



БАЙТИМ

РУКОВОДСТВО ПО УСТАНОВКЕ СИСТЕМЫ

Версия 1.6.16+

Обновлено: 10.1.2026

ООО «Байтим»

Оглавление

1	Введение	3
1.1	Назначение документа	3
1.2	Сокращения и используемые термины	3
2	Основные понятия	4
2.1	Пространства и конфигурации	4
2.2	Учетные записи	4
2.3	Пользователи	5
2.4	Права и роли	5
3	Установка системы	6
3.1	Планирование и выбор оборудования	7
3.2	Типовая архитектура	7
3.3	Настройка нескольких сред	8
3.4	Предварительные требования к установке	9
3.5	Настройка Apache Tomcat	10
3.6	Конфигурация базы данных	10
3.7	Конфигурация приложения	11
3.7.1	<i>Генерация нового пароля для пользователя базы данных</i>	12
3.8	Запуск сервера	12
3.8.1	<i>Управление запуском/остановкой сервера осуществляется с помощью запуска/остановки сервера приложений Tomcat.</i>	13
3.8.2	<i>Запуск системы</i>	13
3.8.3	<i>Вход в систему</i>	13
3.8.4	<i>Остановка системы</i>	13
3.9	Процедура удаления	14

1 Введение

1.1 Назначение документа

Настоящее руководство предназначено для системных администраторов и предполагает, что читатели обладают базовыми знаниями в области администрирования корпоративных систем, настройки сетей, а также имеют некоторый опыт работы с контейнерами приложений Java, такими как Tomcat, Jetty, JBoss и другие. Также окажется полезным опыт в административании баз данных Postgres.

В документе приведены примеры с использованием команд командной строки. В самом приложении применяется язык JavaScript для выполнения макросов и вызовов функций.

1.2 Сокращения и используемые термины

Термин	Значение
JRE	Java Runtime Environment — среда выполнения Java
JVM	Java Virtual Machine — виртуальная машина Java
MFA	Многофакторная аутентификация
OTP	Одноразовый пароль
Postgres	Система управления объектно-реляционными базами данных, с открытым кодом
Servlet	Инструмент, написанный на Java, для разработки веб-приложений
Space	Логическая изоляция функциональности в системе Byteam
Tomcat	Открытый веб-сервер и контейнер сервлетов, разработанный Apache
WAF	Web Application Firewall

2 Основные понятия

2.1 Пространства и конфигурации

Система Byteam поддерживает мультитеннантность, предоставляя несколько виртуальных пространств в рамках одной установки. Данные различных пространств хранятся отдельно друг от друга; списки пользователей, адаптеры, конфигурации и настройки также разделены. Пространства используют один инстанс Tomcat и могут динамически создаваться, запускаться и останавливаться без перезагрузки Tomcat.

Предопределённое пространство под названием "system", создается автоматически, всегда доступно и не может быть удалено. После новой установки это единственное запущенное пространство (если система не поставляется предварительно настроенной с уже созданными пространствами).

Другие пространства должны быть созданы администратором из доступных конфигураций — набора файлов конфигурации, полностью описывающих конкретное приложение.

2.2 Учетные записи

В системе присутствует центральный список учетных записей, содержащий данные пользователей и учетные данные для авторизации, общий для всех пространств. Учетные записи могут быть заблокированы, удалены или отключены по мере необходимости. Только активные записи имеют доступ к системе. Имя учетной записи должно быть уникальным в пределах списка. Например, адрес электронной почты пользователя — это хороший способ идентификации учетной записи.

Пользователи, получающие доступ к системе, могут аутентифицироваться с использованием локальной базы паролей или удаленной системы, поддерживающей протоколы аутентификации, такие как MS Active Directory, сервер LDAP, Azure Active Directory, SCIM2 и системы с поддержкой OpenID Connect. Допускается использование нескольких внешних баз данных, а также их смешивание с локальными учетными записями. В этом случае комбинация базы данных пользователей и имени учетной записи должна быть уникальной.

