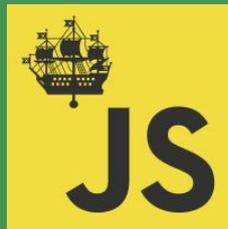


# Разработка кросс-платформенных мобильных гибридных приложений на базе Node.js с использованием Rhomobile

---



Node.js SPb Meetup #3  
31 марта 2017 года



## Солдатенков Дмитрий

Сооснователь, Технический директор  TECHNOLOGIES

Еще в школе понял, что программирование это то, чем я хочу заниматься.

Работал в одной из первых в России фирм по разработке компьютерных игр. Затем работа в большой команде над крупным проектом для разработчиков. Постепенно сфера интересов сместилась на мобильные платформы, и вот уже более 12 лет я занимаюсь именно этим.

Работал в компаниях

AMI, TogetherSoft, Borland, LG Electronics, TWP, Rhomobile, Motorola Solutions, Zebra Technologies.

Подробнее : <https://www.linkedin.com/in/dsoldatenkov>

E-Mail: [dsoldatenkov@tau-technologies.com](mailto:dsoldatenkov@tau-technologies.com)

Умеем/владеем:

- Умеем программировать на JavaScript
- Владеем Node.js платформой
- Node.js фреймворки

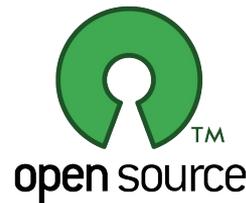
Возможно/желательно :

- HTML/CSS
- Web JavaScript
- Web JavaScript фреймворки



Разрабатывать мобильные приложения :

- для iOS, Android и возможно для WinCE/WM, WP и пр.
- кросс-платформенные
- без специфических знаний о каждой мобильной платформе
- с использованием наших знаний о Node.js и HTML/CSS/JS
- с использованием нашего наработанного кода для Node.js и Web
- с использованием бесплатных решений
- с использованием open source решений
- и еще ...





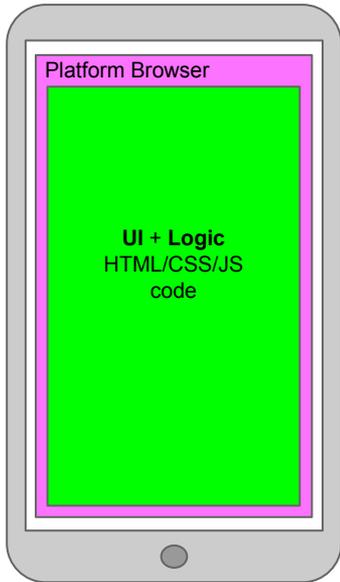
Машинка губозакатывательная, импортная !

**Simple not cross-platform  
Native Application**



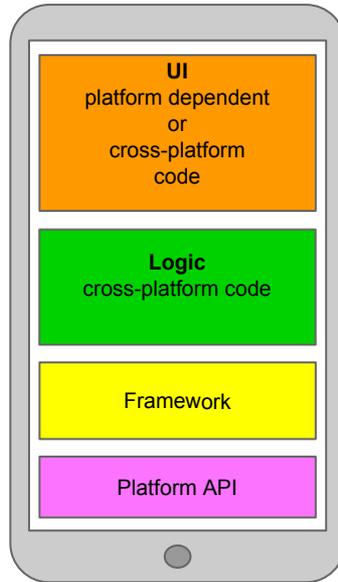
iOS: ObjC, Swift, C++  
Android: Java, C++  
WinCE/WM: C#, C++

**Web  
cross-platform Application**



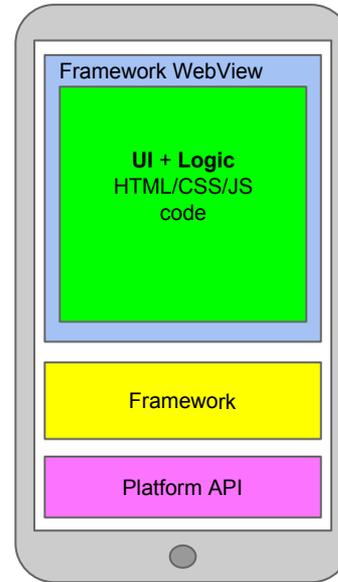
a lot of  
HTML/CSS/JS frameworks

**Native  
cross-platform Application**



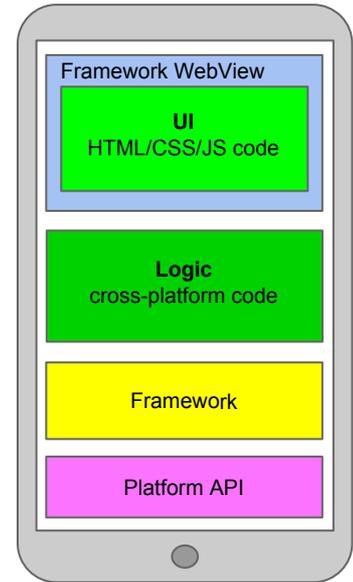
Xamarin (C#)  
Appcelerator (JS)  
React Native (JS)  
NativeScript (JS)  
QT (C++, QML)  
RubyMotion (Ruby)  
CodenameOne (Java)  
Corona (Lua)

**Hybrid  
cross-platform Application**

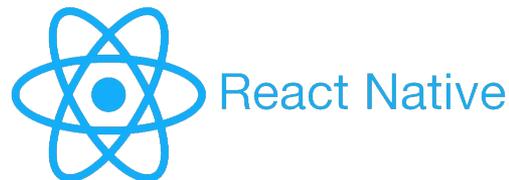


Cordova/PhoneGap  
Rhomobile  
+ a lot of  
HTML/CSS/JS frameworks

**Mixed Hybrid  
cross-platform Application**



Cordova/PhoneGap with  
jxCORE plugin,  
Rhomobile (Ruby/JS)  
+ a lot of  
HTML/CSS/JS frameworks



- Платформа предлагает писать кросс-платформенный код на JavaScript.
- Предлагает платформенно-зависимый UI - JavaScript wrappers на платформенное API
- Также предлагает JavaScript wrappers на все платформенное API.
- Тесная интеграция с Angular
- необходимость изучения платформенного/специфичного API
- “жесткая привязка” к решению



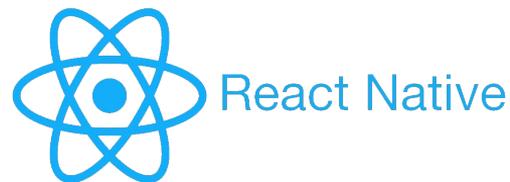
название	<b>NativeScript</b>
разработчик	<b>Progress Software Corporation</b>
сайт	<a href="https://www.nativescript.org/">https://www.nativescript.org/</a>
тип	нативный
исходный код	full  open source
доступность	<b>FREE</b>
Платформы	iOS 

- Комплексное решение.
- Облачные компоненты.
- Разработка как на NativeScript так и на Cordova
- Облачные сервисы по хранению данных, аналитике и т.п.
- WP только для Cordova варианта



название	<b>Telerik Platform</b>
разработчик	<b>Progress Software Corporation</b>
сайт	<a href="http://www.telerik.com/platform">http://www.telerik.com/platform</a>
тип	гибридный / нативный
исходный код	<del> open source</del>
доступность	
Платформы	iOS  

- Платформа предлагает писать кросс-платформенный код на JavaScript.
- Предлагает платформенно-зависимый UI - JavaScript wrappers на платформенное API
- В основе React.js
- необходимость изучения платформенного/специфичного API
- “жесткая привязка” к решению



название	<b>React Native</b>
разработчик	<b>Facebook</b>
сайт	<a href="https://facebook.github.io/react-native/">https://facebook.github.io/react-native/</a>
тип	нативный
исходный код	full  open source
доступность	
Платформы	iOS 

- Платформа позволяет писать кроссплатформенный код на JavaScript. Однако в отличие от PhoneGap не используется локальный WebView - по сути предлагается просто возможность писать обычный код на JavaScript и использовать предоставляемый единый, а также специфичный для всех платформ API для прочего функционала. Может потребовать разработки отдельного UI кода для каждой платформы.
- Приложение имеет лучшую производительность по сравнению с гибридными приложениями, так как по сути является нативным и не несет ограничений и потерь из-за использования WebView. Особенно это касается UI.
- кросс-платформенный UI фреймворк Alloy
- Недостатком является то, что код полностью привязан к платформе и специфичному API - разработчик становится сильно “привязан” к решению.
- Несмотря на использование языка JavaScript, разработчик не сможет быстро привлечь имеющих программистов JavaScript, так как для работы с решением необходимо изучение специфичных API решения.



название	<b>Appcelerator Titanium</b>
разработчик	<b>Appcelerator</b>
сайт	<a href="http://www.appcelerator.com/">http://www.appcelerator.com/</a>
тип	нативный
исходный код	<del> source</del>
доступность	
Платформы	iOS 

- PhoneGap это решение “на базе” Apache Cordova - <https://cordova.apache.org/>. Cordova это решение для осуществления вызовов нативного функционала из JavaScript в локальном WebView приложения. Таким образом приложение может иметь один универсальный код на JavaScript и реализованный на разных платформах единый API для доступа к нативному функционалу. Подробнее о взаимоотношении PhoneGap и Cordova: <https://habrahabr.ru/post/272873/> .
- Также на базе Cordova решения предлагают ведущие игроки на рынке : IBM, Intel, Oracle и другие.
- Платформа хорошо поддерживаемая, развивающаяся и одна из наиболее популярных на рынке.



