

**Описание функциональных характеристик
программного обеспечения и информация,
необходимая для установки и эксплуатации
программного обеспечения iSimpleNotify**

Оглавление

1	4
1.1	5
1.2	6
2	7
2.1	8
2.2	8
2.3	8
2.4	10
3	11
4	13
4.1	14
4.2	15
4.3	15
4.4	17
4.5	17
4.5.1	18
4.5.2	19
4.5.3	19
4.5.4	20
4.5.5	20
4.5.6	21
4.5.7	21
4.6	26
4.7	26
5	28
5.1	29
5.1.1	29
5.1.2	32
5.1.3	35
5.2	37
5.2.1	38
5.3	39
5.3.1	40
5.4	40
5.4.1	40
5.4.2	41
5.4.3	42
5.4.4	43
5.4.5	43
5.4.6	44
5.5	47

1 Функциональные характеристики

1.1 Ключевые принципы разработки ДБО iSimpleLab

- Кроссплатформенный сервер приложений, позволяющий удовлетворить ИТ-Стратегию кредитной организации любого масштаба;
- Промышленный открытый API, реализующий концепцию OpenAPI и позволяющий разработать своими силами или силами экспертной команды независимый гибкий слой интерфейса, а также прямую интеграцию с клиентскими информационными системами;
- Модульная архитектура, позволяющая реализовать независимые и отказоустойчивые компоненты для более стабильной работы всей платформы;
- Портал – автономный front-end платформы iSimpleBank 2.0, который взаимодействует с back-end системы посредством Rest-API. Данное решение позволяет реализовать любые замыслы дизайнеров и открывает новые горизонты по развитию и запуску в личном кабинете Банка любых внешних сервисов не связанных с функционалом классического ДБО;
- Использование инновационных средств безопасности, совмещающих удобство работы с ними и универсальность по отношению к каналу работы (web-кабинет, мобильное приложение на любой платформе, либо внешний портал, в который встроены сервисы банка);
- Готовые средства и возможности интеграции практически со всеми АБС, CRM, карточными процессингами, платежными системами и системами денежных переводов.

Платформа ДБО iSimpleBank 2.0 построена на современных технологиях и использует стандартные промышленные сервера приложений и СУБД, как коммерческие, так и с открытым кодом:

- Java в качестве кросс-платформенного языка разработки;
- Операционные системы: Windows, Unix, Linux;
- Сервера приложений: JBoss Application Server, Oracle WebLogic, IBM WebSphere Application Server;
- СУБД: Oracle, PostgreSQL, Microsoft SQL Server, Firebird/RED Database;
- XML-описания ресурсов системы, таких как конфигурации, визуальные формы, меню, схемы документов, языковые ресурсы;
- Groovy – открытые скрипты, используемые в системе;

Использование веб-сервисов для задач интеграции с другими системами. Клиентская часть интернет-клиента выполнена на технологии полностью тонкого клиента, и работает на следующих операционных системах и браузерах:

- Операционные системы: Windows, Unix, Linux, MacOS, iOS, Android;
- Браузеры: IE, Firefox, Chrome, Safari, Opera, Android и др в зависимости от выбранной Web-платформы.

1.2 Ключевые функции

iSimpleNotify – модуль «Нотификация по SMS и e-mail» является дополнительным к базовому модулю iSimpleRetail и обеспечивает:

- Отправку нотификаций о различных событиях клиентам Банка в соответствии с подпиской;
- Интеграцию с центрами отправки SMS по протоколу SMPP;
- Интеграцию с почтовыми серверами по протоколу SMTP;
- Загрузку событий нотификаций из внешнего источника;
- Автоматическое переключение на альтернативный канал отправки SMS при недоступности основного;
- Проверку IMSI при поддержке SMS-провайдером. (*International Mobile Subscriber Identity (IMSI) – международный идентификатор мобильного абонента (индивидуальный номер абонента), ассоциированный с каждым пользователем мобильной связи стандарта GSM, UTM или CDMA*)

2 Руководство по развертыванию системы ДБО iSimpleRetail

2.1 Назначение документа

Данное руководство предназначено для администраторов, выполняющих развертывание системы ДБО iSimpleRetail.

Основные задачи настоящего руководства:

- определить способы установки каждого компонента системы ДБО iSimpleRetail;
- установить требования к сопутствующему программному обеспечению и их версии;
- сформулировать конфигурации аппаратно-технического обеспечения для различных диапазонов работающих клиентов;
- указать последовательность действий для первоначальной настройки системы из дистрибутива;
- описать процесс запуска системы и проверки её работоспособности после развертывания.

Во избежание возможных сбоев в работе системы порядок установки и настройки должен выполняться строго в соответствии с данным документом.

Настоящий документ передается банку и подразделениям компанией iSimpleLab на условиях конфиденциальности и только для служебного использования.

2.2 Требуемые знания для установки и конфигурирования системы

Процесс установки и конфигурирования системы ДБО iSimpleRetail предполагает наличие у администратора следующих знаний и навыков:

- Навыков администрирования серверов Windows или UNIX/Linux.
- Навыков администрирования серверов веб-приложений.
- Четкого понимания информационной безопасности инфраструктуры компании, в том числе поставщиков проверки безопасности, использования SSL и т.д.
- Четкого понимания порядка доступа к базам данных и прочим информационным средам компании.
- Четкого понимания использования в компании сетевой среды и использования портов.
- Четкого понимания систем криптографической защиты информации (СКЗИ), принципов закрытых ключей и распространения сертификатов при использовании данных сервисов.

2.3 Общие сведения о процессе установки

Что следует знать о процедуре развертывания системы ДБО iSimpleRetail:

- Для стабильной работы системы используйте рекомендованные требования к аппаратно-техническому обеспечению для соответствующих диапазонов

подключенных клиентов физических и юридических лиц (см. пункт Конфигурация оборудования).

- Для гарантированной работы системы рекомендуется строго соблюдать требования к типу и версииности сопутствующего программного обеспечения (см. пункт Сопутствующее программное обеспечение).
- Установка системы ДБО iSimpleRetail выполняется пакетом установки (дистрибутивом) системы.
- Для всех типов СУБД и серверов приложений используется единый пакет установки.
- В процессе установки системы ДБО iSimpleRetail выполняется формирование структуры базы данных сервисными скриптами (см. пункт Создание структуры базы данных) и устанавливается J2EE-контейнер с бизнес-логикой системы (см. пункт Развертывание J2EE-контейнера приложения).
- Допускается проводить создание структуры базы данных с любой рабочей станции, где инициализирована виртуальная машина Java; при этом, обращение к базе данных осуществляется через jdbc-драйвер, входящий в поставку системы.
- Не допускается исполнять SQL-скрипты, поставляемые в дистрибутиве системы, для модификации структуры базы данных сторонними утилитами для исполнения SQL-запросов. Все SQL-скрипты должны исполняться строго предусмотренным для этого инструментарием из дистрибутива. Подобное ограничение связано с тем, что синтаксис SQL-скриптов по-разному интерпретируется сервисной утилитой создания структуры базы данных в зависимости от типа используемой СУБД.

- Принцип установки подсистем продукта ДБО iSimpleBank 2.0

2.4 Описание компонентов системы

Описание компонентов системы ДБО iSimpleRetail приведены в таблице.

Компонента	Описание
СУБД	<p>База данных, в которой хранится:</p> <ul style="list-style-type: none"> ● Список клиентов – пользователей системы, включая их реквизиты, информацию о банковских продуктах и средствах доступа. ● Документы, отправленные клиентами через систему ДБО. ● Различные справочники системы. ● Настройки, выполненные пользователями системы
Сервер приложений	<p>Основной программный модуль системы, обрабатывающий подключения пользователей и реализующий бизнес-логику работы приложения. Сервер приложений взаимодействует со следующими компонентами:</p> <ul style="list-style-type: none"> ● СУБД – для хранения информации. ● Банковскими и внешними системами – для задач интеграции. ● Клиентским рабочим местом – предоставляя веб-интерфейс для работы клиента. ● Рабочим местом сотрудника банка – предоставляя веб-интерфейс для работы сотрудника банка
Веб-сервер распределения нагрузки	<p>Программный модуль, повышающий безопасность и снижающий нагрузку с сервера приложений (см. подраздел Настройка веб-сервера Nginx), распределения нагрузки между кластером серверов приложений (см. подраздел Настройка кластеризации). Также может быть использован для установки порталного решения iSimplePortal.</p>
Рабочее место сотрудника банка	<p>Рабочее место сотрудника банка – операциониста, администратора, офицера безопасности</p>
Клиентское рабочее место	<p>Рабочее место клиента банка. Представляет собой веб-интерфейс системы ДБО iSimpleRetail</p>
Мобильные клиенты iOS/Android	<p>Мобильные приложения для платформ iOS/Android, предоставляющие возможность клиентам банка использовать функции системы со смартфона или планшета</p>

3 Последовательность установки системы ДБО iSimpleRetail

Установка системы ДБО iSimpleRetail должна выполняться в соответствии с последовательностью действий, определённой в настоящем разделе.

