Настройка рабочего места разработчика на React Native

Оглавление

[Настройка рабочего места разработчика на React Native 1](#_Toc512600466)

[1. Термины и сокращения 3](#_Toc512600467)

[2. Введение 3](#_Toc512600468)

[3. *Sonatype Nexus* - менеджер бинарных репозиториев 3](#_Toc512600469)

[3.1 Локальная установка и настройка *Nexus* 3](#_Toc512600470)

[3.2 Установка на *Windows 7* 3](#_Toc512600471)

[3.3 Виды используемых репозиториев 3](#_Toc512600472)

[3.4 Создание прокси-репозитория 3](#_Toc512600473)

[3.5 Создание приватного репозитория 4](#_Toc512600474)

[3.6 Создание группового репозитория 4](#_Toc512600475)

[4. Node.js 4](#_Toc512600476)

[4.1 Установка *Node.js на Windows 7* 4](#_Toc512600477)

[4.2 Пакетный менеджер *npm* 5](#_Toc512600478)

[4.3 Файл конфигурации *.npmrc* 5](#_Toc512600479)

[4.4 Базовая конфигурация npm-пакета - *package.json* 5](#_Toc512600480)

[4.5 Установка npm-пакета 5](#_Toc512600481)

[4.6 Приватная публикация npm-пакетов 6](#_Toc512600482)

[4.6.1 Подготовка параметра \_auth 6](#_Toc512600483)

[5. Работа с эмуляторами 6](#_Toc512600484)

[6. Работа с эмуляторами *Android Studio* 7](#_Toc512600485)

[6.1 Установка 7](#_Toc512600486)

[Для того, чтобы работать с эмуляторами Android Studio, нужно установить Android Studio. Для этого нужно отсюда скачать его дистрибутив и выполнить соответствующую программу установки. 7](#_Toc512600487)

[6.2 Запуск эмулятора 7](#_Toc512600488)

[6.3 Запуск приложения 7](#_Toc512600489)

[7. Работа с эмуляторами *Genymotion* 7](#_Toc512600490)

[8. Ссылки 7](#_Toc512600491)

[8.1 Understanding Packages and Modules 7](#_Toc512600492)

[8.2 Node.js – главный сайт 7](#_Toc512600493)

[8.3 React – главный сайт 7](#_Toc512600494)

[8.4 React Native – главный сайт 7](#_Toc512600495)

[8.5 Nexus Repository Manager OSS 2 7](#_Toc512600496)

[8.6 Дистрибутив Nexus 7](#_Toc512600497)

[8.7 Утилита Certutil 7](#_Toc512600498)

[8.8 Android Studio - главный сайт 7](#_Toc512600499)

[8.9 Genymotion – главный сайт 7](#_Toc512600500)

# Термины и сокращения

*Пакет npm* – файл или директория, описываемая посредством *package.json*.

*Модуль* – то, что может быть загружено программой *Node.js* при помощи функции *require() (*отсюда: *4.1)*.

# Введение

Настоящий документ содержит краткое руководство по настройке рабочего места разработчика на [React](https://reactjs.org/) и [React Native](https://facebook.github.io/react-native/).

# *Sonatype Nexus* - менеджер бинарных репозиториев

Менеджеры бинарных репозиториев обеспечивают организованную работу с множеством артефактов, готовых к конечной сборке.

В функции менеджера репозиториев, в частности, входит:

* проксирование запросов на получение артефактов
* кеширование артефактов с настраиваемыми политиками
* поддержка приватных репозиториев
* структурирование множества репозиториев

В данном случае в качестве артефактов выступают пакеты npm, а для организации работы с пакетами используется open source-продукт компании Sonatype - *Nexus Repository Manager OSS 2* (см. 4.4) (далее по тексту - *Nexus*).

## Локальная установка и настройка *Nexus*

Обычно менеджеры репозиториев устанавливаются на корпоративных серверах - в закрытых сетях. Но для обеспечения единообразной работы в разных средах - в закрытых и открытых сетях - на компьютерах, работающих непосредственно с Интернет, целесообразно иметь локальные установки *Nexus*. В этих случаях репозитории *Nexus* используются для поддержки единообразной среды посредством синхронизации своего содержимого. Это особенно касается синхронизации непубличных составляющих репозиториев.

## Установка на *Windows 7*

1. Скачать дистрибутив Nexus (см. 8.5).
2. Скопировать содержимое архива в подходящее для работы место - *%NEXUS\_HOME%*.
3. Перейти в директорию *%NEXUS\_HOME%\bin*.
4. Из командной строки выполнить скрипт *nexus.bat*.