APACHE  
CORDOVA™



название	<b>PhoneGap</b>
разработчик	<b>Adobe</b>
сайт	<a href="http://phonegap.com/">http://phonegap.com/</a>
тип	гибридный
исходный код	full  open source
доступность	<b>FREE</b>
Платформы	iOS  

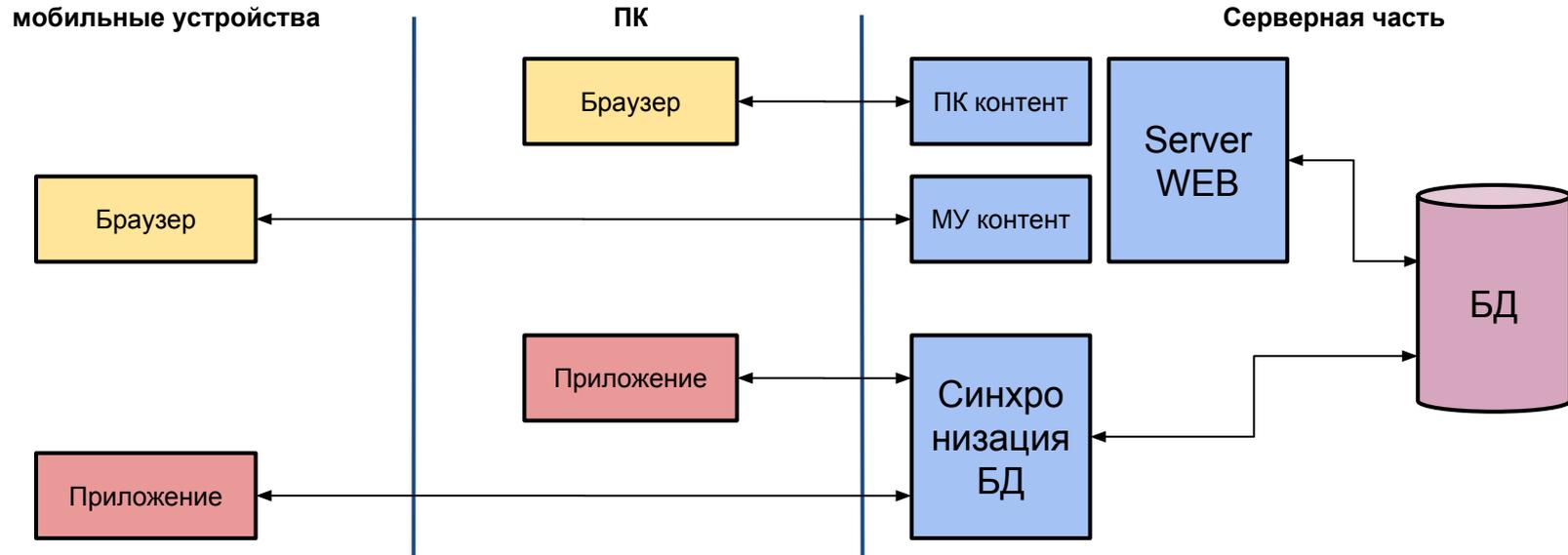
- Rhomobile гибридное решение но не на базе Cordova - используется собственное решение для связи WebView с нативным API
- На устройстве воспроизводится окружение похожее на Ruby on Rails - UI в WebView, а данные, логика реализуются в Ruby контроллерах на локальном HTTP сервере.
- На устройстве воспроизводится окружение похожее на Node.js - UI в WebView, а данные, логика реализуются в Javascript контроллерах на локальном HTTP сервере.
- Возможно использовать решение как чисто гибридное без Ruby/Node.js - в этом случае архитектурно получаем полный аналог Cordova/PhoneGap
- Готовое решение для синхронизации данных с удаленным сервером (поддерживаются основные BD)
- Комплексное решение ориентированное на корпоративный сектор, включая промышленные устройства на WM/WinCE и Android



## RHOMOBILE SUITE

название	<b>Rhomobile</b>
разработчик	<b>TAU Technologies</b>
сайт	<a href="http://tau-technologies.com/">http://tau-technologies.com/</a>
тип	гибридный смешанный
исходный код	full  open source
доступность	<b>FREE</b>
Платформы	iOS    WinCE

Это не просто приложение на мобильном устройстве, а часть большого комплекса ПО, включающего в себя различные части на разных платформах.





- дешевле (меньше кода)
- быстрее разработка (не нужно согласовывать на разных платформах)
- не нужно заботиться о поддержке новых ОС и новых версий ОС
- меньше разработчиков
- можно использовать имеющихся разработчиков и решения (для гибридных)



- хуже производительность (не всегда)
- менее привлекательный интерфейс и функционал (без использования самых свежих возможностей платформ)
- зависимость от поставщика решения (особенно для не open-source)



≠



Гибридные приложения наиболее рациональный выбор для корпоративной разработки.

- возможность использовать имеющихся HTML/CSS/JavaScript специалистов без дополнительного обучения
- можно использовать единый код и компоненты для мобильных приложений и веб-приложений
- нет “жесткой” привязки к используемому решению - например можно легко перенести основной массив HTML/CSS/JavaScript кода между Rhomobile и Cordova/PhoneGap вообще без изменений.
- проблема производительности, особенно UI, теряет актуальность из-за роста производительности аппаратной части и оптимизации WebView.
- в распоряжении разработчика - огромное количество разнообразных JavaScript фреймворков, модулей, готовых решений на любой вкус !



UI на HTML/CSS/JS уступает нативному UI как по визуальному уровню, так и по быстродействию.

Это основная причина популярности Javascript решений с нативным UI - Appcelerator, React Native, NativeScript и др.

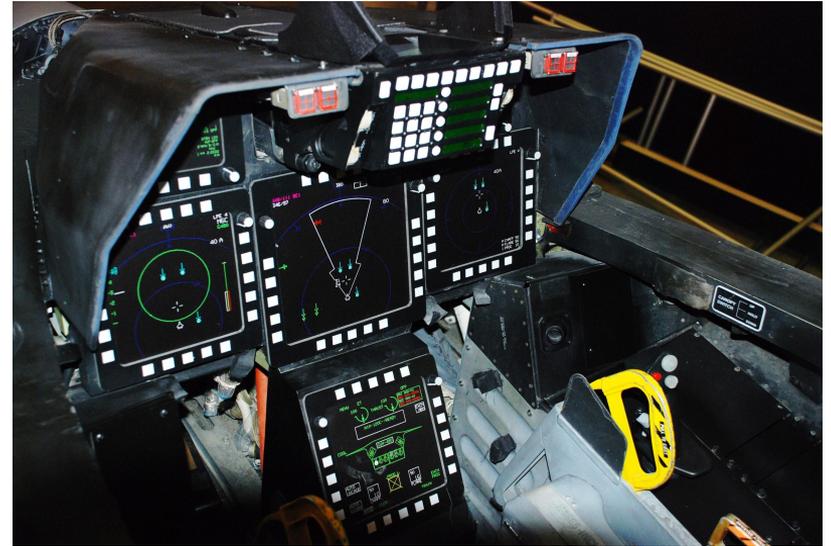


Но не все так плохо !

Во-первых корпоративный сектор имеет невысокие требования по части “красоты”.

Во-вторых современные JavaScript фреймворки предлагают уровень UI мало отличимый от нативного, как по “красотам” там и по быстродействию (особенно на современных устройствах)

1. Различное поведение WebView на разных платформах
2. Различное поведение WebView в разных версиях платформы - особенно актуально при поддержке старых версий Android
3. Тотальная отсталость WebView основанного на IE на WM/WinCE платформе.



Что делать :

1. Использовать готовые JavaScript UI фреймворки, которые решают проблему
2. Использовать собственный WebView а не системный. Например Crosswalk (<https://crosswalk-project.org/>).
3. Использовать порт WebKit под WM/WinCE как в решении Rhomobile.

Проблемы гибридных решений аналогичны проблемам больших web сайтов/порталов.

Если у вас много контента, много данных, то все это “висит” в WebView и производительность падает.

При этом мы вынуждены держать все в WebView - и данные и логику и UI.

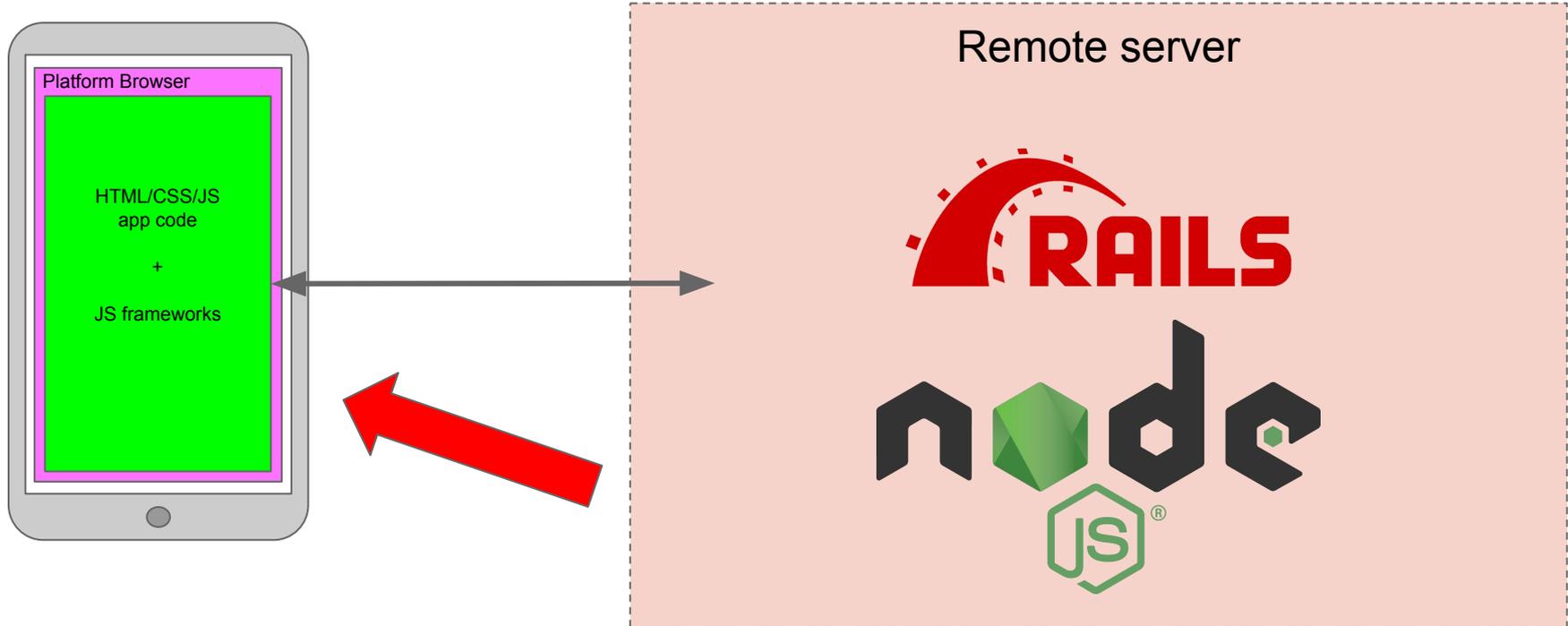
WebView однопоточный, можно выносить код в отдельные потоки, но есть много ограничений.