1. Установка сопутствующего программного обеспечения (см. подраздел Установка сопутствующего программного обеспечения).
2. Загрузка и распаковка файлов с ресурсами для установки (см. пункты Загрузка дистрибутива системы, Подготовка пакета установки для использования).
3. Подготовка рабочей базы данных (см. пункт Подготовка базы данных для загрузки ресурсов приложения).
4. Настройка конфигурационного файла `ups.properties` для подключения к рабочей базе данных для формирования структуры базы данных (см. пункт Настройка конфигурационного файла).
5. Выполнение скриптов создания базы данных, с соблюдением последовательности запуска исполняемых файлов (см. пункт Создание структуры базы данных).
6. Установка и настройка сервера приложений (см. пункт Установка и настройка сервера приложений).
7. Выполнение рекомендованного первоначального конфигурирования сервера приложений (см. пункт Конфигурирование сервера приложений).
8. Развертывание J2EE-контейнера с системой ДБО iSimpleRetail на сервере приложений (см. пункт Развертывание J2EE-контейнера приложения).
9. Настройка параметров подключения к базе данных в конфигурационном файле J2EE-контейнера `spring_datasources.xml` (см. пункт Настройка параметров подключения к базе данных).
10. Конфигурирование ресурсов системы ДБО iSimpleRetail (см. пункт Конфигурирование ресурсов системы ДБО iSimpleRetail).
11. Генерация SSL (см. пункт Генерация ключа SSL и запроса на сертификат).
12. Настройка веб-сервера распределения нагрузки Nginx и организация защищённого SSL-соединения (см. раздел Настройка веб-сервера распределения нагрузки Nginx).
13. Запуск сервера приложений (см. пункт Запуск сервера приложений и проверка доступности системы ДБО iSimpleRetail).
14. Проверка корректности работы приложения и переход на стартовую страницу логина системы (см. пункт Запуск сервера приложений и проверка доступности системы ДБО iSimpleRetail).

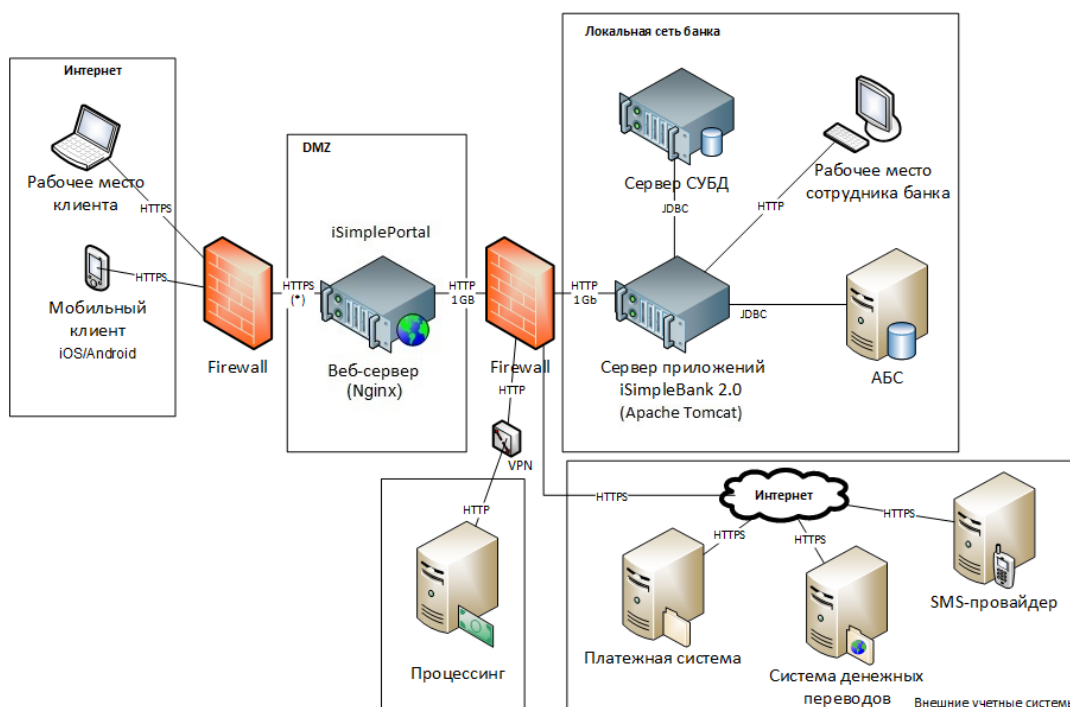
4 Подготовка к установке

Для установки системы ДБО iSimpleRetail необходимо загрузить дистрибутив системы и подготовить его для выполнения процедуры развертывания.

- Физическая схема развертывания
- Логическая схема развертывания
- Требования к сопутствующему программному и аппаратному обеспечению
- Требования к каналам связи
- Конфигурация оборудования
- Подготовка ОС
- Загрузка дистрибутива системы
- Подготовка пакета установки для использования
- Установка Java Virtual Machine

4.1 Физическая схема развертывания

Типовая физическая схема развертывания системы ДБО iSimpleRetail представлена на рисунке:



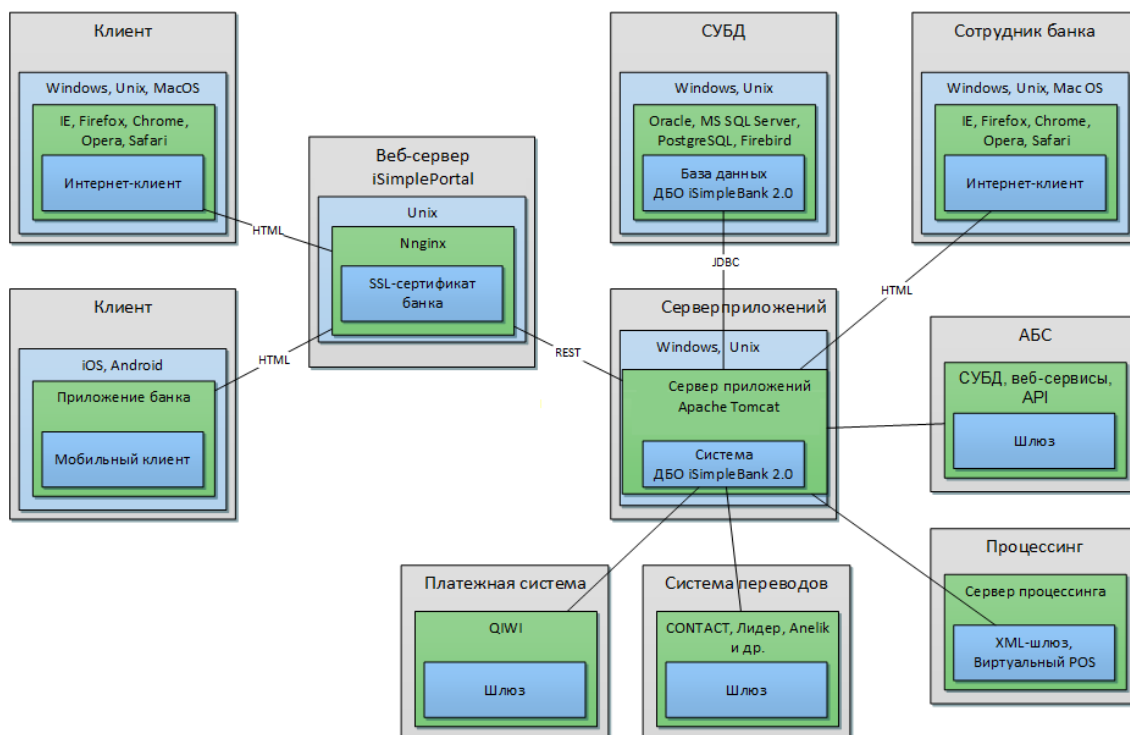
(*) см. подраздел Требования к каналам связи

Комментарии к схеме:

- Приведённая схема отражает типовую конфигурацию и может содержать допущения как по количеству интегрируемых систем, так и используемым серверам;
- В частном случае возможна организация компонентов сервера приложений iSimpleRetail, сервера СУБД и веб-сервера распределения нагрузки на едином сервере ДБО.

4.2 Логическая схема развертывания

Логическая схема развертывания системы ДБО iSimpleRetail представлена на рисунке.



4.3 Требования к сопутствующему программному и аппаратному обеспечению

При выборе типа и версии программного обеспечения рекомендуется ориентироваться на разработчика ПО и аппаратного обеспечения и использовать актуальные версии и модификации, предназначенные для промышленной эксплуатации.

Требования к сопутствующему программному и аппаратному обеспечению системы ДБО iSimpleRetail приведены в таблице:

Компонента	Тип категории	Наименование и версия
Требования к программному обеспечению		
Сервер приложений	Операционная система*	Microsoft Windows Server 2008 или выше
		Unix, Linux
	Сервер приложений	Apache Tomcat версий 7.0.82, 7.0.88, 8.0.52, 8.5.30
	Среда виртуальной машины	Oracle Java Runtime Environment (JRE) версия 1.7, 1.8, разрядность должна соответствовать разрядности операционной системы сервера

	<i>Криптографическая система (при использовании)</i>	<i>Требования к программному и аппаратному обеспечению криптозащиты рассматриваются в подразделе Требования к программному и аппаратному обеспечению документа Криптозащита</i>
Сервер СУБД	СУБД	<i>Oracle Database 10g R2, Oracle Database 11g R2, Oracle Database 12c R1, 12c R2 (с версий 2.0.16.645, 2.0.15.644и7)</i>
		<i>Microsoft SQL Server 2008, Microsoft SQL Server 2012, Microsoft SQL Server 2016</i>
		<i>PostgreSQL от 9.6 и выше, 10, 11, 12.1 (с версий модулей squill-0.16.0, custcore-0.135.0, cust-ibank-fl-0.254.0)</i>
		<i>Firebird 2.0/Red Database версии 2.1.3 или выше</i>
	Операционная система	<i>Соответствующая выбранной СУБД (см. сопроводительную документацию к СУБД)</i>
Прокси-сервер/веб-сервер распределения нагрузки	Веб-сервер	<i>Nginx 1.3.9 и выше</i>
	Операционная система	<i>Unix, Linux (CentOS)</i>
Рабочее место клиента, Рабочее место сотрудника банка	Интернет-браузер	<i>Internet Explorer 11 и выше</i>
		<i>Firefox - актуальная версия (обновляется автоматически)</i>
		<i>Google Chrome - актуальная версия (обновляется автоматически)</i>
		<i>Safari 9 и выше</i>
		<i>Opera - актуальная версия (обновляется автоматически)</i>
	Операционная система*	<i>Все операционные системы, поддерживаемые браузером. Список поддерживаемых операционных систем смотрите в сопроводительной документации к соответствующим браузерам</i>
	Табличный процессор (при использовании вывода отчётов в Excel)	<i>MS Excel (версии 2003 и выше)</i>

