — Название кнопки запуска формы InfoPath

infopath.cancel.button.name — Название кнопки отмены формы InfoPath

(Только для аутентификации по логину и паролю)

userinput.default.login — логин по умолчанию, рекомендуется оставить пустым

userinput.default.password — пароль по умолчанию, рекомендуется оставить пустым