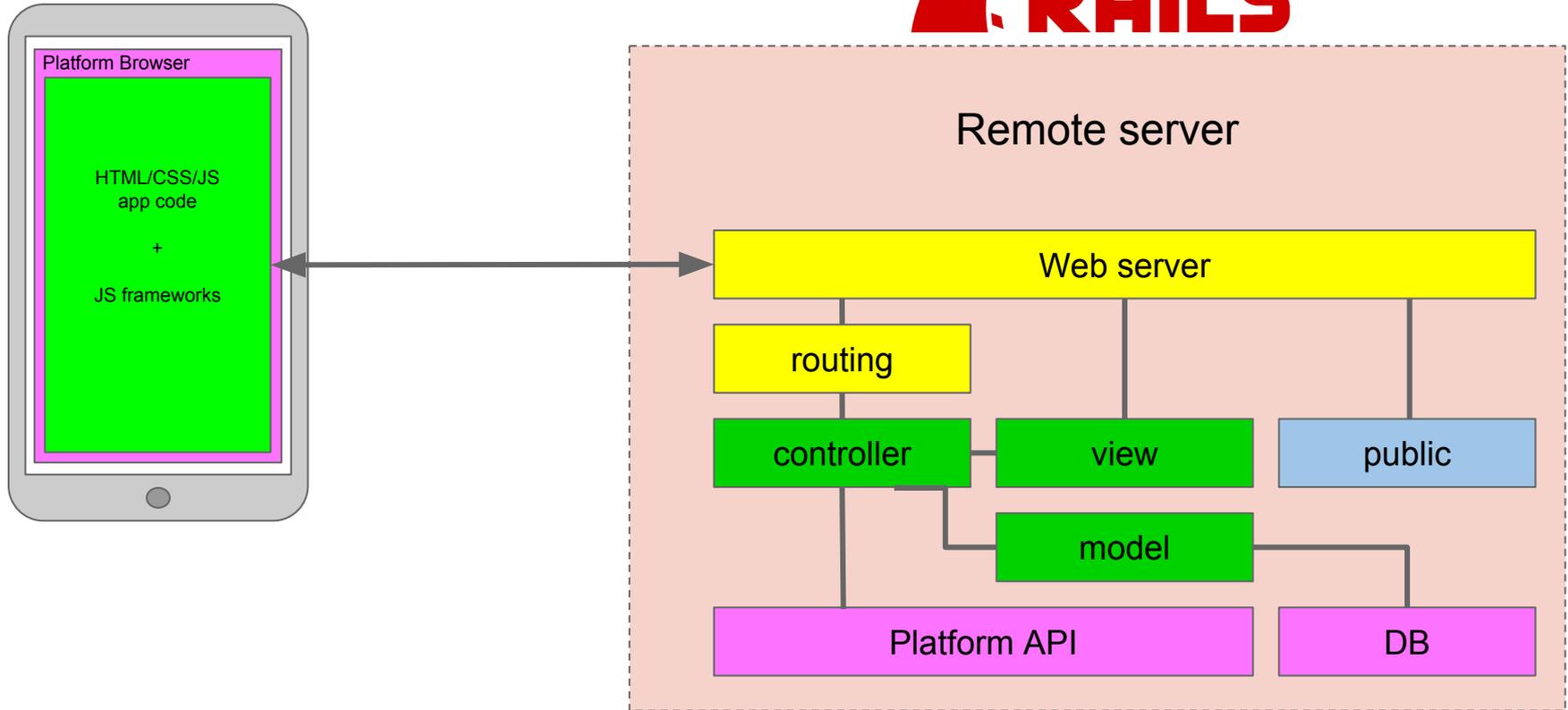
Решение однозначное - выносить часть приложения из WebView !

JXCore или Rhomobile предлагают такое решение путем воссоздания серверного окружения на локальном устройстве - чтобы разработчикам было привычнее и проще. При этом WebView становится легковесной так как содержит только UI, а данные, логика находятся вне WebView и в отдельных потоках.

На сервере имеется большое количество решений и наработано много кода.  
А можно перенести все это на мобильное устройство и получить самодостаточный продукт  
- серверная часть + UI в WebView (или даже в простом browser) ?

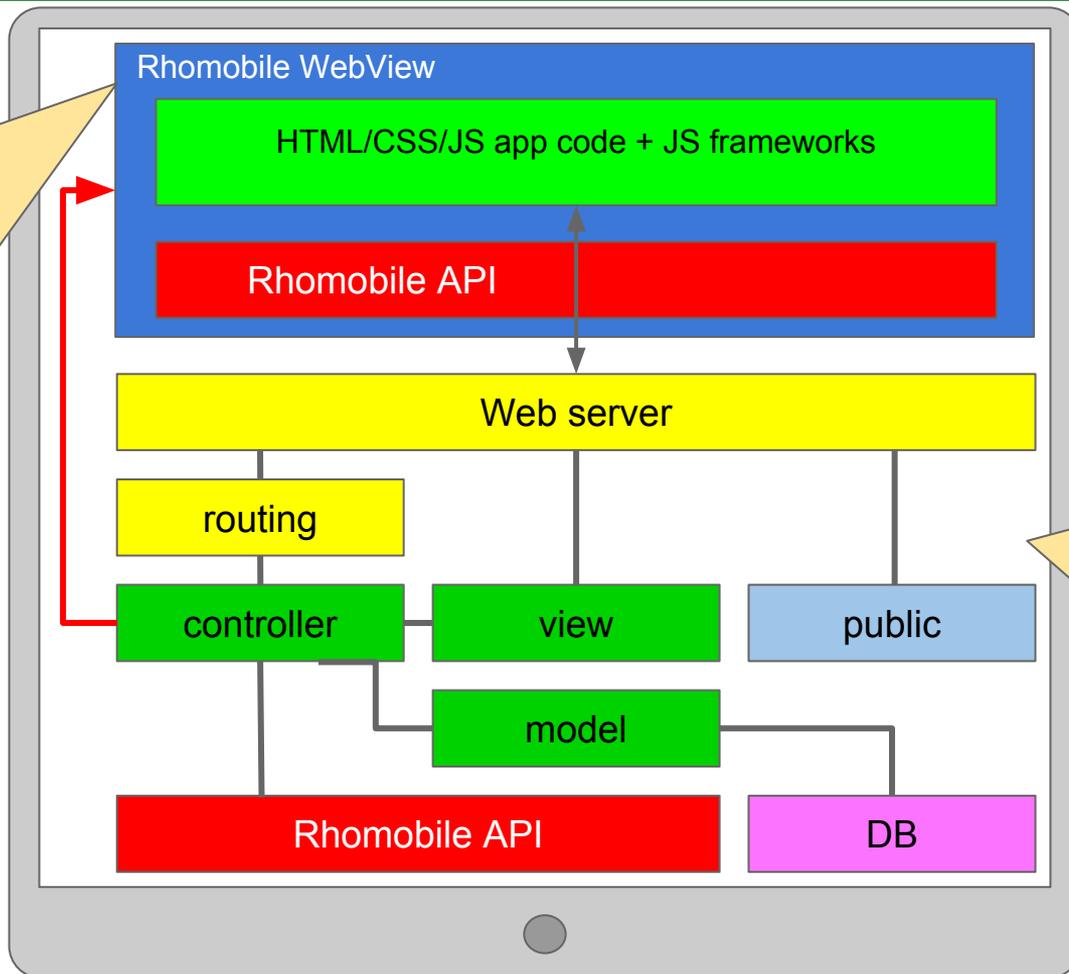


Как это выглядит на примере переноса  
Ruby on Rails функционала



Гибридная архитектура. UI реализуется HTML/CSS/JS контентом в WebView

Разработчик приложения может использовать любой JavaScript UI framework по желанию



**RHODES**

На устройстве мы имеем локальный HTTP сервер с полноценной Ruby VM. Ruby код контроллеров выполняется прямо на устройстве.

WebView с доступом к Node.js API ! Добавьте свой HTML/CSS/JS контент и получите полноценное автономное приложение !

Можно использовать даже нативные Node.js модули !

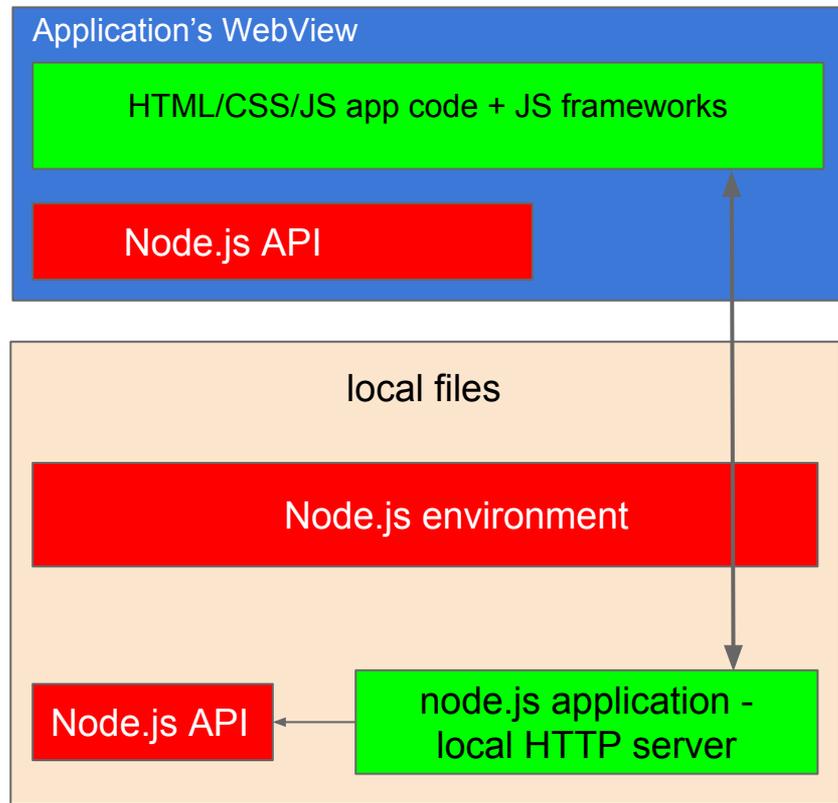
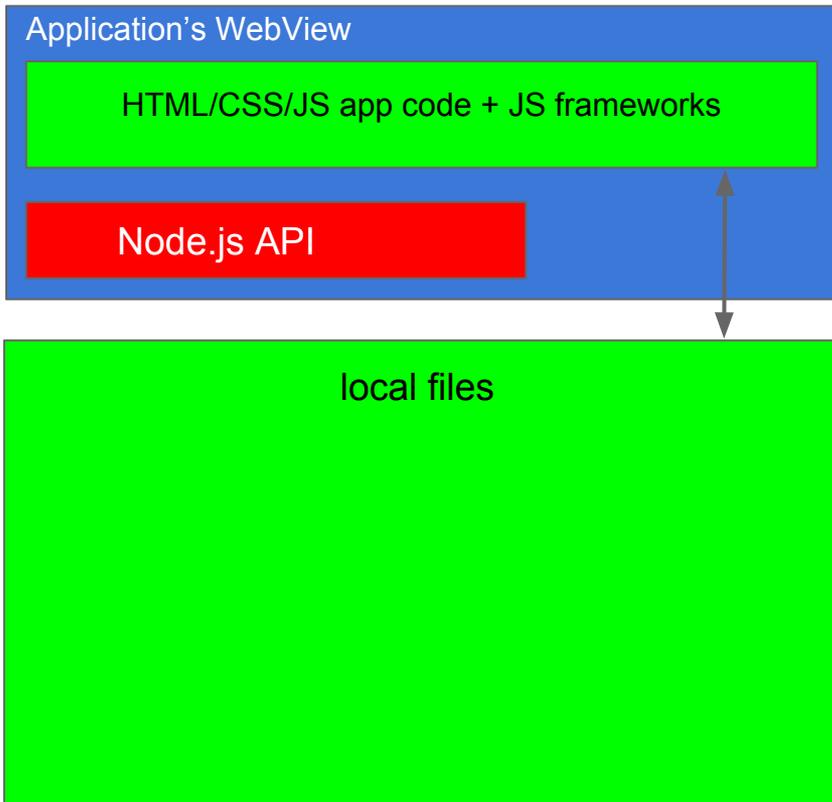
Сравнение решений: <https://habrahabr.ru/post/304574/>