	<i>Криптографическая система (при использовании)</i>	<i>Требования к программному и аппаратному обеспечению криптозащиты рассматриваются в подразделе Требования к программному и аппаратному обеспечению документа "Криптозащита".</i>
--	--	--

4.4 Требования к каналам связи

Для функционирования системы ДБО iSimpleRetail в штатном режиме необходимо придерживаться следующих требований, предъявляемых к каналам связи:

- минимальная величина входящего канала системы ДБО iSimpleRetail рассчитывается по следующей формуле:

*<количество пользователей системы> * <коэффициент одновременно находящихся клиентов в системе> * 32 Кбит*

*Таким образом, если принять параметр <коэффициент одновременно находящихся клиентов в системе> за величину 0,05 (5% пользователей от общего числа находятся одновременно в системе), то при количестве клиентов в 50 000 зарегистрированных пользователей, требование к величине внешнего канала для подключения клиентов составит: 50 000 * 0,05 * 32 Кбит = 80 Мбит. В приведённую формулу расчёта канала для подключения возможно подставить любую величину <коэффициент одновременно находящихся клиентов в системе> в зависимости от прогнозируемой загруженности приложения.*

4.5 Конфигурация оборудования

В разделе представлены конфигурации оборудования для установки системы ДБО iSimpleRetail, исходя из предполагаемого количества обслуживаемых клиентов системы (физических и юридических лиц).

- Конфигурация оборудования XXS (для просмотра ознакомительного пакета iSimpleStart или пробного использования)
- Конфигурация оборудования XS
- Конфигурация оборудования S
- Конфигурация оборудования M
- Конфигурация оборудования L
- Конфигурация оборудования XL (для большой численности клиентов под индивидуальный расчёт)
- Оценка мощностей в зависимости от типа среды

Описание конфигураций:

Конфигурация	Физические лица (клиентов)	Юридические лица (клиентов)	Физические, (ФЛ)/Юридические лица, (ЮЛ) (клиентов)[1]
XXS	<i>До 1000 или тестовой</i>	<i>До 200 или тестовой</i>	<i>ФЛ: до 500 ЮЛ: до 100</i>

	<i>эксплуатации/iSimpleStart</i>	<i>эксплуатации/iSimpleStart</i>	<i>или тестовая эксплуатация/iSimpleStart</i>
XS	<i>От 1 001 до 10 000</i>	<i>От 201 до 2 000</i>	<i>ФЛ: от 501 до 5 000 ЮЛ: от 101 до 1000</i>
S	<i>От 10 001 до 50 000</i>	<i>От 2 001 до 10 000</i>	<i>ФЛ: от 5 001 до 25 000 ЮЛ: от 1001 до 5 000</i>
M	<i>От 50 001 до 100 000</i>	<i>От 10 001 до 20 000</i>	<i>ФЛ: от 25 001 до 50 000 ЮЛ: от 5 001 до 10 000</i>
L	<i>От 100 001 до 200 000</i>	<i>От 20 001 до 40 000</i>	<i>ФЛ: от 50 001 до 100 000 ЮЛ: от 10 001 до 20 000</i>
XL	<i>Свыше 200 001</i>	<i>Свыше 40 001</i>	<i>ФЛ: свыше 100 001 ЮЛ: свыше 20 001</i>

[1] Единый сервер ДБО, в котором обслуживаются как физические, так и юридические лица

4.5.1 Конфигурация оборудования XXS (для просмотра ознакомительного пакета или пробного использования)

Конфигурация оборудования для метрик:

- Для тестовой эксплуатации или просмотра системы при использовании ознакомительного пакета;
- При использовании сервера ДБО только для обслуживания физических лиц – до 1000 клиентов;
- При использовании сервера ДБО только для обслуживания юридических лиц – до 200 клиентов;
- При использовании единого сервера ДБО для физических и юридических лиц – до 500 клиентов физических лиц и до 100 клиентов юридических лиц.

Сервер	Конфигурация
Сервер СУБД + сервер приложений (единый сервер)	<i>Процессор (ядра): 2 x от 2.2 ГГц Оперативная память: 8 Гб Жесткие диски: SATA RAID1</i>
Веб-сервер распределения нагрузки	<i>Требования не предъявляются</i>

Рабочее место пользователя системы (клиент, сотрудник банка)	<i>Требования не предъявляются</i>
---	------------------------------------

4.5.2 Конфигурация оборудования XS

Конфигурация оборудования для метрик:

- При использовании сервера ДБО только для обслуживания физических лиц – от 1 001 до 10 000 клиентов;
- При использовании сервера ДБО только для обслуживания юридических лиц – от 201 до 2 000 клиентов;
- При использовании единого сервера ДБО для физических и юридических лиц – от 501 до 5 000 клиентов физических лиц и от 101 до 1000 клиентов юридических лиц.

Сервер	Конфигурация
Сервер СУБД + сервер приложений (единый сервер)	<i>Процессор (ядра): 4 x от 2.2 ГГц Оперативная память: 16 Гб Жесткие диски: SATA RAID1</i>
Веб-сервер распределения нагрузки	<i>Требования не предъявляются</i>
Рабочее место пользователя системы (клиент, сотрудник банка)	<i>Требования не предъявляются</i>

4.5.3 Конфигурация оборудования S

Конфигурация оборудования для метрик:

- При использовании сервера ДБО только для обслуживания физических лиц – от 10 001 до 50 000 клиентов;
- При использовании сервера ДБО только для обслуживания юридических лиц – от 1 001 до 10 000 клиентов;
- При использовании единого сервера ДБО для физических и юридических лиц – от 5 001 до 25 000 клиентов физических лиц и от 501 до 5 000 клиентов юридических лиц.

Сервер	Конфигурация
Сервер СУБД	<i>Процессор (ядра): 4 x 2.6 ГГц Оперативная память: 16 Гб Жесткие диски: SAS RAID1 от 150 Гб или SSD</i>
Сервер веб-приложений	<i>Процессор (ядра): 8 x 2.6 ГГц Оперативная память: 32 Гб Жесткие диски: SAS RAID1 150 Гб или SSD</i>
Веб-сервер распределения нагрузки	<i>Требования не предъявляются</i>

Рабочее место пользователя системы (клиент, сотрудник банка)	Требования не предъявляются
---	-----------------------------

4.5.4 Конфигурация оборудования М

Конфигурация оборудования для метрик:

- При использовании сервера ДБО только для обслуживания физических лиц – от 50 001 до 100 000 клиентов;
- При использовании сервера ДБО только для обслуживания юридических лиц – от 10 001 до 20 000 клиентов;
- При использовании единого сервера ДБО для физических и юридических лиц – от 25 001 до 50 000 клиентов физических лиц и от 5 001 до 10 000 клиентов юридических лиц.

Сервер	Конфигурация
Сервер СУБД	<p>Аппаратные требования:</p> <p>Процессор (ядра): 8 x 2.6 ГГц</p> <p>Оперативная память: 32 Гб</p> <p>Жесткие диски: дисковая корзина на 1 Тб SAS или SSD (допускается SAS RAID10 или SSD)</p> <p>Требования к сопутствующему ПО:</p> <p>СУБД: Microsoft SQL Server или Oracle Database</p>
Сервер приложений	<p>Процессор (ядра): 16 x 2.6 ГГц</p> <p>Оперативная память: 64 Гб</p> <p>Жесткие диски: SAS RAID1 150 Гб или SSD</p>
Веб-сервер распределения нагрузки	Требования не предъявляются
Рабочее место пользователя системы (клиент, сотрудник банка)	Требования не предъявляются

4.5.5 Конфигурация оборудования L

Конфигурация оборудования для следующих метрик:

- При использовании сервера ДБО только для обслуживания физических лиц – от 100 001 до 200 000 клиентов;
- При использовании сервера ДБО только для обслуживания юридических лиц – от 20 001 до 40 000 клиентов;
- При использовании единого сервера ДБО для физических и юридических лиц – от 50 001 до 100 000 клиентов физических лиц и от 10 001 до 20 000 клиентов юридических лиц.

Сервер	Конфигурация
Сервер СУБД	Аппаратные требования:

	<p>Процессор (ядра): 16 x 2.8 ГГц</p> <p>Оперативная память: 64 Гб</p> <p>Жесткие диски: дисковая корзина на 2 Тб SAS или SSD (допускается SAS RAID10 или SSD)</p>
	<p>Требования к сопутствующему ПО:</p> <p>СУБД: Oracle Database</p> <p>Операционная система: Linux</p>
Сервер веб-приложений – 2 единицы в программном кластере	<p>Аппаратные требования:</p> <p>Процессор (ядра): 16 x 2.8 ГГц</p> <p>Оперативная память: 64 Гб</p> <p>Жесткие диски: SAS RAID1 300 Гб или SSD</p>
	<p>Требования к сопутствующему ПО:</p> <p>Операционная система: Linux</p>
Веб-сервер распределения нагрузки	<p>Аппаратные требования:</p> <p>Процессор (ядра): 4 x 2.6 ГГц</p> <p>Оперативная память: 8 Гб</p> <p>Жесткие диски: SATA RAID1</p>
	<p>Требования к сопутствующему ПО:</p> <p>Операционная система: Linux</p>
Рабочее место пользователя системы (клиент, сотрудник банка)	Требования не предъявляются

4.5.6 Конфигурация оборудования XL (для большой численности клиентов под индивидуальный расчёт)

Конфигурация оборудования для следующих метрик:

- При использовании сервера ДБО только для обслуживания физических лиц – свыше 200 001 клиента;
- При использовании сервера ДБО только для обслуживания юридических лиц – свыше 40 001 клиента;
- При использовании единого сервера ДБО для физических и юридических лиц – свыше 100 001 клиента физических лиц и свыше 20 001 клиента юридических лиц.