Все пользователи имеют доступ к REST API по умолчанию. Доступ к объектам через API предоставляет те же права и видимость объектов, что и при доступе к ним через пользовательский интерфейс. Право доступа может быть отозвано администратором и предоставлено только конкретным ролям или пользователям настройками прав или конфигурационным файлом.

Для локальных учетных записей пользователи могут сами изменять и восстанавливать свои пароли. Для учетных записей, управляемых удаленно, пароли (или другие настройки учетных данных) управляются удаленной системой авторизации и не могут быть изменены средствами системы.

2.3 Пользователи

Пользователь — это представление учетной записи в конкретном пространстве. Для каждого пространства, доступного учетной записи, создается отдельный пользователь, который виден только в этом конкретном пространстве. Некоторые атрибуты учетной записи при изменении автоматически распространяются по всем пространствам. Однако возможно изменять атрибуты пользователя в одном пространстве, не влияя на значения в других доступных пространствах.

Записи пользователей содержат всю информацию о пользователе, настройки интерфейса, предпочтения и другие данные профиля, которые обновляются при использовании системы.

Пользователь может быть одновременно авторизован только в одном пространстве. Существуют функции для переключения пользователя между пространствами, если одна учетная запись имеет доступ к нескольким пространствам. Система запоминает последнее используемое пространство, и пользователь автоматически авторизуется в нем при следующем входе.

Если учетная запись синхронизируется с внешней базой данных LDAP, все данные учетной записи распространяются на атрибуты ее пользователей во всех пространствах.

2.4 Права и роли

В каждом пространстве пользователь может иметь различный набор разрешений. Разрешения — это атомарные права, определяемые в каждой конфигурации (например, "Может экспортировать файлы", "Может администрировать пространства", "Может видеть меню разработки" и так далее). Как правило, разрешения предоставляют доступ к объекту или возможность вызова определённой функции.

Для упрощения управления разрешениями в системе определены роли (например, "Разработчик", "Руководитель проекта"), которые содержат заранее определённый набор разрешений, обычно необходимых сотруднику с определённой должностью или функцией в процессе.

Рекомендуется назначать пользователям роли вместо отдельных разрешений. Если в системе используются группы пользователей (синхронизированные с LDAP или определённые локально), роли предпочтительнее назначать на группы, а не отдельным пользователям.

Эффективные права пользователя формируются из следующих элементов:

- Все права, назначенные непосредственно пользователю
- Все права всех ролей, назначенных пользователю
- Все права всех ролей, назначенных всем группам, в которые входит пользователь

3 Установка системы

Эта глава описывает процесс первичной установки системы Byteam на новый сервер.

На сайте производителя доступны скрипты для автоматической установки Byteam на дистрибутивы Ubuntu, RedHat и CentOS Linux. Эти скрипты выполняют все шаги автоматически, скачивая необходимые файлы, с настройками по умолчанию.

Далее в тексте описан также процесс ручной процесс установки, если стандартные скрипты неприменимы или требуются нестандартные настройки.

Для установки на сервере Linux (nginx+Tomcat+Postgres) используйте подготовленный скрипт:

<https://www.byteam.ru/distr/install/linux/install.sh>

Скрипт должен выполняться от имени привилегированного пользователя ОС. Скрипт автоматизирует следующее:

- Обновление патчей ОС
- Установку базы данных Postgres (если она не установлена)
- Установку Java (если она не установлена)
- Создание пользователя tomcat с домашним каталогом в /home/tomcat
- Скачивание и установку поддерживаемой версии Tomcat с изменёнными конфигурационными файлами
- Настройку Tomcat как системной службы systemd
- Настройку nginx для работы на порту 80 с использованием Tomcat как обратного прокси
- Скачивание актуальной стабильной версии Byteam и её деплой в приложение Tomcat

Система запускается после завершения выполнения скрипта, где администратору предстоит продолжить настройки через графический интерфейс. На следующем этапе необходимо также проверить конфигурацию nginx в /etc/nginx/sites-enabled и загрузить подготовленный SSL-сертификат для обеспечения защищённого соединения.