название	<b>NW.js</b>
разработчик	<b>NW.js community</b>
тип	гибридный
исходный код	full  open source
доступность	<b>FREE</b>
сайт	<a href="http://nwjs.io/">http://nwjs.io/</a>
Платформа	Linux, Mac OS X, Windows



название	<b>Electron</b>
разработчик	<b>Electron community</b>
тип	гибридный
исходный код	full  open source
доступность	<b>FREE</b>
сайт	<a href="http://electron.atom.io/">http://electron.atom.io/</a>
Платформа	Linux, Mac OS X, Windows



JXCore - ныне закрытый проект по портированию Node.js на мобильные платформы.

В настоящее время существует ответвление - Thali Project

JXCore FAQ: <http://www.golang.org/jxcore/>

Доля нативных модулей Node.js - около 10%, доля скачивания - около 2%.

Анализ доли нативных пакетов Node.js: <http://www.golang.org/node-gyp-and-node-js-on-mobile-platforms/>

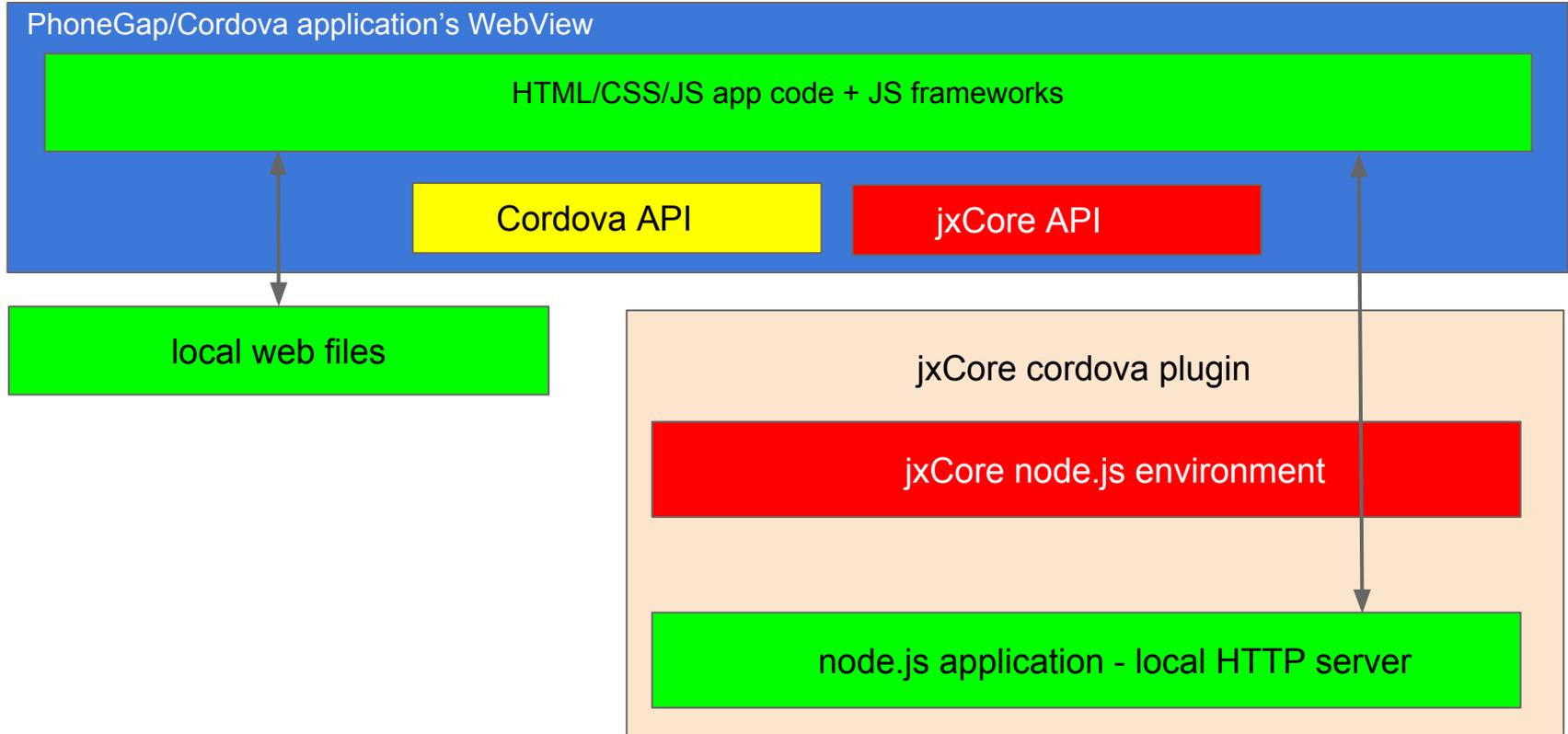
Есть JXCore Cordova plugin: <https://github.com/jxcore/jxcore-cordova>

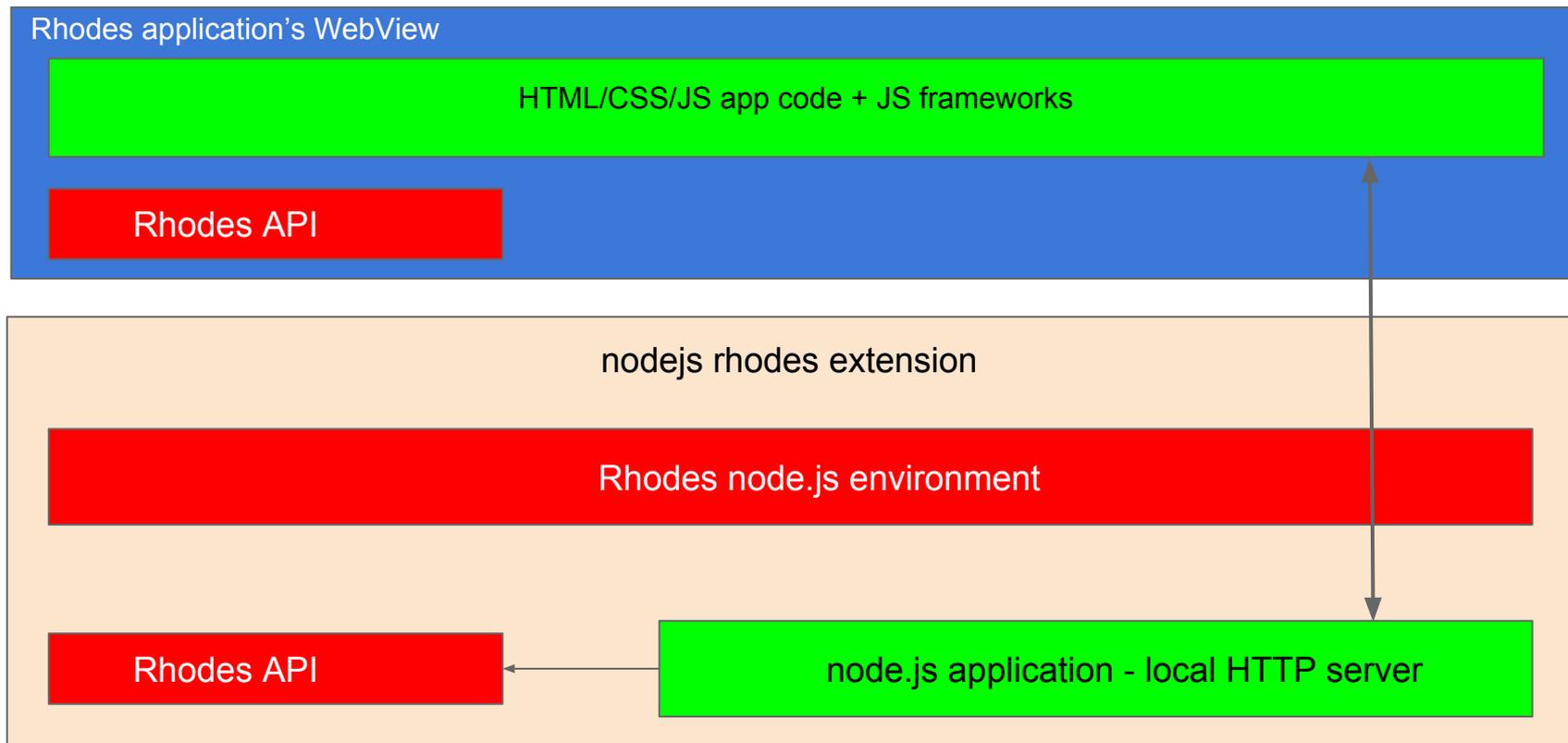


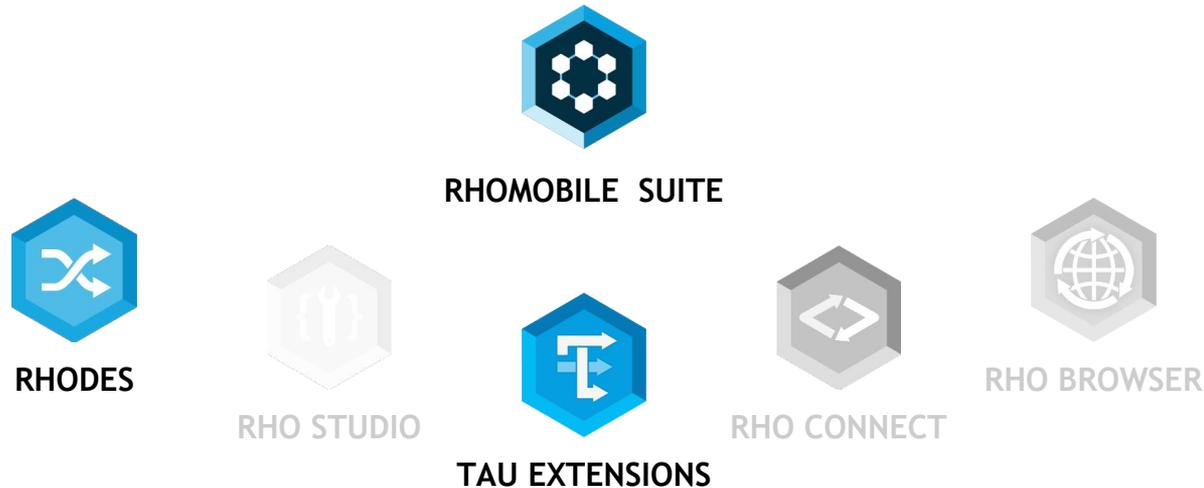
название	<b>JXCore</b>
разработчик	<b>Nubisa</b> (прекращено)
тип	Node.js порт
исходный код	full  open source
доступность	<b>FREE</b>
сайт	<a href="https://github.com/jxcore/jxcore">https://github.com/jxcore/jxcore</a>
Платформа	iOS 