В результате этого будет создан Windows-сервис *nexus*, с автоматическим режимом запуска.

Вход в Nexus выполняется из браузера по адресу *http://%nexus\_host%:8081/nexus*. Для Nexus, установленного локально, соответственно: *http://localhost:8081/nexus*.

Исходные параметры входа для администратора: *admin/admin123*.

## Виды используемых репозиториев

В работе используется три вида репозитория:

* прокси-репозиторий - в основном используется для кэширования артефактов
* приватный репозиторий - используется для публикации приватных пакетов
* групповой репозиторий - используется как объединяющий репозиторий для первых двух

## Создание прокси-репозитория

Создание прокси-репозитория выполняется следующим образом:

* Войти в *Nexus*, как администратор.
* Перейти по ссылке в раздел *Repositories*.
* Выполнить действие *Add/Proxy Repository.*
* Заполнить поля:
  + *Repository ID* (указать ‘npm-proxy’),
  + *Repository Name* (указать ‘Npm Proxy’),
  + *Repository Type* (указать ‘proxy’),
  + *Provide*r (указать ‘npm’),
  + *Remote Storage Location* (указать ‘*https://registry.npmjs.org/*’)
* Сохранить изменения нажатием кнопки *Save*.

## Создание приватного репозитория

Создание прокси-репозитория выполняется следующим образом:

* Войти в *Nexus*, как администратор.
* Перейти по ссылке в раздел *Repositories*.
* Выполнить действие *Add/Hosted Repository.*
* Заполнить поля:
  + *Repository ID* (указать ‘npm-proxy’),
  + *Repository Name* (указать ‘Npm Proxy’),
  + *Repository Type* (указать ‘proxy’),
  + *Provide*r (указать ‘npm’),
* Сохранить изменения нажатием кнопки *Save*.

## Создание группового репозитория

Создание прокси-репозитория выполняется следующим образом:

* Войти в *Nexus*, как администратор.
* Перейти по ссылке в раздел *Repositories*.
* Выполнить действие *Add Repository Group.*
* Заполнить поля:
  + *Repository ID* (указать ‘npm-group’),
  + *Repository Name* (указать ‘Npm Group’),
  + *Provide*r (указать ‘npm’),
* Сохранить изменения нажатием кнопки *Save*.

# Node.js

*Node.js* является ключевым звеном разработки на *Javascript* и в том числе на *React/React Native*, где *Node.js* выполняет роль движка, на котором работают почти все средства разработки: препроцессоры, компиляторы, сборщики, ....

## Установка *Node.js на Windows 7*

1. С сайта *Node.js* (8.2) скачать версию программы установки node-v\*.msi, рекомендуемую большинству пользователей.
2. Выполнить программу установки.
3. Проверить корректность установки можно, например, выполнением из командной строки команд: *node -v* или *npm –v*. В ответ должна быть выдана строка с идентификатором версии*.*

## Пакетный менеджер *npm*

Пакетный менеджер *npm* поддерживает базовый интерфейс командной строки, через который выполняется вся необходимая работа с пакетами: публикация, скачивание, запуск, и т.п..

## Файл конфигурации *.npmrc*

В Windows 7 файл конфигурации npm находится в файле *%USER\_HOME%/.npmrc*

Адрес удалённого репозитория

Главным параметром конфигурации является *registry* - адрес репозитория, который служит источником данных для *npm*. По умолчанию используется адрес

*https://registry.npmjs.org/*

В данном случае, при использовании *Nexus* параметру *registry* задаётся значение:

*http://%NEXUS\_HOST%:8081/nexus/content/repositories/npm-group/*:

*...*

*registry=http://%NEXUS\_HOST%:8081/nexus/content/repositories/npm-group/*

...

## Базовая конфигурация npm-пакета - *package.json*

Файл *package.json* содержит в себе главную информацию о пакете: название, версию, зависимости, и т.п.. Любая директория, в которой есть этот файл, интерпретируется как npm-пакет.