Аппаратная конфигурация определяется индивидуально по запросу в компанию.

4.5.7 Оценка мощностей в зависимости от типа среды

Пример расчета для обслуживания юридических лиц – от 10 001 до 20 000 клиентов.

Тип среды	С	Т	С	С	Р	Р	Р	HDD	Т	П	Периодично	Л
	е	и	С	С	А	А	А	рост	и	о	сть(день/н	и
	р	п	С	С	А	А	А		п	т	еделя/мес)	ц
	в	((А	А	А	А		д	р		е

	е р	Ф и з / в и р т)	к с л - е с я с е р и п)	с с л е з с о д)	с с л е з с о д)	с с л е з с о д)	с с л е з с о д)	в год (Gb)	ис ко в ог о п р ос т ра нс т ва (h i s s d o r 15 k/ m i d 10 k/ o w 7k)	е б н о с т ь в р е з е р в н о м к о п и р о в а н и и (д а/ н е т)		Н з и и н а с и с т е м н о е П О
Пром ышл енна я	С е р в е р п р и л о ж е н и й	ф и з / в и р т	1 6	2 4	6 4	4 4	3 0	Пара метр завис ит от рост а колич естве а доку мент ов, расче т произ води тся по форм уле 10 КБ на доку мент	hi (S A S R A/ D 1 ил и S S D)	д а	день	Л и ц е н з и я з а в и с и т о т в ы б р а н н о

	о ж е н и й С е р в е р Б Д							S S D)	р и е м к и н о в ы х в е р с и й с и с т е м ы)		в и с и т о т в ы б р а н н о й О С / С К З И	M S S Q L / O r a c l e
--	--	--	--	--	--	--	--	--------------	---	--	---	--

4.6 Загрузка дистрибутива системы

Установка программного комплекса ДБО iSimpleRetail выполняется ресурсами системы, входящими в поставку с дистрибутивом системы. Компания iSimpleLab размещает дистрибутивы для развертывания и обновления на FTP-ресурсе компании <ftp://support.isimplelab.com>. Персонализированные дистрибутивы доступны для загрузки под индивидуальным логином и паролем. Используйте путь: /BUILDS/ на FTP-ресурсе после авторизации для поиска соответствующего дистрибутива для установки системы. Логин и пароль к FTP предоставляет Центр сервисов компании <https://service.isimplelab.com> по запросу.

4.7 Установка Java Virtual Machine

Система ДБО iSimpleRetail разработана на языке Java и предполагает исполнение java-кода компилируемой средой Java. В связи с этим обязательным условием, накладываемым на использование системы ДБО iSimpleRetail, является наличие установленной/инициализированной Java-машины, в окружении, где производится исполнение исходного кода системы.

На первом шаге развёртывания следует выполнить установку Java Virtual Machine (см. раздел Требования к сопутствующему программному и аппаратному обеспечению). После установки Java рекомендуется проверить доступность java-машины командой "**java -version**". Если для **Windows** Java устанавливалась инсталляционным пакетом, или в **UNIX/Linux** Java была назначена в переменных средах, то при выполнении команды "**java -version**" отобразится приблизительно следующий текст:

```
java version "1.7.0"  
Java(TM) SE Runtime Environment (build 1.7.0-b147)  
Java HotSpot(TM) Client VM (build 21.0-b17, mixed mode, sharing)
```

Если информационный текст не отобразится, то сервер приложений не сможет выполнить инициализацию Java и нужно или выполнить соответствующее конфигурирование сервера приложений, указав путь к ресурсам Java (см. подраздел Конфигурирование сервера приложений) или восстановить инициализацию Java в переменных средах операционной системы.

5 Процедура установки системы

- Подготовка базы данных для загрузки ресурсов приложения
- Установка и настройка сервера приложений
- Процедура развертывания для веб-сервера Apache Tomcat
- Настройка параметров подключения к базе данных
- Запуск сервера приложений и проверка доступности системы ДБО iSimpleRetail

5.1 Подготовка базы данных для загрузки ресурсов приложения

- Подготовка базы данных
 - Подготовка Microsoft SQL Server
 - Подготовка Oracle Database
 - Подготовка PostgreSQL
 - Подготовка Firebird
- Настройка конфигурационного файла для создания структуры базы данных
 - Параметры соединения с базой данных Microsoft SQL
 - Параметры соединения с базой данных Oracle Database
 - Параметры соединения с базой данных PostgreSQL
 - Параметры соединения с базой данных Firebird
- Создание структуры базы данных
 - Oracle Database. Создания индексов на внешних ключах

Система ДБО iSimpleRetail реализует абстрактный уровень работы с ресурсами баз данных, что позволяет использовать все распространенные промышленные серверы баз данных, используя единые структурированные ресурсы системы.

Взаимодействие приложения с базой данных реализуется средствами JDBC драйвера, соответствующего типу сервера базы данных. Все необходимые JDBC драйверы включены в поставку дистрибутива с системой.

Для загрузки в базу данных ресурсов приложения необходимо выполнить первоначальные процедуры подготовки базы данных, рассмотренные в настоящем разделе:

- Подготовка (создание) базы данных для создания структуры (см. подраздел Подготовка базы данных для загрузки ресурсов приложения).
- Настройка конфигурационного файла для подключения к базе данных (см. подраздел Настройка конфигурационного файла для создания структуры базы данных).
- Создание структуры базы данных сервисными скриптами (см. подраздел Создание структуры базы данных).

5.1.1 Подготовка базы данных

Процедура подготовки базы данных предполагает создание рабочей базы данных системы, в которой будут храниться данные и объекты системы. В последствии в созданной базе при помощи сервисных скриптов будет создаваться структура системы ДБО iSimpleRetail. Порядок подготовки базы данных описан в соответствии с выбранным типом базы данных.

- Подготовка Microsoft SQL Server
- Подготовка Oracle Database
- Подготовка PostgreSQL
- Подготовка Firebird

5.1.1.1 Подготовка Microsoft SQL Server

Настоящая документация не содержит описания установки и процесса конфигурирования сервера баз данных Microsoft SQL Server. Используйте документацию к серверу баз данных для детальных настроек Microsoft SQL Server и настроек производительности баз данных, поставляемую Microsoft.

Подготовка Microsoft SQL Server выполняется в следующей последовательности:

- Регистрация/создание пользователя Microsoft SQL Server, при помощи которого приложение будет авторизоваться к базе данных.
- Создание базы данных, в которую приложение установит ресурсы и будет использовать в работе.

5.1.1.1.1 Регистрация/создание пользователя в Microsoft SQL Server

Создайте для развертываемого приложения в Microsoft SQL Server нового пользователя или используйте существующего.

Обязательным условием при создании пользователя в Microsoft SQL Server для работы приложения является поддержка пользователем метода аутентификации SQL Server Authentication.

5.1.1.1.2 Создание базы данных Microsoft SQL Server

- Создайте новую базу данных Microsoft SQL Server средствами Microsoft SQL Server в кодировке **Cyrillic_General_CI_AS**.
- Назначьте роль **db_owner** на созданную базу данных для пользователя, через которого приложение будет авторизоваться в Microsoft SQL Server.



Внимание! Для уменьшения нагрузки на жесткий диск при работе системы ДБО с БД MS SQL (версии Enterprise), необходимо отключить SQL Server Reporting Services. Для построения отчетности из БД достаточно использовать Analysis Services, который не вызывает дополнительной нагрузки на БД. Для увеличения производительности БД необходимо ограничивать максимальное выделение ОЗУ для инстанса (экземпляра) в размере 70-80% от общей оперативной памяти сервера (рекомендация используется в случае расположения БД на выделенном сервере).

5.1.1.2 Подготовка Oracle Database

Настоящая документация не содержит описания установки и процесса конфигурирования сервера баз данных Oracle Database. Используйте документацию к серверу баз данных для детальных настроек Oracle Database.

Подготовка Oracle Database включает регистрацию пользователя Oracle Database с правами создавать и изменять таблицы.