название	<b>Thali</b>
разработчик	<b>Thali</b> (спонсирует Microsoft)
тип	гибридный (Cordova +JXCore)
исходный код	full  open source
доступность	<b>FREE</b>
сайт	<a href="http://thaliproject.org/">http://thaliproject.org/</a>
Платформа	iOS 

JXCore Cordova plugin: <https://github.com/jxcore/jxcore-cordova>







## Rhodes

- **Платформа для разработки кросс-платформенных гибридных и смешанных гибридных приложений**
- Возможна как разработка только на **HTML/CSS/JS**, так и с использованием **Ruby**
- Включает в себя большое количество модулей, включая поддержку Barcode, Printing и т.п.
- Поддерживается **iOS, Android, WinCE/WM, WP**

## Tau Extensions

Набор расширений, включает в себя :

- Crosswalk WebView для Android (замена системного в приложении)
- WebKit WebView для WinCE/WM (замена системного IE в приложении)
- улучшенный OpenSSL модуль для Android
- **реализация Node.js на базе jxCore**

Перед установкой пожалуйста настройте вашу систему. Подробнее:

<http://docs.tau-technologies.com/en/5.5/guide/rhobile-install>

<http://docs.tau-technologies.com/en/5.5/guide/nativesdksetup>

Есть три способа установки - необходимо поставить 5.5.18 или выше :

- Скачать и установить с помощью пакета установки.  
URL для скачивания : <http://rhobile.tau-technologies.com/index.html>
- Установить основной и вспомогательные GEMs (для начала достаточно основного)



```
$ gem install rhodes
```

- Установить исходный код(в приложении придется вручную задавать путь к платформе) и вручную установить нужные GEMs  
Исходный код : <https://github.com/rhobile/rhodes>

Затем необходимо настроить пути к платформенным SDK:

```
$ rhodes-setup
```

Есть три способа установки - необходимо поставить 5.5.18 или выше :

- Скачать и установить с помощью пакета установки.  
URL для скачивания : <http://rhomobile.tau-technologies.com/index.html>



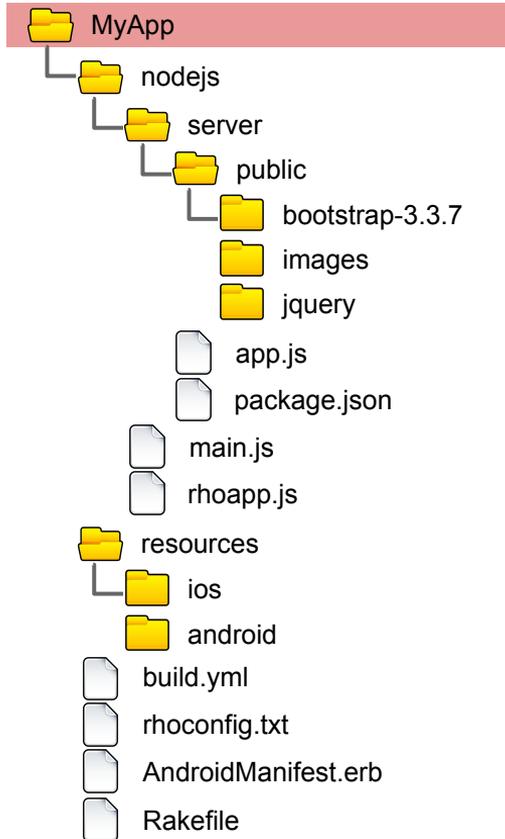
- Установить GEM вручную

```
$ gem install rho-tau-extensions
```

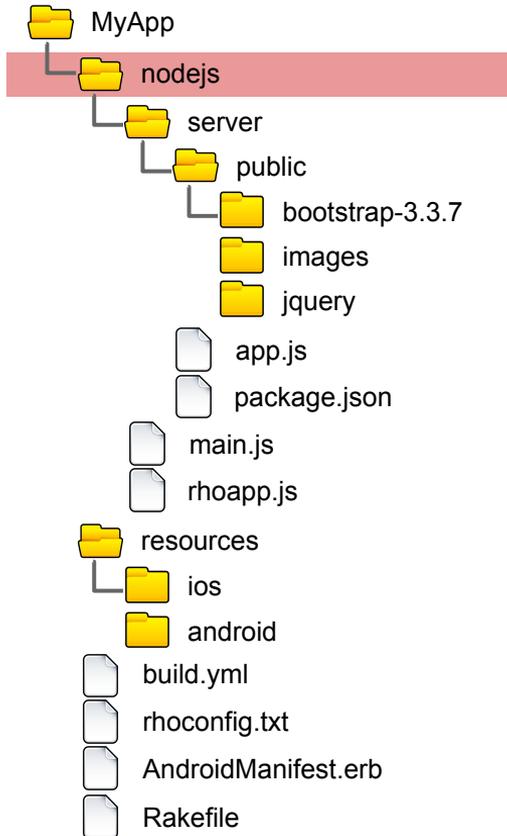
Создадим приложение  
(подробнее [http://docs.tau-technologies.com/en/5.5/guide/creating\\_a\\_project](http://docs.tau-technologies.com/en/5.5/guide/creating_a_project)) :

```
$ rhodes nodejsapp MyApp
```

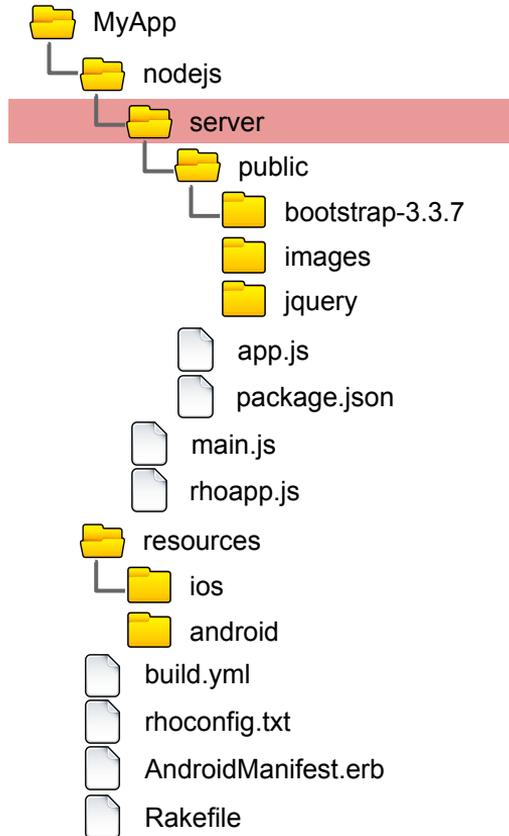
**rhodes** - команда для генерации “болванок”:  
приложения, модели, расширения (extension).  
Сгенерированный код сразу работоспособен.



Мы получили папку MyApp внутри которой находится “болванка” Rhodes приложения

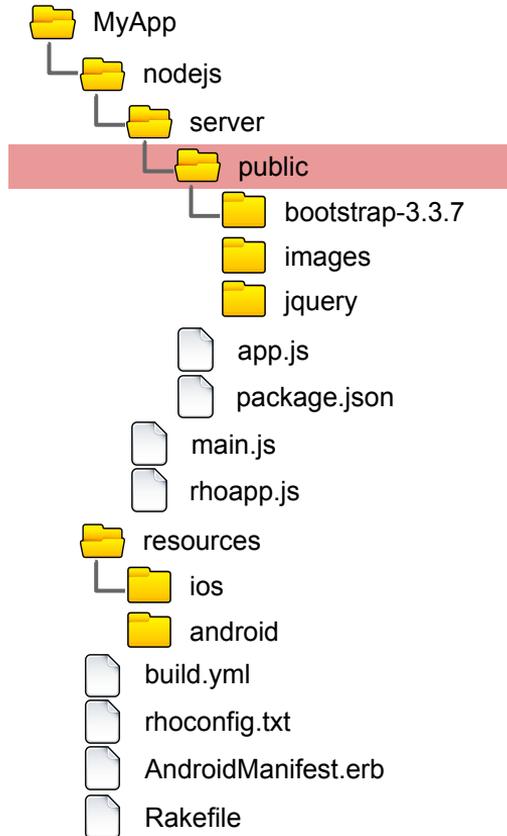


папка **nodejs** содержит код node.js приложения и вспомогательные файлы.

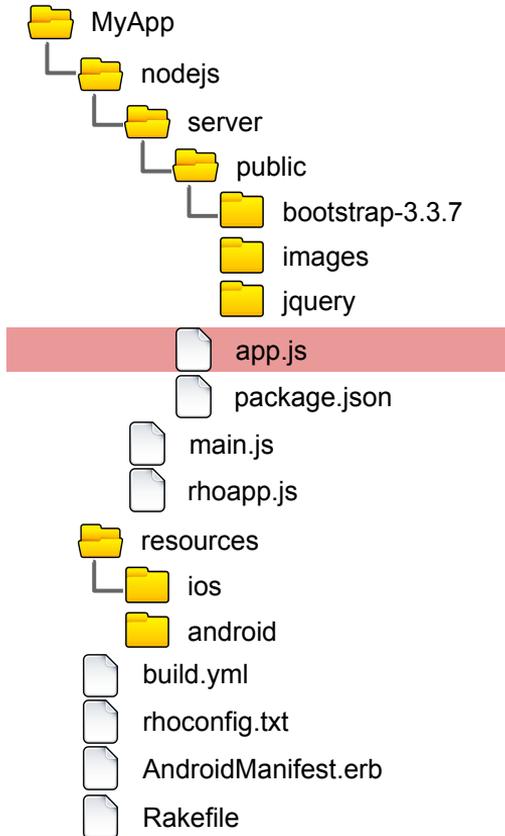


папка **server** содержит непосредственно код приложения и соответствует корневой папке обычного node.js приложения на desktop платформах.