Типовой файл минимального приложения, созданного с помощью [create-react-native-app](https://github.com/react-community/create-react-native-app)) имеет вид:

*{*

*"name": "crna",*

*"version": "0.1.0",*

*"private": true,*

*"devDependencies": {*

*"react-native-scripts": "1.13.2",*

*"jest-expo": "26.0.0",*

*"react-test-renderer": "16.3.0-alpha.1"*

*},*

*"main": "./node\_modules/react-native-scripts/build/bin/crna-entry.js",*

*"scripts": {*

*"start": "react-native-scripts start",*

*"eject": "react-native-scripts eject",*

*"android": "react-native-scripts android",*

*"ios": "react-native-scripts ios",*

*"test": "jest"*

*},*

*"jest": {*

*"preset": "jest-expo"*

*},*

*"dependencies": {*

*"expo": "^26.0.0",*

*"react": "16.3.0",*

*"react-native": "~0.54.1"*

*}*

*}*

## Установка npm-пакета

Установка npm-пакета выполняется из директории пакета командой

*npm install*

или, более кратко:

*npm i*

## Приватная публикация npm-пакетов

Поскольку в *.npmrc* в качестве умолчательного реестра записан групповой реестр, для непосредственной публикации пакета в приватный (*hosted*) реестр его адрес нужно указывать явно:

*npm publish* ***--registry http://%nexus\_host%:8081/nexus/content/repositories/npm-hosted/***

Кроме того, чтобы публикация была возможной, файл *.npmrc* должен содержать параметры: *email*, *always-auth* и *\_auth*.

### Подготовка параметра \_auth

В ОС Windows значение параметра *\_auth* получается на основе *username/password* при помощи использования утилиты [certutil](http://cmd.readthedocs.io/certutil.html) (см. 5.6). Для этого создаётся текстовый файл *in.txt* с содержимым вида ‘*admin:admin123’*.

Далее выполняется команда:

*certutil /encode in.txt out.txt*

После этого файл *out.txt* содержит результат:

*-----BEGIN CERTIFICATE-----*

*YWRtaW46YWRtaW4xMjM=*

*-----END CERTIFICATE-----* ,

значение которого задаётся в параметре \_auth файла *.npmrc*. Фрагмент файла *.npmrc* в этом случае может иметь вид:

...

*email=ivanpetrov@gmail.com*

*always-auth=true*

*\_auth="YWRtaW46YWRtaW4xMjM="*

...

# Работа с эмуляторами

Для разработки приложений на React Native  широко используются эмуляторы устройств моделей разнообразных устройств, на которых могут быть установлены различные версии операционных систем, главным образом, Android и iOS.

Для разработки приложений на Android обычно используются эмуляторы [*Android Studio*](https://developer.android.com/studio/) или [*Genymotion*](https://www.genymotion.com/).

Для разработки приложений на iOS используются эмуляторы [*XCode*](https://developer.apple.com/xcode/), которые могут работать только в операционных системах, разрабатываемых для компьютеров Apple Mac.

Однако в случае использования подхода, при котором разрабатывается только кроссплатформенная составляющая на javascript, в принципе можно использовать любой эмулятор независимо от целевой платформы.

# Работа с эмуляторами *Android Studio*

## Установка

# Для того, чтобы работать с эмуляторами Android Studio, нужно установить Android Studio. Для этого нужно [отсюда](https://developer.android.com/studio/) скачать его дистрибутив и выполнить соответствующую программу установки.

## Запуск эмулятора

Запуск эмулятора из командной строки выполняется командой:

*%ANDROID\_SDK\_HOME%/tools/bin/emulator -avd DeviceName*

*DeviceName* - имя одного из эмуляторов установленных устройств.

Список этих устройств можно получить, выполнив команду:

*%ANDROID\_SDK\_HOME%/tools/bin/avdmanager list avd*

## Запуск приложения

Перед запуском приложения должен быть запущен эмулятор устройства.

Если приложение создавалось командой [create-react-native-app](https://facebook.github.io/react-native/docs/getting-started.html), то оно запускается на выполнение командой:

*npm start*

# Работа с эмуляторами *Genymotion*

Эмуляторы *Genymotion* (см. 8.8) в общем случае являются платными.

При некоммерческом использовании эмулятора доступны бесплатные версии, которые устанавливаются в индивидуальном порядке.

# Ссылки

## [Understanding Packages and Modules](https://docs.npmjs.com/getting-started/packages)

## [Node.js](https://nodejs.org/en/) – главный сайт

## [React](https://reactjs.org/) – главный сайт

## [React Native](https://facebook.github.io/react-native/) – главный сайт

## [Nexus Repository Manager OSS 2](https://help.sonatype.com/repomanager2)

## [Дистрибутив Nexus](https://www.sonatype.com/oss-thank-you-zip)

## Утилита [Certutil](http://cmd.readthedocs.io/certutil.html)

## [Android Studio](https://developer.android.com/studio/index.html) - главный сайт

## [Genymotion](https://www.genymotion.com/) – главный сайт