

АХІОМ JDK

**Ахіом NIK Pro
Руководство по созданию
нативного образа из
приложения Java**

Ахіом NIK Pro | Март 2025

Copyright © 2019-2025 Все права защищены АО "АКСИОМ" (АКСИОМ)

Программное обеспечение АКСИОМ содержит программное обеспечение с открытым исходным кодом. Дополнительная информация о коде сторонних разработчиков доступна на сайте https://axiomjdk.ru/third_party_licenses. Для дополнительной информации о том, как получить копию исходного кода, можно обратиться по адресу info@axiomjdk.ru.

ДАННАЯ ИНФОРМАЦИЯ МОЖЕТ ИЗМЕНЯТЬСЯ БЕЗ ПРЕДВАРИТЕЛЬНОГО УВЕДОМЛЕНИЯ. АКСИОМ ПРЕДОСТАВЛЯЕТ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ "КАК ЕСТЬ" БЕЗ КАКИХ-ЛИБО ГАРАНТИЙ, АКСИОМ ПРЯМО ОТКАЗЫВАЕТСЯ ОТ ВСЕХ ПОДРАЗУМЕВАЕМЫХ ГАРАНТИЙ, ВКЛЮЧАЯ, НО НЕ ОГРАНИЧИВАЯСЬ ПОДРАЗУМЕВАЕМЫМИ ГАРАНТИЯМИ ТОВАРНОЙ ПРИГОДНОСТИ И ПРИГОДНОСТИ ДЛЯ ОПРЕДЕЛЕННОЙ ЦЕЛИ.

АКСИОМ НИ ПРИ КАКИХ ОБСТОЯТЕЛЬСТВАХ НЕ НЕСЕТ ОТВЕТСТВЕННОСТИ ЗА ЛЮБЫЕ КОСВЕННЫЕ, СЛУЧАЙНЫЕ, СПЕЦИАЛЬНЫЕ, ШТРАФНЫЕ ИЛИ КОСВЕННЫЕ УБЫТКИ, ИЛИ УБЫТКИ ОТ ПОТЕРИ ПРИБЫЛИ, ДОХОДА, ДАННЫХ ИЛИ ИСПОЛЬЗОВАНИЯ ДАННЫХ, ПОНЕСЕННЫЕ ВАМИ ИЛИ ЛЮБОЙ ТРЕТЬЕЙ СТОРОНОЙ, БУДЬ ТО В РЕЗУЛЬТАТЕ ДЕЙСТВИЯ ДОГОВОРА ИЛИ ДЕЛИКТА, ДАЖЕ ЕСЛИ АКСИОМ БЫЛО ПРЕДУПРЕЖДЕНО О ВОЗМОЖНОСТИ ТАКИХ УБЫТКОВ.

Использование любого программного продукта АКСИОМ регулируется соответствующим лицензионным соглашением, которое никоим образом не изменяется условиями данного уведомления. Программные продукты и фирменные наименования: Axiom JDK, Axiom JDK Pro, Axiom Runtime Container Pro, Axiom Linux, Libercat, Libercat Certified и АКСИОМ принадлежат АКСИОМ и их использование допускается только с разрешения правообладателя.

Товарный знак Linux® используется в соответствии с сублицензией от Linux Foundation, эксклюзивного лицензиата Линуса Торвальдса, владельца знака на всемирной основе. Java и OpenJDK являются товарными знаками или зарегистрированными товарными знаками компании Oracle и/или ее аффилированных лиц. Другие торговые марки являются собственностью их соответствующих владельцев и используются только в целях идентификации.

Содержание

1. Введение	4
2. Установка Axiom NIK Pro	5
Linux	5
Windows	5
3. Создание проекта Spring Boot	7



1. Введение

В этом документе вы узнаете, как создать нативный образ из Java-приложения, основанного на Spring Boot 3.0. [Spring Boot 3.0.0-M5](#) содержит множество существенных улучшений, включая встроенную поддержку нативных образов Axiom NIK Pro.

2. Установка Axiom NIK Pro

Linux

Для работы с нативными образами лучше всего использовать мощный компьютер с несколькими гигабайтами оперативной памяти. Вы можете воспользоваться облачным сервисом или рабочей станцией. В дальнейшем мы будем использовать команды Linux bash. Команды macOS аналогичны.