Если Oracle 12+ под учетными записями sys или system выполнить:

```
alter session set "_ORACLE_SCRIPT"=true;
```

```
----- CREATE USER [пользователь] IDENTIFIED BY [пароль] -----
----- DEFAULT TABLESPACE [tablespace] QUOTA целое число [K][M] ON [tablespace]
-----
----- TEMPORARY TABLESPACE [tablespace] QUOTA целое число [K][M] ON
```

```
[tablespace] --  
----- ALTER USER [пользователь] quota unlimited on users; -----
```



Внимание! При создании базы данных (Instance) должна определяться кодировка **Win1251**

5.1.1.2.1 Минимальный набор ролей пользователя Oracle Database

Для корректной работы системы создаваемому пользователю Oracle Database должен быть предоставлен следующий минимальный набор ролей и привилегий:

```
GRANT CREATE TABLE TO [пользователь];  
GRANT CREATE PROCEDURE TO [пользователь];  
GRANT CREATE TRIGGER TO [пользователь];  
GRANT CREATE VIEW TO [пользователь];  
GRANT CREATE SEQUENCE TO [пользователь];  
GRANT CREATE VIEW TO [пользователь];  
GRANT DELETE ANY TABLE TO [пользователь];  
GRANT DROP ANY TABLE TO [пользователь];  
GRANT DROP ANY PROCEDURE TO [пользователь];  
GRANT DROP ANY TRIGGER TO [пользователь];  
GRANT DROP ANY VIEW TO [пользователь];  
GRANT ALTER ANY TABLE TO [пользователь];  
GRANT ALTER ANY TABLE TO [пользователь];  
GRANT ALTER ANY PROCEDURE TO [пользователь];  
GRANT ALTER ANY TRIGGER TO [пользователь];
```

5.1.1.3 Подготовка PostgreSQL

Настоящая документация не содержит описания установки и процесса конфигурирования сервера баз данных PostgreSQL. Используйте документацию к серверу баз данных для детальных настроек PostgreSQL и настроек производительности баз данных, поставляемую разработчиком PostgreSQL.

Подготовка PostgreSQL выполняется в следующей последовательности:

- Регистрация пользователя PostgreSQL, при помощи которого приложение будет авторизоваться к базе данных.
- Создание базы данных, в которую приложение установит ресурсы и будет использовать в работе. При создании базы данных производится привязка базы данных к созданному ранее пользователю PostgreSQL, под которым система ДБО iSimpleRetail будет обращаться к рабочей базе данных.

5.1.1.3.1 Регистрация пользователя в PostgreSQL

Создайте для развертываемого приложения в PostgreSQL нового пользователя средствами PostgreSQL или используйте существующего пользователя.

5.1.1.3.2 Создание базы данных PostgreSQL

- Создайте новую базу данных PostgreSQL средствами PostgreSQL в кодировке **UTF-8**.
- Сделайте привязку пользователя PostgreSQL с созданной базой данных PostgreSQL.

5.1.1.4 Подготовка Firebird

Firebird – свободная база данных с открытым исходным кодом, официальный сайт сообщества Firebird www.firebirdsql.org.

Настоящая документация не содержит описания установки и последующего конфигурирования Firebird. Используйте документацию к серверу баз данных для

детальных настроек Firebird и настроек производительности баз данных, поставляемую разработчиком Firebird.

Для базы данных Firebird компания iSimpleLab предоставляет готовые инструментариумы для создания базы данных, которые можно использовать наравне со штатными средствами создания базы данных Firebird. Средства создания базы данных Firebird включены в установочный пакет системы ДБО iSimpleRetail.

Далее описывается процесс создания базы данных Firebird инструментами, предоставляемыми компанией iSimpleLab. Для создания базы данных Firebird средствами Firebird воспользуйтесь соответствующей документацией, предоставляемой разработчиком Firebird.

Подготовка базы данных Firebird средствами инструментов iSimpleRetail выполняется в следующей последовательности:

5.1.1.4.1 Создание базы данных Firebird

После выполнения настроек соединения с базой данных в конфигурационном файле `ups.properties` (см. подраздел Настройка конфигурационного файла для создания структуры базы данных) - создайте базу данных при помощи сервисной утилиты **create_base**, предоставляемой iSimpleLab.

Для создания базы данных Firebird выполните исполняемый файл **create_base.cmd|sh** из каталога `\create_db\stends\system\sql\`, при этом:

- в каталоге `\create_db\stends\` должен находиться актуальный файл с настройками для создания базы данных **ups.properties**;
- должны быть созданы каталоги, указанные в параметре **[ups.db.url]** файла `ups.properties` (см. пункт Параметры соединения с базой данных Firebird).

При запуске исполняемого файла **create_base** на экране консольного режима запуска отобразится процесс создания файла базы данных.

Контролируйте процесс правильности создания базы данных. При успешном создании базы данных в консольном режиме отобразится сообщение – **Create SUCCEEDED**. По пути **[ups.db.url]** из секции параметров создания базы данных конфигурационного файла **ups.properties** будет создан файл базы данных. В случае возникновения ошибки в создании базы, на экране консольного режима исполнения утилиты **create_base** появится сообщение **Create FAILED** и результат ошибки зафиксируется в файл протокола.

Протокол работы исполнения **create_base** записывается в лог-файл `\create_db\stends\system\sql\ups.log`

5.1.2 Настройка конфигурационного файла для создания структуры базы данных

На этом шаге установки необходимо отредактировать конфигурационный файл для соединения пакета установки с базой данных.

Конфигурационный файл для модификации располагается в разархивированном пакете установки, действие по распаковке которого определено в пункте Подготовка пакета установки для использования.

Путь размещения файла: **<пакет установки>\create_db(.zip)\stends\ups.properties**.

Настройка конфигурационного файла должна производиться в зависимости от типа базы данных, используемой в банке. Необходимо раскомментировать (убрать символ #) у той секции с параметрами, которая соответствует используемому типу сервера базы данных. Для других типов подключений параметры должны быть закомментированы с использованием символа #.

Описание параметров конфигурирования файла `ups.properties` для соответствующих типов СУБД приводится в разделах:

- Параметры соединения с базой данных Microsoft SQL

- Параметры соединения с базой данных Oracle Database
- Параметры соединения с базой данных PostgreSQL
- Параметры соединения с базой данных Firebird

5.1.2.1 Параметры соединения с базой данных Microsoft SQL

Формат назначения параметров соединения с базой данных Microsoft SQL Server для установки приложения:

```
ups.db.accessmode=mssql
ups.db.driver=com.microsoft.sqlserver.jdbc.SQLServerDriver
ups.db.url=jdbc:sqlserver://[host]:1433;databaseName=[database];selectMethod=cursor;sendStringParametersAsUnicode=false
ups.db.user=[username]
ups.db.password=[password]
```

Параметры соединения с базой данных Microsoft SQL Server в файле **ups.properties** приведены в таблице:

Переменная	Описание
<i>[host]</i>	<i>Имя или IP-адрес сервера баз данных Microsoft SQL Server. При размещении базы данных на сервере, где используется приложение, допускается указывать переменные localhost или 127.0.0.1.</i>
<i>[port]</i>	<i>Порт соединения с базой данных. Порт по умолчанию 1443</i>
<i>[database]</i>	<i>База данных для работы приложения</i>
<i>[username]</i>	<i>Пользователь Microsoft SQL Server, которым будет выполняться аутентификация</i>
<i>[password]</i>	<i>Пароль пользователя Microsoft SQL Server</i>

Пример заполненной секции соединения с базой данных Microsoft SQL Server:

```
ups.db.accessmode=mssql
ups.db.driver=com.microsoft.sqlserver.jdbc.SQLServerDriver
ups.db.url=jdbc:sqlserver://[host]:1433;databaseName=[database];selectMethod=cursor;sendStringParametersAsUnicode=false
ups.db.user=dblogin
ups.db.password=dbpassword
```

5.1.2.2 Параметры соединения с базой данных Oracle Database

Формат назначения параметров соединения с базой данных Oracle Database для установки приложения:

```
ups.db.accessmode=oracle
ups.db.driver=oracle.jdbc.driver.OracleDriver
ups.db.url=jdbc:oracle:thin:@localhost:1521:ORCL
ups.db.user=[username]
ups.db.password=[password]
ups.db.sysuser=sys AS SYSDBA
ups.db.syspassword=sys
```

Параметры соединения с базой данных Oracle Database в файле **ups.properties** приведены в таблице:

Переменная	Описание
<code>[username]</code>	Пользователь Oracle Database, которым будет выполняться аутентификация к базе данных приложения
<code>[password]</code>	Пароль пользователя Oracle Database
<code>[sysuser]</code>	Администратор Oracle Database, которым будет выполняться аутентификация к базе данных приложения
<code>[syspassword]</code>	Пароль администратора Oracle Database
<code>[host]</code>	Имя или IP-адрес сервера баз данных Oracle Database
<code>[port]</code>	Порт соединения с базой данных. Порт по умолчанию 1521
<code>[instance]</code>	База данных для работы приложения

Пример заполненной секции соединения с базой данных Oracle Database:

```
ups.db.accessmode=oracle
ups.db.driver=oracle.jdbc.driver.OracleDriver
ups.db.url=jdbc:oracle:thin:@localhost:1521:ORCL
ups.db.user=system
ups.db.password=open
ups.db.sysuser=sys AS SYSDBA
ups.db.syspassword=[password]
```

5.1.2.3 Параметры соединения с базой данных PostgreSQL

Формат назначения параметров соединения с базой данных PostgreSQL для установки приложения:

```
ups.db.accessmode=postgres
ups.db.driver=org.postgresql.Driver
ups.db.url=jdbc:postgresql://[host]:[port]/[database]
ups.db.user=[username]
ups.db.password=[password]
```

Параметры соединения с базой данных PostgreSQL в файле `ups.properties` приведены в таблице:

Переменная	Описание
<code>[host]</code>	Имя или IP адрес сервера баз данных PostgreSQL
<code>[port]</code>	Порт соединения с базой данных. Порт по умолчанию 5432
<code>[database]</code>	База данных для работы приложения
<code>[username]</code>	Пользователь PostgreSQL, которым будет выполняться аутентификация
<code>[password]</code>	Пароль пользователя PostgreSQL

Пример заполненной секции соединения с базой данных PostgreSQL:

```
ups.db.accessmode=postgres
ups.db.driver=org.postgresql.Driver
```

```
ups.db.url=jdbc:postgresql://dbservername:5432/dbname
ups.db.user=dblogin
ups.db.password=dbpassword
```