3.1 Планирование и выбор оборудования

Система Byteam может быть развернута на большинстве современных аппаратных и программных платформ. Она была протестирована на следующих платформах, совместимость с которыми подтверждена:

- MS Windows 8, 10, 11, Server (64bit и 32bit)
- Linux (Astra Linux, ALT Linux, РЕД ОС, Ubuntu, CentOS, RedHat, Rokit) на процессорах Intel или ARM
- Mac OS X (последние версии)

Как правило, в продуктивной среде активно используются системы Linux, включая российские сборки.

Требуемое программное обеспечение:

- **Сервер:** Java 8 (OpenJDK), сервер приложений Tomcat 9, база данных PostgreSQL (версии 10–16)
- **Клиент:** Современный веб-браузер (Яндекс браузер, MS Edge, Firefox, Chrome, Safari, Opera)

Клиентский интерфейс также работает на мобильных устройствах, где отображение адаптируется к размеру экрана.

3.2 Типовая архитектура

Архитектура развертывания зависит от предполагаемого количества пользователей и сценариев использования. Для небольших установок (до 100 пользователей) база данных и сервер приложений могут размещаться на одном компьютере. Для более крупных установок рекомендуется выделенный сервер базы данных.

Во всех конфигурациях рекомендуется использовать веб-прокси перед сервером приложений, подключённым к порту сервера Tomcat. Если требуется защищённое соединение HTTPS, его следует настраивать на внешнем соединении прокси-сервера, а не на самом Tomcat.

В средах с высоким уровнем защищенности, при использовании WAF, WAF следует размещать перед прокси-сервером NGINX, фильтруя входящий трафик.

Благодаря инновационной внутренней архитектуре приложение потребляет значительно меньше системных ресурсов, чем обычные приложения типа OLTP при такой же нагрузке. В силу этих особенностей, в отличие от классической архитектуры, сервер приложений нагружается меньше, чем сервер базы данных. Поэтому в большинстве ситуаций кластеризация не имеет смысла, так как один сервер приложений

может обслуживать тысячи пользователей. В случае необходимости, система поддерживает различные режимы работы в кластере, что может быть полезно для обеспечения высокой доступности и географической распределенности.

При большом количестве внешних интеграций также рекомендуется использовать выделенный сервер интеграции.