Кроме того предполагается что эта папка является корневой папкой локального http сервера, который должно запустить node.js приложение.



папка **public** содержит статический контент локального http сервера, который должно запустить node.js приложение. В эту папку при сборке будет добавлена папка **api** содержащая \*.js файлы для доступа к Rhodes API из Javascript кода WebView. Изначально папка заполнена bootstrap и jquery файлами, которые могут понадобится для работы контента в WebView.



**app.js** это главный файл node.js приложения - именно этот файл запускается на старте. В этом файле необходимо запустить локальный http сервер на заданном порту и уведомить Rhodes о том что сервер стартовал.

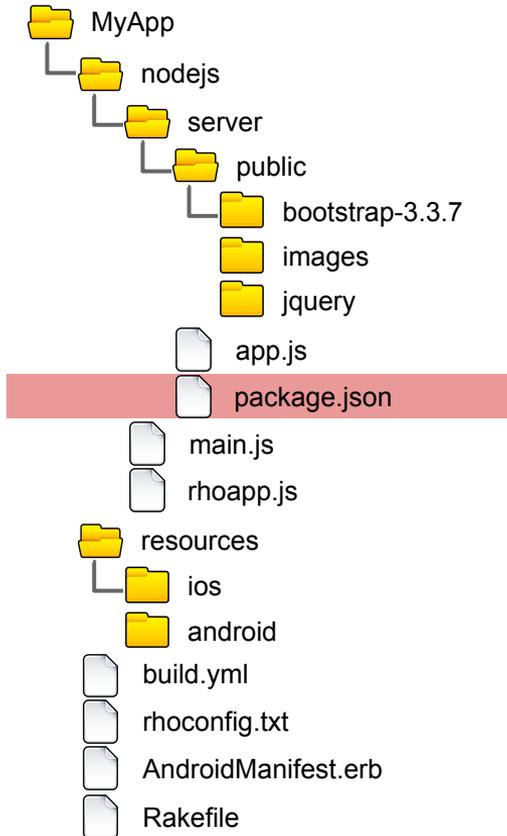
```
var server_port = Rho.System.NodejsServerPort
```

```
var path = require('path');
var express = require('express');
var app = express();
```

```
app.use('/public', express.static(path.join(__dirname, 'public')));
```

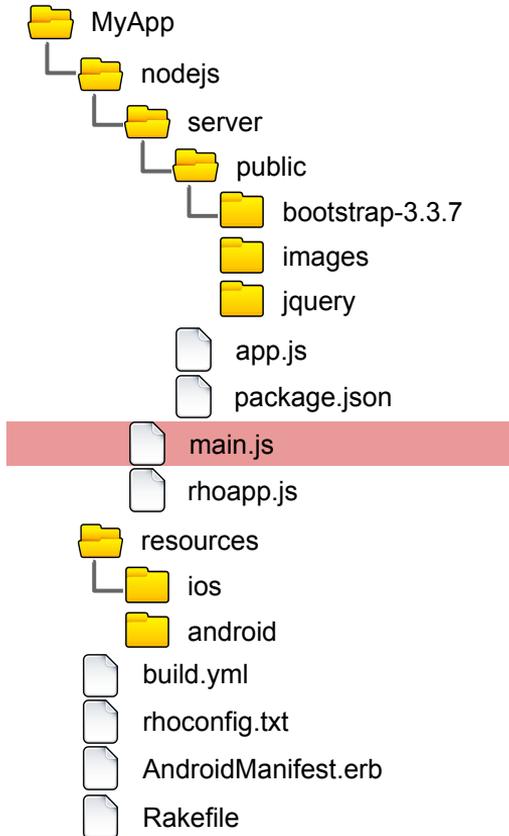
```
app.get('/', function (req, res) {
  res.send('Hello World! (' + Date.now() + ")");
});
```

```
var server = app.listen(server_port, function () {
  Rho.Log.info("Express server is started. (port: "+server_port+)", "Node.js JS");
  // application must be inform RHomobile platform about starting of http server !
  Mobile.httpServerStarted();
});
```

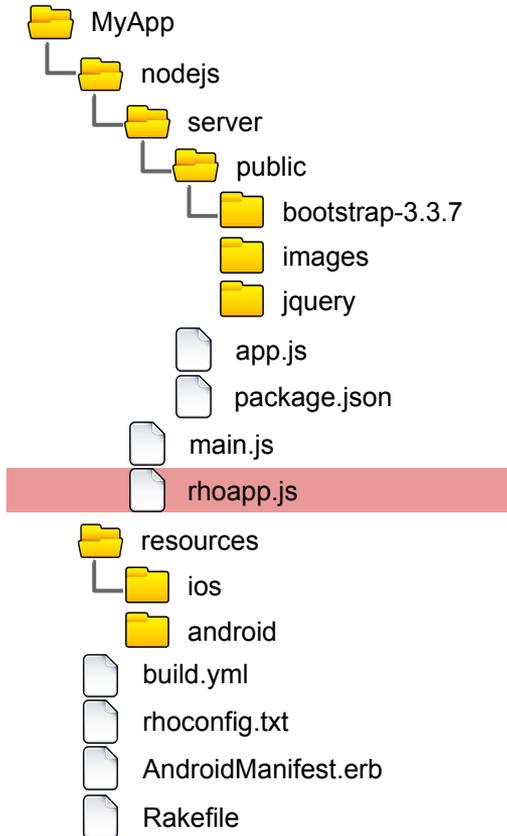


**package.json** - файл описания node.js приложения, соответствует такому в обычном node.js приложении. Содержит список необходимых node модулей.

```
{
  "name": "rhonodejsapplication",
  "version": "0.0.1",
  "engines": {
    "node": "0.10.x",
    "npm": "1.3.x"
  },
  "private": true,
  "scripts": {
    "start": "node app.js"
  },
  "dependencies": {
    "express": "3.3.7",
    "jade": "*"
  }
}
```



**main.js** - код инициализации основного функционала, в том числе инициализация Rhodes API.  
Разработчик не должен менять содержимое этого файла.

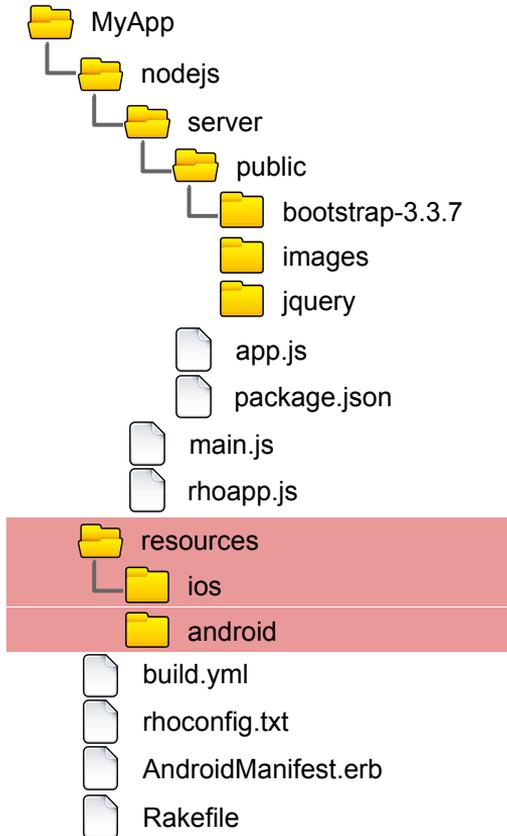


**rhoapp.js** - этот файл выполняется на этапе инициализации Rhodes приложения. По умолчанию содержит код обработки основных событий приложения на платформе - активация, деактивация и т.п. Может меняться разработчиком. Ниже приведен фрагмент кода, который в момент активации приложения создает Native Toolbar (системный UI компонент все WebView)

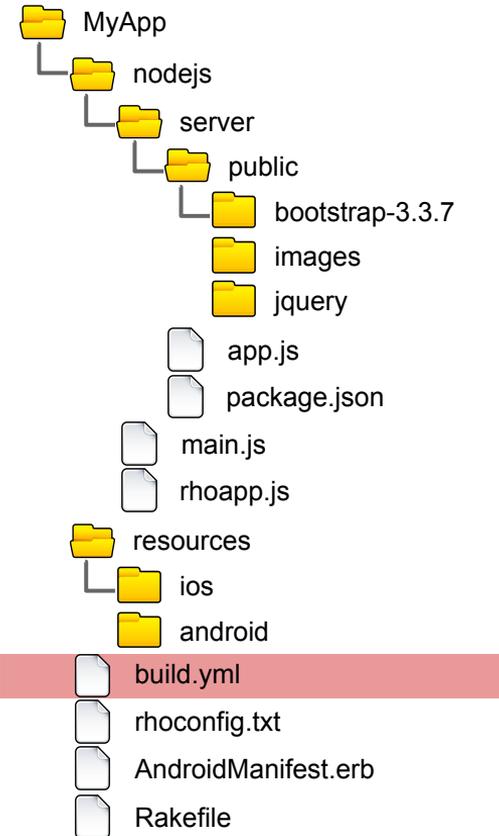
```
...
function onRhomobileApplicationActivated() {
    Rho.Log.info("Node.js event : APP_EVENT_ACTIVATED", "Node.js JS App");

    var native_toolbar = [
        {"action": "back", "icon": "/nodejs/server/public/images/bar/back_btn.png"},
        {"action": "separator"},
        {"action": "home", "icon": "/nodejs/server/images/bar/home_btn.png"},
        {"action": "refresh"},
        {"action": "options", "icon": "/nodejs/server/images/bar/gears.png"}
    ];
    Rho.NativeToolbar.create(native_toolbar);
}
...

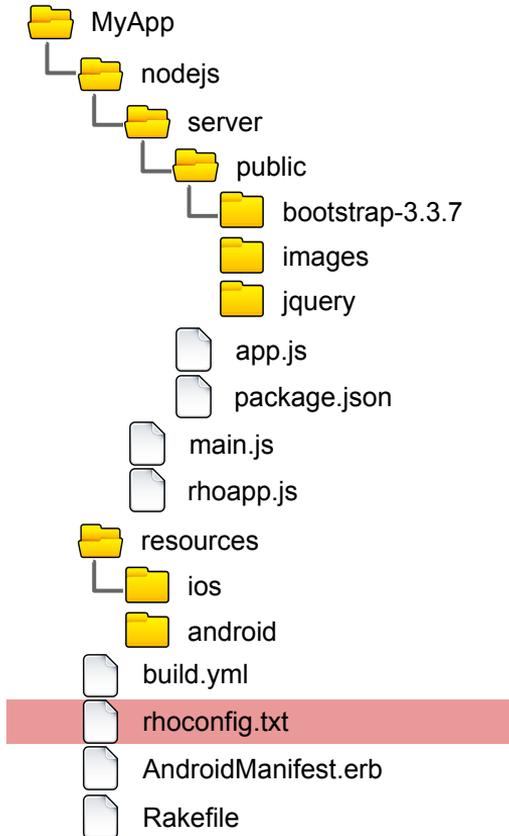
```



папка **resources** содержит иконки, загрузочные изображения, изображения для iTunes и прочие ресурсы необходимые на соответствующей платформе для создания полноценного приложения.