5.1.2.4 Параметры соединения с базой данных Firebird

Формат назначения параметров соединения с базой данных Firebird для установки приложения:

```
ups.accessmode=interbase
ups.db.driver=org.firebirdsql.jdbc.FBDriver
ups.db.user=[username]
ups.db.password=[password]
ups.db.url=jdbc:firebirdsql:[host]/[port]:[database file]
```

Параметры соединения с базой данных Firebird в файле **ups.properties** приведены в таблице:

Переменная	Описание
<i>[username]</i>	Имя пользователя Firebird, под которым приложение будет авторизоваться в базу данных. Имя пользователя по умолчанию SYSDBA
<i>[password]</i>	Пароль для пользователя, указанного в параметре <i>[ups.db.user]</i> . Пароль по умолчанию masterkey
<i>[ups.db.url]</i>	Параметры соединения с сервером Firebird и путь размещения файла базы данных

Пример заполненной секции соединения с базой данных Firebird:

```
ups.accessmode=interbase
ups.db.driver=org.firebirdsql.jdbc.FBDriver
ups.db.user=SYSDBA
ups.db.password=masterkey
ups.db.url=jdbc:firebirdsql:dbservername/3050:C:/appname/db/dbname.fdb
```

5.1.3 Создание структуры базы данных

После определения настроек параметров подключения к базе данных следует приступить к непосредственной процедуре создания структуры базы данных. Допускается выполнять действие по созданию рабочей базы данных с произвольной рабочей станции, но при этом на рабочей станции, где исполняется утилита создания структуры базы данных (`perform.cmd|sh`), должна быть инициализирована Java-машина (см. подраздел Установка Java Virtual Machine).




Используйте разархивированный пакет установки, действие по распаковке которого определено в пункте Подготовка пакета установки для использования.

Для формирования структуры базы данных следует выполнить файл **perform.cmd|sh** из директории **create_db\stends\system\sql** и далее последовательно исполнять сервисный файл **update.cmd|sh** из каждого каталога конфигураций, которые расположены по пути: **\create_db\stends\<каталог со скриптами>\sql** распакованного пакета установки.

Таким образом последовательность запуска исполняемых файлов из каталогов должна быть следующей:



Внимание! Необходимо строго придерживаться приведенной последовательности запуска исполняемых файлов!

1. Конфигурация **system** (конфигурация ядра системы) – файл **perform.cmd|sh** из каталога **system**.
2. Конфигурация **qrtz** (конфигурация планировщика системы) – файл **update.cmd|sh** из каталога **qrtz**.
3. Конфигурация **doc** (системная конфигурация документооборота) – файл **update.cmd|sh** из каталога **doc**.
4. Конфигурация **ibankfl** (прикладная конфигурация системы ДБО *iSimpleRetail* для физических лиц) – файл **update.cmd|sh** из каталога **ibankfl**.
5. Конфигурация **ibank_core** (ядровая конфигурация системы ДБО *iSimpleRetail*) – файл **update.cmd|sh** из каталога **ibank_core**.
6. Конфигурация **ibankul** (прикладная конфигурация системы ДБО *iSimpleRetail* для юридических лиц) – файл **update.cmd|sh** из каталога **ibankul**. Должна выполняться в случае ее наличия.
7. Конфигурация **1c-direct** (конфигурация для взаимодействия с **1c-direct**) – файл **update.cmd|sh** из каталога **1c-direct**. Должна выполняться в случае ее наличия.
8. Конфигурация **notify-isimplemsg**  (конфигурация для взаимодействия с **iSimpleMessageGate**) - из каталога **notify-isimplemsg**. Должна выполняться в случае ее наличия.
9. Конфигурация **invoice_gis_gmp**  (прикладная конфигурация системы ДБО *iSimpleRetail*) – файл **update.cmd|sh** из каталога **invoice_gis_gmp**. Должна выполняться в случае ее наличия.
10. Конфигурация **ibankul_curr**  (прикладная конфигурация системы ДБО *iSimpleRetail*) – файл **update.cmd|sh** из каталога **ibankul_curr**. Должна выполняться в случае ее наличия.
11. Конфигурация **custom_folder** (конфигурация модуля кастомизации) – файл **update.cmd|sh** из каталога **custom_folder**. Должна выполняться в случае ее наличия.
12. Конфигурация **kladr** (конфигурация справочника КЛАДР) – файл **update.cmd|sh** из каталога **kladr**. Создание структуры базы данных импорта справочника КЛАДР не является обязательной для работы системы.
13. Конфигурация **sbp** (конфигурация системы быстрых платежей) - файл **update.cmd|sh** из каталога **sbr**. Должна выполняться в случае ее наличия.
14. Конфигурация **sbp_emulator** (конфигурация эмулятора системы быстрых платежей) - файл **update.cmd|sh** из каталога **sbp_emulator**. Должна выполняться в случае ее наличия.
15. Конфигурация **pfm-ibankfl** (конфигурация сервиса Персональный финансовый менеджер) - файл **update.cmd|sh** из каталога **pfm-ibankfl**. Должна выполняться в случае ее наличия.
16. Конфигурация **best2pay** (конфигурация сервиса Перевод с карты на карту через партнера Best2Pay) - файл **update.cmd|sh** из каталога **best2pay**. Должна выполняться в случае ее наличия.

17. Конфигурация **emulator** (конфигурация поставляемых эмуляторов АБС, ПЦ, ПС) – файл **update.cmd|sh** из каталога **emulator**. Должна выполняться в случае необходимости настройки эмулятора работы системы и является необязательным действием.

- a. Конфигурация **abs_emulator** (конфигурация поставляемого эмуляторов АБС) – из каталога **abs_emulator**. Должна выполняться в случае необходимости настройки эмулятора АБС и является необязательным действием.
- b. Конфигурация **custom_emulator** (конфигурация поставляемых эмуляторов АБС, ПЦ, ПС) – из каталога **custom_emulator**. Должна выполняться в случае необходимости настройки эмуляторов и является необязательным действием.

При запуске обновления конфигурации (исполнение файла **update.cmd|sh**) на экране консольного режима отобразится процесс исполнения скриптов.

При выполнении скриптов отобразится вопрос о согласии на добавление нового скрипта, на который следует утвердительно ответить – ввести с клавиатуры символ «Y».

В процессе исполнения SQL-скриптов необходимо контролировать успешность их применения. При успешном наполнении базы данных в консольном режиме должно формироваться сообщение «Update SUCCEEDED», при выполнении с ошибкой «Update FAILED».

Протокол исполнения скриптов можно детально просмотреть в лог-файле **\create_db\stends\<наименование конфигурации>\sql\lups.log**.



Конфигурация **ibankul_curr** добавлена начиная с версий **2.0.15.644u39**, **2.0.16.671**.



Конфигурация **invoice_gis_gmp** добавлена начиная с версии **2.0.16.676**.



Конфигурации **ibank_core** и **ibankul** добавлены начиная с версий **cust-ibank-core-0.28.0**, **cust-ibank-ul-0.27.0**, **cust-ibank-ul-currency-0.15.0**.



Конфигурация **notify-isimplemsg** добавлена начиная с версии **2.0.16.684**.



Конфигурация **1c-direct** добавлена начиная с версии **ibank-fl-standart-3.1.0**.



Конфигурация **abs_emulator** добавлена начиная с версии **ibank-ul-standart-4.1.0**.



Конфигурация **best2pay** добавлена начиная с версии **ibank-ul-standart-4.1.0**.

5.1.3.1 Oracle Database. Создания индексов на внешних ключах

5.2 Установка и настройка сервера приложений

В данном разделе описываются процедуры установки и настройки сервера приложений.

В настоящем руководстве приводится описание процедуры установки и настройки для сервера приложений Apache Tomcat. Получить рекомендации по установке и настройке других серверов приложений, с которыми сертифицирована система ДБО iSimpleRetail (см. подраздел Требования к сопутствующему программному и аппаратному обеспечению), можно обратившись в компанию iSimpleLab.

- Развертывание и настройка Apache Tomcat

5.2.1 Развертывание и настройка Apache Tomcat

- Конфигурационные файлы:
- Запуск приложения:

Используется версия Apache-Tomcat 7.0.82.

Java 1.7 Oracle (Для Linux ОС rpm во вложении).

5.2.1.1 Конфигурационные файлы:

apache-tomcat\conf\server.xml

Указываем порты http/https/ajp по необходимости использования.

Если AJP не используется блоки с AJP необходимо закомментировать:

```
<Connector port="8080" protocol="HTTP/1.1"
    connectionTimeout="20000"
    maxThreads="1000"
    redirectPort="8443" />

<Connector port="8109" protocol="AJP/1.3" redirectPort="8443" />
```

IP адрес или hostname сервера:

```
<Engine name="Catalina" defaultHost="192.168.10.236">

<Host name="192.168.10.236" appBase="webapps"
    unpackWARs="true" autoDeploy="true">
```

Для запрета на отображение версии Apache Tomcat необходимо в конфигурационный файл server.xml между тегами <Host> и </Host> добавить следующую строку с настройками:

```
<Valve className="org.apache.catalina.valves.ErrorReportValve"
    showReport="false" showServerInfo="false"/>
```

Пример блока Host в этом случае:

```
<Host name="192.168.10.236" appBase="webapps"
    unpackWARs="true" autoDeploy="true">
    <Valve className="org.apache.catalina.valves.AccessLogValve"
        directory="logs"
        prefix="localhost_access_log." suffix=".txt"
        pattern="%h %l %u %t &quot;%r&quot; %s %b" />
    <Valve className="org.apache.catalina.valves.ErrorReportValve"
        showReport="false" showServerInfo="false"/>
</Host>
```

Отключаем SSLEngine, закомментировав блок (т.к. SSL настраивается на прокси-сервере, не на веб-сервере):

```
<Listener className="org.apache.catalina.core.AprLifecycleListener"
    SSLEngine="on" />
```

apache-tomcat\bin\catalina

Все описанные для standalone.conf в разделе Конфигурирование сервера приложений также выполняются и для конфигурационного файла catalina

War-контейнер приложения необходимо распаковать в папку \apache-tomcat\webapps\ROOT



В целях безопасности каталоги docs, examples, host-manager, manager подлежат удалению.