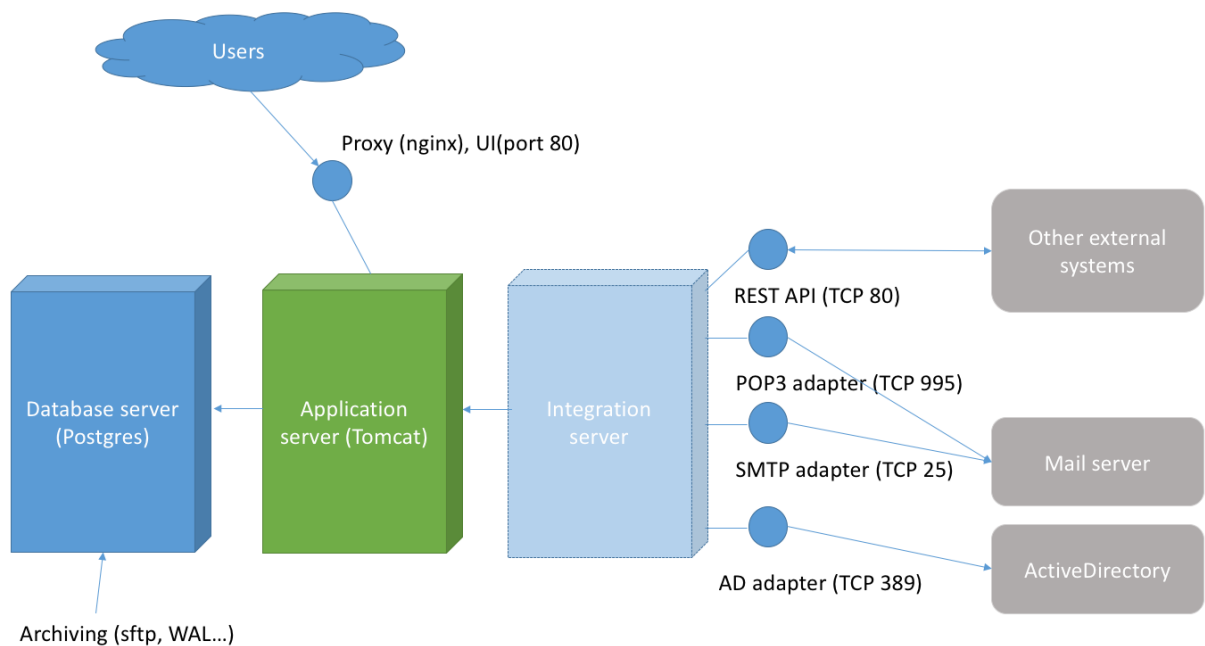


Рисунок 1 Типовая архитектура развертывания

3.3 Настройка нескольких сред

Хорошей практикой является наличие по крайней мере одной среды разработки и одной среды тестирования в дополнение к производственной среде. В высоконагруженных развертываниях также необходимо иметь эталонную среду, аналогичную рабочей, где все изменения тестируются в первую очередь.

При копировании данных из одной среды в другую учитывайте следующее:

Конфигурационные файлы содержат описание среды, такое как имя хоста, расположение файла, конфигурации, настройки ведения журнала и т.д. «ти файлы не должны копироваться между стендами.

База данных содержит некоторые элементы, которые обычно должны быть изменены или анонимизированы при перемещении данных из одной среды в другую:

- Настройки адаптера интеграции, включая системные адреса и сертификаты/пароли
- Лицензионный ключ

- Пользователи, логины, пароли

3.4 Предварительные требования к установке

1. Установка Java

Если JRE еще не установлена в системе, загрузите и установите последнюю версию JRE для вашей платформы непосредственно у поставщика платформы. Проверьте версию Java, которая должна поддерживаться системой.

Рекомендуется установить 64-разрядную JRE, если ваша система ее поддерживает. В крупных производственных установках это обязательное требование, в противном случае, Byteam не сможет использовать всю доступную оперативную память системы. Прочтите инструкции по установке Oracle JVM (или другой JVM) для вашей платформы.

После установки системная переменная JAVA_HOME должна быть определена и видна пользователю приложения, указывающей на правильную версию JRE.

2. Установка Apache Tomcat

Приложение может быть запущено на существующем по умолчанию Tomcat, предустановленном вместе с вашей ОС, но рекомендуется установить выделенный экземпляр Tomcat в отдельную папку от оригинального дистрибутива в <https://tomcat.apache.org>

Убедитесь, что приняты соответствующие меры безопасности для защиты вашей установки. Подробности также доступны в документации Tomcat по адресу <https://tomcat.apache.org/tomcat-9.0-doc/setup.html> Рекомендуется создать отдельного пользователя ОС для Tomcat, чтобы избежать запуска приложения под учетной записью административной системы.

При установке для платформы Windows рекомендуется скачивать tomcat в виде zip-архива с официального сайта, а не в виде установщика службы Windows. После загрузки распакуйте пакет в произвольный каталог, например, C:\tomcat.

При установке Linux Tomcat обычно предоставляется в системном дистрибутиве, но рекомендуется загрузить пакет .tar с веб-сайта Apache Tomcat, который имеет стандартные настройки

3. Установка СУБД PostgreSQL

Процедура установки Postgres может варьироваться в зависимости от типа системы. Для большинства дистрибутивов Linux существуют предварительно собранные пакеты, которые могут быть уже частью дистрибутива ОС.

Ознакомьтесь с документацией по установке для типа и версии платформы.