Чтобы установить Axiom NIK Pro, загрузите файл `.tar.gz` и распакуйте его следующим образом.

```
export INSTALL_DIR=<Путь к каталогу для установки>
tar -C $INSTALL_DIR -xzf axiom-vm-pro-openjdk21.0.3+10-23.1.3+1-linux-
amd64.tar.gz
```

Убедитесь, что Axiom NIK Pro установлен правильно:

```
java -version
openjdk version "21.0.3" 2024-04-16
OpenJDK Runtime Environment Axiom-NIK-23.1.3 (build 21.0.3+10)
OpenJDK 64-Bit Server VM Axiom-NIK-23.1.3 (build 21.0.3+10, mixed mode,
sharing)
```

Вы можете добавить подкаталог `<каталог установки Axiom NIK Pro>/bin` в `$PATH` для текущего сеанса терминала, если это необходимо, а также указать путь в переменной `JAVA_HOME`. Затем вы можете использовать команды Maven с префиксом `JAVA_HOME=`.

```
export PATH=<каталог установки Axiom NIK Pro>/bin:$PATH
export JAVA_HOME=<каталог установки Axiom NIK Pro>
```

Если вы получаете сообщение об ошибке "java: No such file or directory" в Linux, возможно вы установили Axiom NIK Pro для Alpine Linux, а не для Linux. Проверьте тот ли установочный файл вы скачали и установили.

Windows

Чтобы установить Axiom NIK Pro как отдельный пакет на свой компьютер.

1. Загрузите файл пакета `.zip` и распакуйте его в папку следующим образом:

```
Invoke-WebRequest "<адрес URL для скачивания установочного файла>/axiom-vm-pro-openjdk21.0.3+10-23.1.3+1-windows-amd64.zip"  
Expand-Archive axiom-vm-pro-openjdk21.0.3+10-23.1.3+1-windows-amd64.zip  
-DestinationPath .
```

Приведенные выше команды загружают архив и распаковывают Axiom NIK Pro в текущий каталог. Если вы являетесь клиентом с активным договором поддержки, войдите на [портал поддержки](#), скопируйте ссылку на продукт Axiom NIK Pro в разделе "Технологии" и вставьте эту ссылку в команду `Invoke-WebRequest` выше.

2. Добавьте путь к <каталог установки Axiom NIK Pro>/bin в переменную PATH следующим образом.
 - a. В поисковой строке меню «Пуск» введите `env` и выберите пункт **Edit the system environment variables**.
 - b. Кликните по кнопке **Environment Variables**.
 - c. В нижней части окна под секцией **System Variables**, найдите строку со значением **Path** в первом столбце. Затем нажмите кнопку **Edit**.
 - d. В окне **Edit environment variable** нажмите кнопку **New** и введите путь к подкаталогу `/bin`, например:

```
C:/Axiom/AxiomNIK-Pro-23-OpenJDK-21/bin/
```

В данном окне можно также отредактировать или упорядочить пути. Нажмите **OK**, когда закончите.

После применения изменений в настройке переменных окружения возможно потребуются перезапуск приложений (например, приложения PowerShell). Рекомендуется перезагрузить всю операционную систему, чтобы гарантировать, что все приложения запустились с новым значением переменной окружения PATH. Для проверки текущего значения переменной окружения PATH выполните следующую команду в PowerShell:

```
$env:PATH
```

Для подробной информации об установке Axiom NIK Pro, см. документ *Инструментарий Нативных Образов Axiom NIK Pro: Руководство по установке*.

3. Создание проекта Spring Boot

Самый простой способ создать новый проект Spring Boot - это сгенерировать его с помощью [Spring Initializr](#). Выберите Java 17, Maven, JAR и версию Spring SNAPSHOT (3.0.2 на момент написания этого документа), затем заполните поля для метаданных проекта. Нам не нужны никакие зависимости.

Создайте проект и убедитесь, что все работает:

```
time java -jar ./target/native-image-demo-0.0.1-SNAPSHOT.jar

real    0m1.404s
user    0m4.883s
sys     0m0.169s
```

Запуск Spring занимает всего секунду, но с нативным образом GraalVM мы можем добиться еще меньшего времени.

Давайте создадим проект с помощью следующей команды:

```
JAVA_HOME= ./mvnw -Pnative native:compile
```

Результирующий исходный образ находится в указанном каталоге.

Проверьте, насколько быстро запускается исходный образ:

```
time /home/username/test/native-image-demo/target/native-image-demo

real    0m0.026s
user    0m0.013s
sys     0m0.013s
```

Запуск происходит в несколько раз быстрее, чем при использовании стандартного Spring JAR.