Имя	Дата изменения	Тип	Размер
docs	01.03.2018 13:22	Папка с файлами	
examples	01.03.2018 13:23	Папка с файлами	
host-manager	01.03.2018 13:23	Папка с файлами	
manager	01.03.2018 13:23	Папка с файлами	
ROOT	01.03.2018 13:35	Папка с файлами	

5.2.1.2 Запуск приложения:

Запуск на Linux ОС производится командой с ключом на выбор.

Ключ **run** отобразит в текущей консоли детали запуска (catalina.log) и привяжет к вашей сессии процесс:

```
[root@localhost bin]# sh catalina.sh run
```

Ключ **start** запустит независимый процесс web сервиса:

```
[root@localhost bin]# sh catalina.sh start
```

Запуск на Windows ОС

Для запуска лучше создать в директории **bin** отдельный **bat** файл. Назовём его startserver.bat.

В файле укажем путь к необходимой Java и ключ запуска:

```
set "JAVA_HOME=C:\Program Files\Java\jre7"  
catalina.bat start
```

5.3 Процедура развертывания для веб-сервера Apache Tomcat

5.3.1 Развертывание J2EE-контейнера приложения

Процедура установки файлов контейнера с системой отличается для различных серверов приложений

В следующих подразделах описываются процедуры развертывания Java-контейнера приложения.

5.4 Настройка параметров подключения к базе данных

Существует два способа настройки соединения с СУБД в WEB приложении

1. JDBC. Указание параметров подключения в приложении, в файле **spring_datasources.xml**.
2. JNDI. Указание параметров подключения в WEB-сервере и указание названия этого подключения в файле **spring_datasources.xml**.
 - Подключение к базе данных Microsoft SQL Server
 - Подключение к базе данных Oracle Database
 - Подключение к базе данных PostgreSQL
 - Подключение к базе данных Firebird
 - Подключение к базе данных JNDI

После формирования структуры базы данных (см. пункт Создание структуры базы данных) и развертывания J2EE-контейнера приложения (см. пункт Развертывание J2EE-контейнера приложения) необходимо настроить подключение к базе данных. Для корректного подключения к базе данных следует определить параметры подключения в конфигурационном файле J2EE-контейнера, расположенном по адресу: **<каталог развёртывания JBoss AS>\standalone\deployments\ROOT.war\WEB-INF\spring_datasources.xml**.

5.4.1 Настройка типа используемой базы данных

Определите используемый Java-контейнером тип источника подключения сервера базы данных и настроек безопасности в секции настроек файла **.WEB-INF\spring_datasources.xml**:

```
<bean name="dataSourceDefaultTpl" abstract="true"
parent="<зарегистрированный идентификатор подключения к базе данных>" />
```

где параметр **<зарегистрированный идентификатор подключения к базе данных>** принимает значения для различных серверов баз данных:

dataSourceMSSQLTpl – для сервера базы данных Microsoft SQL Server;

dataSourceOracleTpl – для сервера базы данных Oracle Database;

dataSourcePostgresTpl – для сервера базы данных PostgreSQL;

dataSourceFirebirdTpl – для сервера базы данных Firebird и RED Database.

dataSourceJndi – для сервера базы данных, описатель подключений которой задан на WEB-сервере.

Пример определения источника базы данных для сервера базы данных Microsoft SQL Server:

```
<bean name="dataSourceDefaultTpl" abstract="true"
parent="dataSourceMSSQLTpl" />
```

Начиная с версии **2.0.14.628** добавлены вспомогательные настройки пула соединений в виде параметров (*additional*) в настройках **spring_datasources.xml**:

```
<bean name="dataSourcePostgresTpl" abstract="true">
<property name="driverClassName" value="org.postgresql.Driver" />
<property name="url" value="jdbc:postgresql://127.0.0.1:5433/ibank" />
<property name="username" value="postgres" />
<property name="password" value="123456" />
<property name="poolSize" value="100" />
<property name="additional">
<map>
<entry key="maxConnLifetimeMillis" value="-1"/>
<entry key="initialSize" value="0"/>
<entry key="minIdle" value="1"/>
<entry key="maxWaitMillis" value="-1"/>
<entry key="validationQuery" value="select 1"/>
<entry key="testOnCreate" value="false"/>
<entry key="testOnBorrow" value="true"/>
<entry key="testOnReturn" value="false"/>
<entry key="testWhileIdle" value="true"/>
<entry key="lifo" value="true"/>
<entry key="poolPreparedStatements" value="true"/>
<entry key="maxOpenPreparedStatements" value="5"/>
<entry key="connectionInitSqls" value=""/>
<entry key="disconnectionSqlCodes" value=""/>
<entry key="fastFailValidation" value="true"/>
</map>
</property>
</bean>
```

Описание параметров*: <https://commons.apache.org/proper/commons-dbcp/configuration.html>

*Поддерживаются только параметры из указанного примера.

5.4.2 Подключение к базе данных Microsoft SQL Server

Секция параметров соединения с базой данных Microsoft SQL Server конфигурационного файла **spring_datasources.xml**:

```
<bean name="dataSourceMSSQLTpl" abstract="true">
<property name="driverClassName"
value="com.microsoft.sqlserver.jdbc.SQLServerDriver" />
<property name="url"
value="jdbc:sqlserver://[host]:[port];databaseName=[database];selectMetho
d=cursor;sendStringParametersAsUnicode=false" />
<property name="username" value="[username]" />
<property name="password" value="[password]" />
</bean>
```

Описание переменных параметров подключения к базе данных Microsoft SQL Server:

Переменная	Описание
[username]	Пользователь Microsoft SQL Server, которым будет выполняться аутентификация к базе данных приложения
[password]	Пароль пользователя Microsoft SQL Server
[host]	Имя или IP-адрес сервера баз данных Microsoft SQL Server
[port]	Порт соединения с базой данных. Порт по умолчанию 1443.

<i>[database]</i>	<i>База данных для работы приложения</i>
-------------------	--

Пример заполненной секции соединения с базой данных Microsoft SQL Server конфигурационного файла **spring_datasources.xml**:

```
<bean name="dataSourceMSSQLTpl" abstract="true">
<property name="driverClassName"
value="com.microsoft.sqlserver.jdbc.SQLServerDriver" />
<property name="url" value="jdbc:sqlserver://
DBServerName:1433;databaseName=dbservername;selectMethod=cursor;sendStringParametersAsUnicode=false" />
<property name="username" value="dblogin" />
<property name="password" value="dbpassword" />
</bean>
```

5.4.3 Подключение к базе данных Oracle Database

Секция параметров соединения с базой данных Oracle Database конфигурационного файла **spring_datasources.xml**:

```
<bean name="dataSourceOracleTpl" abstract="true">
<property name="driverClassName" value="oracle.jdbc.driver.OracleDriver" />
<property name="url" value="jdbc:oracle:thin:@[host]:1521:[instance]" />
<property name="username" value="[username]" />
<property name="password" value="[password]" />
<!-- <property name="poolSize" value="100" />-->
</bean>
```

Описание переменных параметров подключения к базе данных Oracle Database:

Переменная	Описание
<i>[username]</i>	<i>Пользователь Oracle Database, которым будет выполняться аутентификация к базе данных приложения</i>
<i>[password]</i>	<i>Пароль пользователя Oracle Database</i>
<i>[host]</i>	<i>Имя или IP-адрес сервера баз данных Oracle Database</i>
<i>[port]</i>	<i>Порт соединения с базой данных. Порт по умолчанию 1443.</i>
<i>[instance]</i>	<i>База данных для работы приложения</i>

Пример заполненной секции соединения с базой данных Oracle Database конфигурационного файла **spring_datasources.xml**:

```
<bean name="dataSourceOracleTpl" abstract="true">
<property name="driverClassName" value="oracle.jdbc.driver.OracleDriver" />
<property name="url" value="jdbc:oracle:thin:@[host]:1521:[instance]" />
<property name="username" value="system" />
<property name="password" value="open" />
<!-- <property name="poolSize" value="100" />-->
</bean>
```

5.4.4 Подключение к базе данных PostgreSQL

Секция параметров соединения с базой данных PostgreSQL конфигурационного файла `spring_datasources.xml`:

```
<bean name="dataSourcePostgresTpl" abstract="true">
  <property name="driverClassName" value="org.postgresql.Driver" />
  <property name="url"
    value="jdbc:postgresql://[host]:[port]/[database]" />
  <property name="username" value="[username]" />
  <property name="password" value="[password]" />
</bean>
```

Описание переменных параметров соединения с базой данных PostgreSQL:

Переменная	Описание
<code>[username]</code>	Пользователь PostgreSQL, которым будет выполняться аутентификация к базе данных приложения
<code>[password]</code>	Пароль пользователя PostgreSQL
<code>[host]</code>	Имя или IP-адрес сервера баз данных PostgreSQL
<code>[port]</code>	Порт соединения с базой данных. Порт по умолчанию 5432.
<code>[database]</code>	База данных для работы приложения