Общая документация по Postgres доступна здесь:

<https://www.postgresql.org/docs/current/tutorial-install.html>

Рекомендуется помещать индексные файлы базы данных на более быстрый диск, желательно SSD. На тестовых серверах достаточно использовать параметры базы

данных по умолчанию. По параметрам производственной системы проконсультируйтесь с администратором базы данных или представителем поставщика для уточнения оптимальных настроек для данного профиля загрузки конфигурации.

3.5 Настройка Apache Tomcat

Определите нахождение папки конфигурации Tomcat и измените файлы `server.xml` и `context.xml`, как показано ниже. Обычно каталог находится в `$TOMCAT_HOME/conf`.

Обратите внимание: В некоторых дистрибутивах Linux папка конфигурации может отличаться от папки приложения Tomcat. В специализированных средах параметры конфигурации могут находиться в скриптах и т.д. Проконсультируйтесь с системным администратором, если вы не уверены, где находится каталог конфигурации.

Установите параметры сервера Tomcat:

1. В `server.xml` файле внесите следующие изменения:

В разделе `<Connector>` добавьте `URLEncoding` и атрибуты `maxThreads` следующим образом:

```
<Connector URLEncoding="UTF-8" maxThreads="400" connectionTimeout="20000"
port="8080" protocol="HTTP/1.1" redirectPort="8443"/>
```

Примечание: установите атрибут `maxThreads` близкий к ожидаемому пиковому количеству одновременных пользователей, умноженное на 2. Если это число меньше 400, установите `maxThreads` в значение 400.

2. Отключите кэширование статического контента в `context.xml`:

```
<Context cachingAllowed="false">
```

3.6 Конфигурация базы данных

Если вы используете отдельные серверы для экземпляра базы данных и экземпляра приложения, убедитесь, что ваш брандмауэр разрешает подключение к базе данных, а настройки безопасности Postgres разрешают удаленное подключение с сервера приложений.

Этапы настройки:

- Создание пользователя базы данных Postgres
- Создайте новую базу данных Postgres и предоставьте пользователю все права доступа к этой базе данных.
- Убедитесь, что ваша база данных настроена на правильную кодировку символов и сортировку для выбранного языка.

Например, чтобы создать базу данных 'byteam' с пользователем 'jupiter' и паролем 'first' выполните следующие команды из консоли `psql`. Команды также могут выполняться из клиента с графическим интерфейсом, такого как `pgAdmin` или подобного.

```
create user jupiter with password 'first';
create database byteam;
grant all privileges on database byteam to jupiter;
```

Для Postgres версий 12 и выше отключите `jit`. `Jit` включен по умолчанию, начиная с версии 12, и может вызвать проблемы с производительностью. В файле `postgresql.conf` найдите строку, содержащую `"jit="`, или добавьте одну строку, чтобы она выглядела следующим образом:

```
jit = off
```

3.7 Конфигурация приложения

Шаги:

1. Создайте локальную директорию, в которую будут размещены конфигурационные файлы. После создания определите переменную среды с именем `BYTEAM_PATH`, которая указывает на расположение этого каталога. Рекомендуется установить переменную в скрипте запуска `Tomcat` или скрипте пользовательской среды (`.profile` и т.д.), чтобы переменная загружалась при каждом запуске сервера. Отредактируйте файл и вставьте команду, например:

```
SET BYTEAM_PATH=/home/tomcat/byteam
```

- Убедитесь, что переменная настроена при перезагрузке системы, введя команду в `init` или `login` скрипте.
2. Если `Tomcat` запускается как системная служба, эта переменная окружения может быть установлена скриптом определения демона.
 3. В папке из шага 1 создайте подпапку с именем `"conf"` и внутри этой папки создайте файл с именем `"default.properties"`. В приведенном выше примере файл будет расположен по адресу:


```
/home/tomcat/byteam/conf/default.properties
```
 4. Поместите конфигурацию подключения к базе данных в файл `default.properties` или скопируйте значения по умолчанию из файла шаблона

```
# Пример конфигурационного файла:
dbname=jupiter
```

```
dbport=5432
dbhost=localhost
dbuser=jupiter
dbpassword=..
dbmaxconnections=20
dbinitconnections=10
dbdriverclass=org.postgresql.Driver
```

5. Разверните архив WAR в Tomcat, скопировав архив в папку tomcat webapps. Процесс развертывания см. в документации Tomcat. Для тестирования среды можно использовать autodeploy. В производственных средах автоматическое развертывание должно быть отключено, чтобы избежать непредвиденных перезапусков приложений.