**build.yml** - файл содержит различные настройки приложения, необходимые для сборки или генерации платформенных файлов.



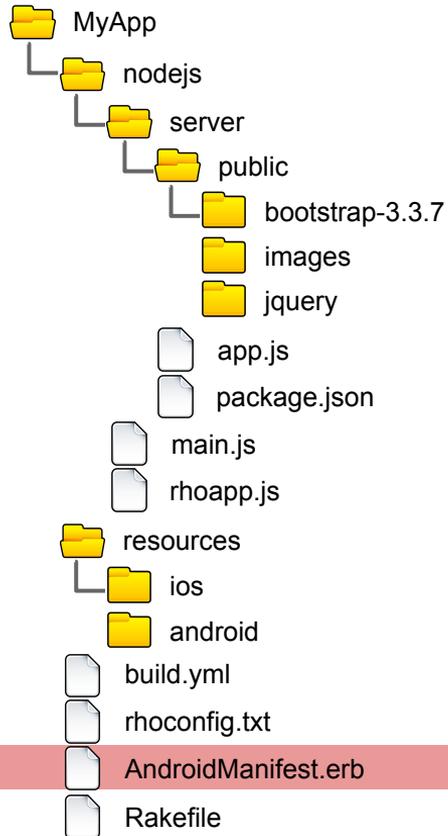
**rhoconfig.txt** - файл содержит различные настройки, используемые во время работы приложения. Файл включается в пакет сборки приложения и в режиме run-time можно считывать и менять значения ключей из этого файла через соотв. API. Содержит, в частности, стартовый URL приложения - то есть URL на локальном HTTP сервере, который будет открыт в WebView при старте приложения. Также содержит настройки вывода Log сообщений

```
# startup page for your application
start_path = '/'

# Rhodes log properties
# log level
# 0-trace, 1-info(app level), 2-warnings, 3-errors
# for production set to 3
MinSeverity = 1

# enable copy log messages to standard output, useful for debugging
LogToOutput = 1

...
```

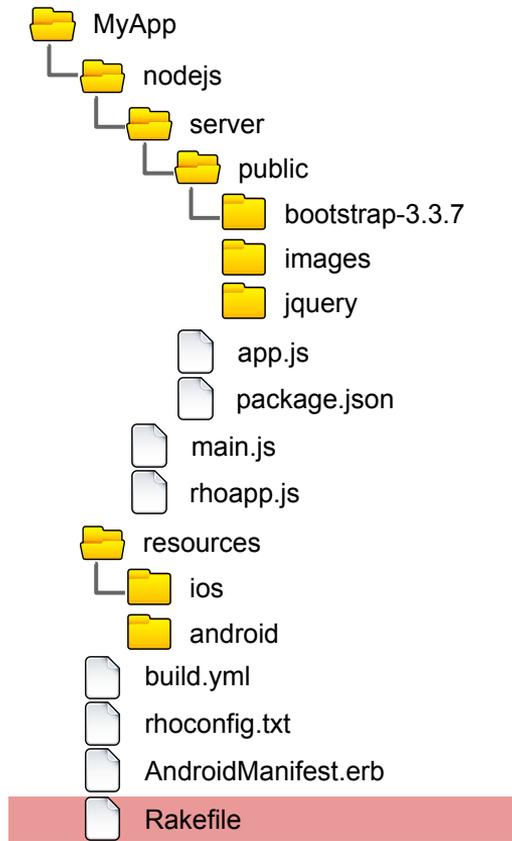


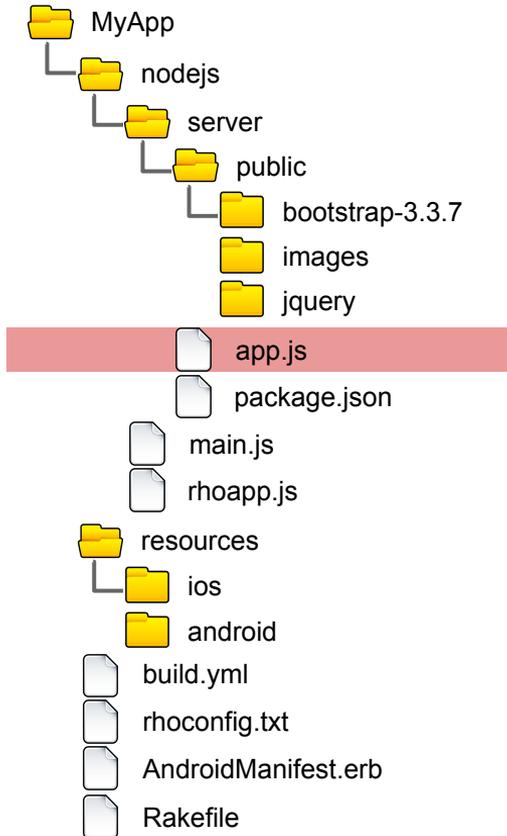
**AndroidManifest.erb** - шаблон для генерации Manifest для Android платформы - можно редактировать для дополнительной настройки.

Подробнее;

[http://docs.tau-technologies.com/en/5.5/guide/build\\_android](http://docs.tau-technologies.com/en/5.5/guide/build_android)

стандартный скрипт для работы **rake** команд





Из любого javascript файла выполняющегося в Node.js, доступно Rhodes API. Для доступа к этому API ничего не нужно делать дополнительно. API инициализируется в момент выполнения `/nodejs/main.js` файла на страте приложения.

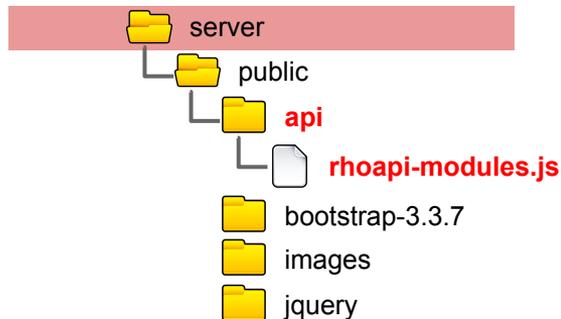
С Rhodes API можно ознакомиться тут :

<http://docs.tau-technologies.com/en/5.5/guide/apisummary>

```
...
```

```
Rho.Log.info("my log message !", 'my_module');
```

```
...
```



Когда приложение развернуто на устройстве, то в папке **/public/** на локальном HTTP сервере появляется папка **api** в этой папке находится JavaScript файл **rhoapi-modules.js** этот файл генерируется на этапе сборки и содержит объявление API Rhodes и всех подключенных extensions. Для доступа к этим API достаточно включить этот файл в ваш HTML.

С Rhodes API можно ознакомиться тут :

<http://docs.tau-technologies.com/en/5.5/guide/apisummary>

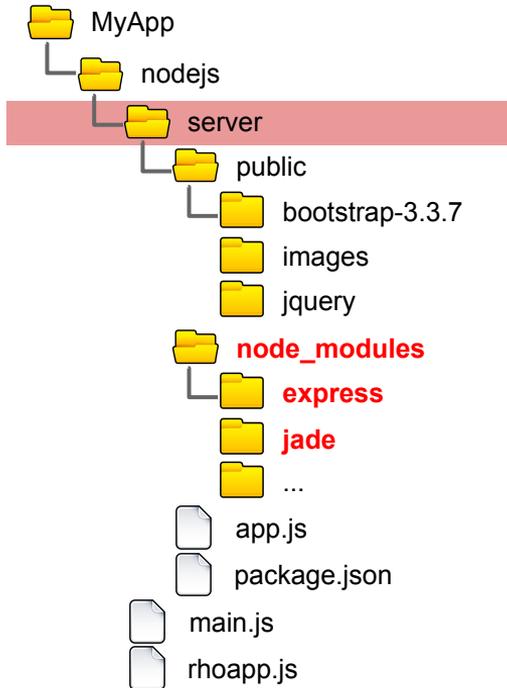
```
<head>
  <script src="/public/api/rhoapi-modules.js" type="text/javascript"></script>
</head>

...

<script type="text/javascript">

function showMessageInLog() {
  Rho.Log.info("my log message !", 'my_module');
}

</script>
```

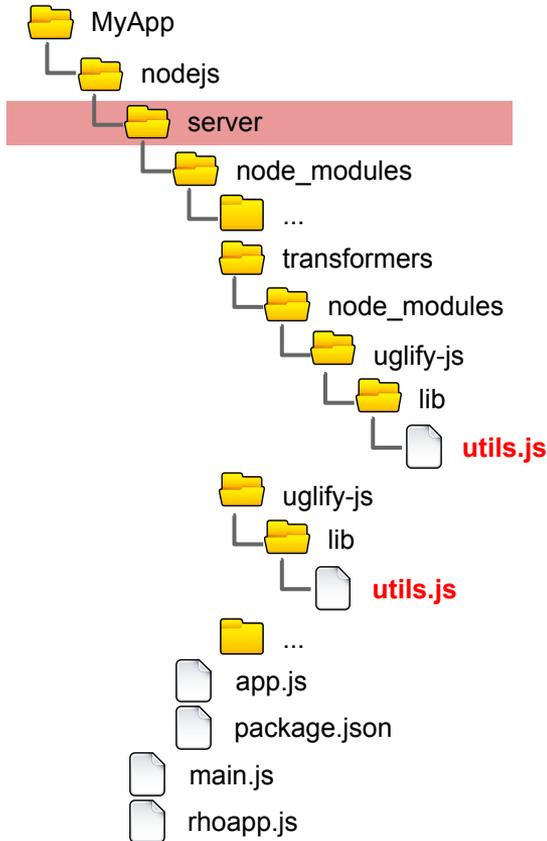


Перед сборкой приложения нам необходимо установить используемые node модули.

Для этого, как и для обычного Node.js приложения - переходим в корневую папку - **nodejs/server** и выполняем стандартную команду :

```
$ npm install
```

После выполнения команды у нас появится стандартный каталог **node\_modules** содержащий используемые модули со всеми зависимостями.