Пример заполненной секции соединения с базой данных PostgreSQL конфигурационного файла `spring_datasources.xml`:

```
<bean name="dataSourcePostgresTpl" abstract="true">
  <property name="driverClassName"
    value="com.microsoft.sqlserver.jdbc.SQLServerDriver" />
  <property name="url"
    value="jdbc:postgresql://dbservername:5432/dbname" />
  <property name="username" value="dblogin" />
  <property name="password" value="dbpassword" />
</bean>
```

5.4.5 Подключение к базе данных Firebird

Секция параметров соединения с базой данных Firebird конфигурационного файла `spring_datasources.xml`:

```
<bean name="dataSourceFirebirdTpl" abstract="true">
  <property name="driverClassName" value="org.firebirdsql.jdbc.FBDriver" />
  <property name="url"
    value="jdbc:firebirdsql://[host]:[port]/[database file]" />
  <property name="username" value="[username]" />
  <property name="password" value="[password]" />
  <!-- <property name="poolSize" value="100" />-->
</bean>
```

Описание переменных параметров соединения с базой данных Firebird:

Переменная	Описание
------------	----------

<i>[host]</i>	<i>IP адрес или имя сервера, где размещается база данных приложения. При размещении базы данных на сервере, где используется приложение, допускается указывать переменные localhost или 127.0.0.1. При использовании выделенного сервера баз данных обязательным условием возможности взаимодействия приложения с базой данных является наличие установленных драйверов базы данных на сервере с приложением.</i>
<i>[port]</i>	<i>Порт соединения с базой данных. Порт по умолчанию 3050.</i>
<i>[database file]</i>	<i>Имя файла базы данных. Например: C:/appname/db/dbname.fdb. Для операционных систем Windows и UNIX/Linux разделителем путей в указании пути к базе данных служит символ "/"</i>
<i>[username]</i>	<i>Имя пользователя Firebird, под которым приложение будет авторизоваться в базу данных. Имя пользователя по умолчанию SYSDBA.</i>
<i>[password]</i>	<i>Пароль для пользователя, указанного в параметре [ups.db.user]. Пароль по умолчанию masterkey.</i>

Пример секции параметров соединения с базой данных Firebird конфигурационного файла **spring_datasources.xml**:

```
<bean name="dataSourceFirebirdTpl" abstract="true">
  <property name="driverClassName" value="org.firebirdsql.jdbc.FBDriver" />
</>
  <property name="url"
value="jdbc:firebirdsql://dbservername:3050/C:/appname/db/dbname.fdb" />
  <property name="username" value="SYSDBA" />
  <property name="password" value="masterkey" />
  <!-- <property name="poolSize" value="100" />-->
</bean>
```

5.4.6 Подключение к базе данных JNDI

Подключение к базе данных JNDI описано в конфигурационном файле сервера приложений.

Секция параметров соединения с базой данных по описателю JNDI конфигурационного файла **spring_datasources.xml**:

```
<bean name="dataSourceJndi" abstract="true">
<property name="jndiName" value="java:/comp/env/jdbc/iSimpleBank"/>
</bean>
```

Создание JNDI описателя соединения с БД осуществляется в файловых настройках сервера приложений или в интерфейсе администратора сервера приложений.

5.4.6.1 Настройка JNDI подключения в Apache Tomcat APACHE_TOMCAT\conf\context.xml

```
<Context>
...
<Resource name="jdbc/iSimpleBank"
auth="Container"
type="javax.sql.DataSource"
driverClassName="org.postgresql.Driver"
url="jdbc:postgresql://127.0.0.1:5432/iSimplebank"
username="postgres"
password="PASSWORD"
initialSize="0"
maxTotal="200"
maxIdle="200"
```

```

minIdle="0"
maxWaitMillis="600000"
validationQuery="select 1"
testOnBorrow="true"
defaultAutoCommit="false"
defaultTransactionIsolation="READ_COMMITTED"
poolPreparedStatements="true"/>
...
</Context>

```

В каталог `APACHE_TOMCAT\lib` необходимо скопировать используемый JDBC драйвер (входящий в поставку системы) для работы с СУБД.

Название файла	Тип базы данных
<code>postgresql-9.0-801.jdbc4.jar</code>	<i>PostgreSQL</i>
<code>ojdbc14.jar</code>	<i>Oracle</i>
<code>sqljdbc4.jar</code>	<i>MSSQL</i>
<code>jaybird-full-2.1.6.jar</code>	<i>Firebird</i>

5.4.6.2 Настройка JNDI подключения в JBoss AS

Выполняется в файле `standalone.xml` <каталог развёртывания JBoss AS>`standalone\configuration\`

```

..
<subsystem xmlns="urn:jboss:domain:datasources:1.0">
<datasources>
<datasource jta="false" jndi-name="java:/comp/env/jdbc/iSimpleBank"
pool-name="iSimpleBank" enabled="true" use-java-context="true" use-
ccm="false">
<connection-
url>jdbc:postgresql://127.0.0.1:5432/iSimpleBank</connection-url>
<driver-class>org.postgresql.Driver</driver-class>
<driver>postgresql</driver>
<pool>
<min-pool-size>2</min-pool-size>
<max-pool-size>100</max-pool-size>
<prefill>>false</prefill>
<use-strict-min>>false</use-strict-min>
<flush-strategy>FailingConnectionOnly</flush-strategy>
</pool>
<security>
<user-name>postgres</user-name>
<password>system</password>
</security>
<validation>
<check-valid-connection-sql>SELECT 1</check-valid-connection-sql>
<validate-on-match>>false</validate-on-match>
<background-validation>>false</background-validation>
<background-validation-millis>120000</background-validation-millis>
</validation>
<statement>
<prepared-statement-cache-size>0</prepared-statement-cache-size>
<share-prepared-statements>>true</share-prepared-statements>
</statement>
</datasource>
<drivers>
<driver name="postgresql" module="org.postgresql">
<xa-datasource-class>org.postgresql.xa.PGXADatasource</xa-datasource-
class>
</driver>

```

```

</drivers>
</datasources>
</subsystem>
...

```

5.4.6.3 Для MS SQL

```

...
<subsystem xmlns="urn:jboss:domain:datasources:1.0">
  <datasources>
    <datasource jta="false" jndi-name="iSimpleBank" pool-name="iSimpleBank"
      enabled="true" use-java-context="true" use-ccm="false">
      <connection-
url>jdbc:sqlserver://192.168.10.236:1434;databaseName=SEB;selectMethod=cu
rsor;sendStringParametersAsUnicode=false</connection-url>
      <driver-class>com.microsoft.sqlserver.jdbc.SQLServerDriver</driver-
class>
      <driver>sqlserver</driver>
      <pool>
        <min-pool-size>50</min-pool-size>
        <max-pool-size>100</max-pool-size>
        <prefill>>false</prefill>
        <use-strict-min>>false</use-strict-min>
        <flush-strategy>FailingConnectionOnly</flush-strategy>
      </pool>
      <security>
        <user-name>sa</user-name>
        <password>Axios100</password>
      </security>
      <validation>
        <check-valid-connection-sql>SELECT 1</check-valid-connection-sql>
        <validate-on-match>>false</validate-on-match>
        <background-validation>>false</background-validation>
        <background-validation-millis>120000</background-validation-millis>
      </validation>
      <statement>
        <prepared-statement-cache-size>0</prepared-statement-cache-size>
        <share-prepared-statements>>true</share-prepared-statements>
      </statement>
    </datasource>
    <drivers>
      <driver name="sqlserver" module="com.microsoft.sqlserver.jdbc">
        <xa-datasource-
class>com.microsoft.sqlserver.jdbc.SQLServerXADataSource</xa-datasource-
class>
      </driver>
    </drivers>
  </datasources>
</subsystem>
...

```

Для того, чтобы JBoss AS мог работать с драйвером, необходимо в **<каталог развёртывания JBoss AS>\modules** описать драйвер.

Пример для **MSSQL**:

1. Создать каталог **<каталог развёртывания JBoss AS>\modules\com\microsoft\sqlserver\jdbc**
2. В созданный каталог скопировать драйвер sqljdbc4.jar
3. Создать файл main.xml содержащий описатель драйвера

```

<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1"

```

```

name="com.microsoft.sqlserver.jdbc">
<resources>
<resource-root path="sqljdbc4.jar"/>
</resources>
<dependencies>
<module name="javax.api"/>
<module name="javax.transaction.api"/>
</dependencies>
</module>

```

Пример для PostgreSQL:

1. Создать каталог
<каталог развёртывания JBoss AS>\modules\org\postgresql\main
2. В созданный каталог скопировать драйвер postgresql-9.0-801.jdbc4.jar
3. Создать файл main.xml содержащий описатель драйвера

```

<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1" name="org.postgresql">
<resources>
<resource-root path="postgresql-9.0-801.jdbc4.jar"/>
</resources>
<dependencies>
<module name="javax.api"/>
<module name="javax.transaction.api"/>
</dependencies>
</module>

```

5.5 Запуск сервера приложений и проверка доступности системы ДБО iSimpleRetail

На завершающем этапе развёртывания системы ДБО iSimpleRetail необходимо проверить доступность приложения, выполнив переход на стартовую страницу системы.

- Запустите сервер приложений. Процедуры запуска и останова соответствующего сервера приложений рассмотрены в подразделе **Запуск и останов сервера приложений** документации по первоначальной настройке системы ДБО iSimpleRetail.
- Откройте в браузере ссылку на стартовую страницу системы и убедитесь в доступности контента. Ссылка для доступа к системе ДБО iSimpleRetail приводится в подразделе **Доступ к системе ДБО iSimpleRetail**. документации по первоначальной настройке системы ДБО iSimpleRetail