Параметр dbmaxconnections должен быть настроен в соответствии с типом установки. Для производственных систем используйте не менее (количество одновременных пользователей / 2). Для тестирования или разработки используйте минимальную настройку 10. Для систем общего назначения без большой нагрузки 20 является безопасным выбором.

3.7.1 Генерация нового пароля для пользователя базы данных

В этом примере конфигурации зашифрованный пароль по умолчанию имеет значение "first". Система ожидает этот пароль

Если вы не предоставите никакой конфигурации. Если вы хотите установить другой пароль базы данных, вы можете использовать утилиту шифрования паролей EncryptPassword. Утилита написана на java, поэтому ее можно запустить из командной строки в любой операционной системе. Запустите его с помощью следующей команды:

```
java -jar EncryptPassword.jar
```

Вам будет предложено ввести пароль и повторно ввести его на следующем шаге, чтобы проверить, совпадает ли новый пароль. После этого создается зашифрованный хэш из пароля. Используйте этот хэш в атрибуте dbpassword файла default.properties, как показано выше.

3.8 Запуск сервера

Управление запуском/остановкой сервера осуществляется с помощью запуска/остановки сервера приложений Tomcat.

3.8.1 Управление запуском/остановкой сервера осуществляется с помощью запуска/остановки сервера приложений Tomcat.

3.8.2 Запуск системы

После новой установки сначала запустите систему, вручную запустив Tomcat, и проверьте файлы журналов tomcat на наличие возможных ошибок (по умолчанию они находятся в файле **catalina.out**).

Перед запуском сервера можно проверить, запущен ли процесс для базы данных *postgres* (без него приложение не запустится).

В системе Linux с запущенным по умолчанию tomcat в качестве системной службы выполните:

```
sudo service tomcat start
```

При выборочной установке вместо этого запустите скрипт запуска Tomcat.

Когда в журнале появляется следующее сообщение, запуск системы выполнен правильно:

```
<init>@<none> INFO server.BootUp - Byteam started with state: OK
```

Предположим, что ваш WAR-файл называется "byteam.war", развертывание называется "byteam", откройте WEB-браузер и перейдите в <http://yourhostname:8080/byteam>. Если вы изменили имя развертывания или имя архива, путь будет другим.

3.8.3 Вход в систему

В веб-браузере вы должны увидеть окно с запросом на вход в систему. Используйте известные учетные данные по умолчанию для входа в систему:

Username: admin

Password: admin123

Затем измените пароль пользователя admin при первой удобной возможности. Это можно сделать в настройках приложения в разделе Безопасность.

3.8.4 Остановка системы

Остановите экземпляр Tomcat, выполнив специальный скрипт или системную команду.

Например, в системе Linux с запущенным по умолчанию tomcat в качестве системной службы выполните:

sudo service tomcat stop

При остановке системы все пользователи будут выведены из системы, соединение с базой данных будет закрыто, внешние службы и службы синхронизации будут остановлены. В файле журнала вы должны увидеть сообщение с подтверждением:

```
INFO server.BootUp - Shutdown complete
```

3.9 Процедура удаления

Чтобы удалить программу:

1. Остановите Tomcat (сервисом операционной системы или другим способом)
2. Удалите файл *Byteam.war* и папку *Byteam* в подкаталоге Tomcat "webapps"
3. Если есть какие-либо скрипты автозапуска, удалите их из папок автозагрузки системы (в зависимости от платформы).
4. Удалите все файлы из папки `$BYTEAM_PATH`
5. Удалите базу данных Postgres (см. руководство пользователя Postgres), согласно инструкции для вашей платформы

Перед удалением системы желательно сохранить резервную копию данных.