К сожалению в старой версии uglify-js (которая загружается для express.js версии 3.x) содержится код, который не работает в jxCore реализации Node.js. Это известная проблема, которая по словам разработчиков uglify-js исправлена в новых версиях. Необходимо в файле utils.js найти строку

```
if (!(words instanceof Array)) words = words.split(" ");
```

и заменить ее на строку

```
if (!(Object.prototype.toString.call(words) === '[object Array]')) words = words.split(" ");
```

Причем это необходимо сделать два раза :

1. непосредственно в модуле uglify-js
2. внутри модуля transformers - в его локальной копии модуля uglify-js

Или можно просто загрузить готовые модули из одного из наших примеров, где они сразу исправленные лежат в репозитории : [https://github.com/tauplatform/NodeJSApp/tree/master/nodejs/server/node\\_modules](https://github.com/tauplatform/NodeJSApp/tree/master/nodejs/server/node_modules)

Запускаем наше приложение на iPhone Simulator:

```
$ rake run:iphone
```

Кроме того, можно генерировать XCode проект, чтобы открыть его в XCode и собирать/запускать приложение на симуляторе и устройстве.

Генерация XCode проекта :

```
$ rake rake build:iphone:setup_xcode_project
```

Созданный проект располагается:



Подробнее: [http://docs.tau-technologies.com/en/5.5/guide/build\\_ios](http://docs.tau-technologies.com/en/5.5/guide/build_ios)

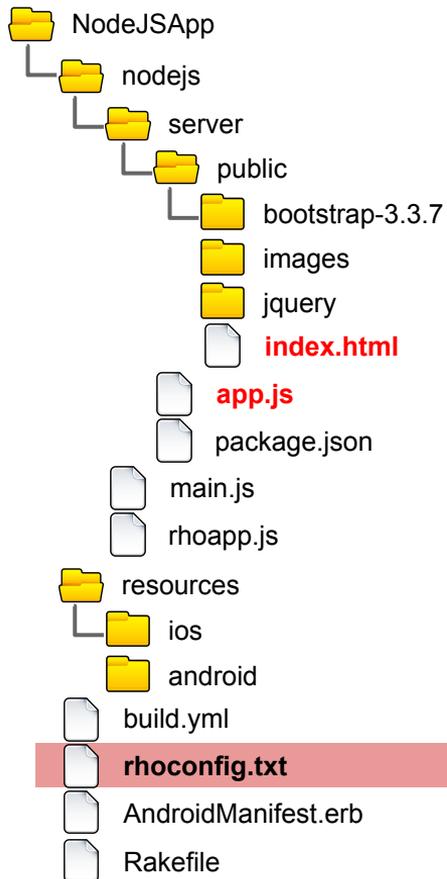
Запускаем наше приложение на Android Simulator:

```
$ rake run:android
```

Запускаем приложение на подключенном устройстве :

```
$ rake run:android:device
```

Подробнее: [http://docs.tau-technologies.com/en/5.5/guide/build\\_android](http://docs.tau-technologies.com/en/5.5/guide/build_android)



Исходный код - <https://github.com/tauplatform/NodeJSApp>

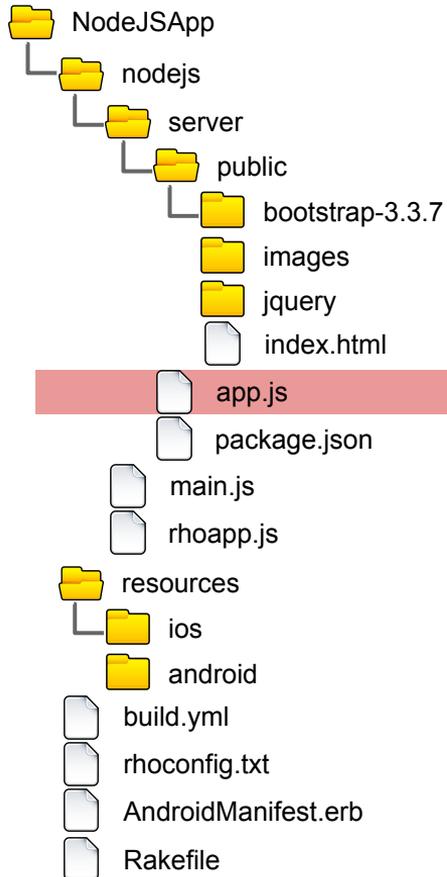
В файле конфигурации **rhoconfig.txt** мы определили стартовый адрес: **/public/index.html**

это URL на нашем локальном HTTP сервере, который создается в **/app.js**

Это значит что после создания локального HTTP сервера в WebView будет открыт URL **http://127.0.0.1:port/public/index.html**

```
# startup page for your application
start_path = '/public/index.html'

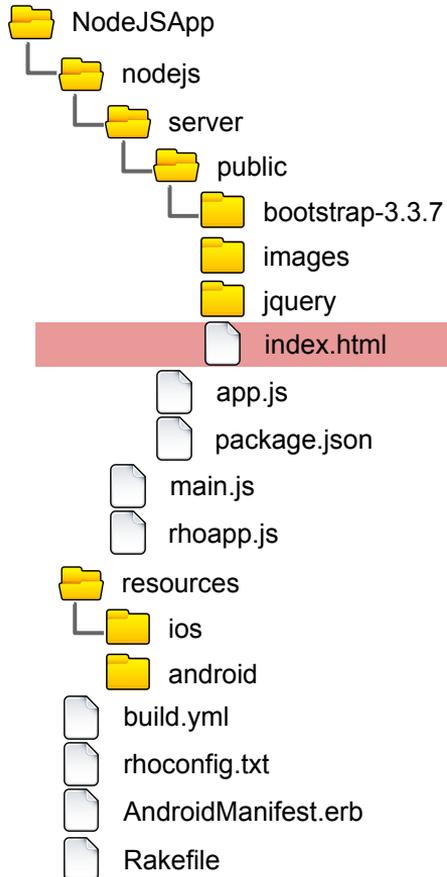
...
```



Исходный код - <https://github.com/tauplatform/NodeJSApp>

В файле node.js приложения мы регистрируем глобальную функцию, которая выводит сообщение в лог и запускает javascript код в текущем WebView

```
...  
global.myfunc = function() {  
  Rho.Log.info("$$$$$$$$$$$$$$$$ RUN NODEJS CODE !!!", 'Node.js JS');  
  Rho.WebView.executeJavascript("showAlert();");  
};  
...
```



Исходный код - <https://github.com/tauplatform/NodeJSApp>

В файле index.html мы имеем активный элемент при нажатии на который срабатывает функция **onExecuteNodejsCode()**. Кроме того мы имеем функцию **showAlert()**, которую будем вызывать из node.js кода через **Rho.WebView.executeJavascript()**

Таким образом при нажатии на элемент произойдет следующее:

1. сработает JS код в WebView, который вызовет JS код в контексте node.js
2. JS код в node.js в свою очередь произведет вызов JS из WebView и мы увидим alert.

```
<script type="text/javascript">

function onExecuteNodejsCode() {
    Rho.Nodejs.executeJavascript("myfunc();");
}

function showAlert() {
    alert("Alert !!!");
}

</script>
```

JXCoreTest это более комплексный пример. За основу взят готовый пример node.js приложения с использованием express.js, jade и sqlite3. Оригинальный пример **nodeling** взят отсюда:

<https://github.com/jasonbrice/nodeling>

в этом примере имеются sqlite DB файлы, с которыми производится работа - показ содержимого, редактирование и т.п.

Готовый к запуску код приложения находится тут:

<https://github.com/tauplatform/JXCoreTest>

сделаны необходимые исправления и загружены необходимые node\_modules

Для того чтобы это приложение заработало в Rhomobile, пришлось сделать несколько изменений.

1. пропатчить **uglify-js**
2. поменять пару **jade** файлов - возможно в исходном примере изначально были эти ошибки
3. поменять путь по которому осуществляется открытие DB файлов. Дело в том что для node.js приложения текущий каталог это каталог запуска, а на мобильном устройстве это не так. Поэтому надо задавать путь полностью. Корневой каталог node.js приложения можно узнать вызвав **process.cwd()**

Все описанные изменения произведены в следующем commit:

<https://github.com/tauplatform/JXCoreTest/commit/b60f8f11675f6a24b1be2c90cd459ca8d7fdab39>

По сути node.js мобильное приложение мало чем отличается от обычного desktop node.js приложения. Поэтому разработчик может создать обычное node.js приложение отладить его на desktop

обычными средствами и потом перенести в папку `/nodejs/server/` Rhomobile приложения.

Какие отличия:

1. немного отличается создание локального сервера - нужно запросить порт и уведомить когда сервер будет готов.
2. Вместо desktop browser у вас будет WebView на мобильном устройстве - другой размер, есть незначительные отличия в поддержке стандартов - рекомендуем пользоваться соотв. JS UI frameworks. В принципе можно открыть ваш сервер и в мобильном browser с мобильного устройства, тогда эти отличия не важны
3. В вашем Javascript коде (как в node.js, так и выполняемом в WebView) доступно Rhodes API (доступ к различному функционалу устройства, например к камере или распознаванию barcode). Также вы можете создавать свои расширения путем разработки native extensions и реализованное вами API также будет доступно в любом Javascript коде как и Rhodes API. Подробнее: [http://docs.tau-technologies.com/en/5.5/guide/native\\_extensions](http://docs.tau-technologies.com/en/5.5/guide/native_extensions)
4. на устройстве другое расположение файлов вызванное требованиями платформы. Мы рекомендуем все user файлы хранить в каталоге **Rho.Application.userFolder**. Этот каталог доступен на запись и соотв. образом будет помещен в backup (актуально для iOS) и т.п.

Список стандартных модулей :

<http://docs.tau-technologies.com/en/5.5/guide/apisummary>

Работа с RestAPI удаленного сервера из Ruby и JS кода в Rhodes:

[http://docs.tau-technologies.com/en/5.5/guide/web\\_services](http://docs.tau-technologies.com/en/5.5/guide/web_services)

Как расширять функционал создавая новые нативные расширения Rhodes:

[http://docs.tau-technologies.com/en/5.5/guide/native\\_extensions](http://docs.tau-technologies.com/en/5.5/guide/native_extensions)

Как легко синхронизировать данные с удаленным сервером ?

<http://docs.tau-technologies.com/en/5.5/guide/synchronization>

Как автоматически менять контент на устройствах без пересборки приложения ?

<http://docs.tau-technologies.com/en/5.5/guide/live-update>

Где можно посмотреть примеры использования Rhodes API ?

<https://github.com/tauplatform/rhodes-system-api-samples>

<https://github.com/tauplatform/kitchensink>

<http://tau-technologies.com>

