

Хочется странного - web UI на Java для desktop приложений

Юрий Артамонов
CUBA Platform



1.

ЗАЧЕМ ?

Кому вообще нужен Desktop в 2018 ?



Что такого есть в Desktop приложениях, чего нет в Web?

(ну или почти)

Работа с локальными устройствами и файлами

- » Обработка данных на клиенте
- » Доступ к принтерам, сканерам, телефонам, кард-ридерам
- » Интеграция с ОС

Независимость от сервера

- » Работа оффлайн
- » Отказоустойчивость
- » Локальный кэш данных

Наш кейс - Desktop-клиент для продукта Sherlock Taxi

Доступ к локальным устройствам

Интеграция с телефонией на рабочих местах.

Локальный кэш данных

Большие объемы справочных данных, география, аккаунты и т.д.

Отказоустойчивость

Обновление всех серверов без остановки приёма заказов, оффлайн.

Enterprise - он такой

The screenshot displays the Enterprise software interface, which is divided into several sections:

- Top Navigation Bar:** Includes links for Job management, Controllers, Customer service, Driver & Fleet Management, Pricing, Common Data, Administration, and Help.
- Job management Section:**
 - Active Circuit Group:** Shows 177 jobs in progress, 824 prebooks, and 8 issues.
 - Job Amend/Info:** Buttons for Amend, Create Note, Export, and Unlock Job.
 - Locate:** Buttons for Reallocate, Set Status, Finish Job, Unallocate, Send Job, and Ban Driver.
 - Execution Control:** Buttons for Reallocate, Set Status, Finish Job, Unallocate, Send Job, and Ban Driver.
 - Communication:** Buttons for Driver, Client, and SMS.
 - Tools:** Buttons for View Customer and Driver Jobs.
- Search Filter:** Includes fields for Account, Clients Individual, Journey Type, Status, Passenger on board, and Payment Type.
- Table:** A table listing jobs with columns for Priority, Number, Service, Driver, Allocated Date, Job Date, Route, Status, and Count. The table shows various routes and their corresponding status and counts.
- Customer Service Section:**
 - New Booking:** Fields for Customer, Account, Phone number, and Service details.
 - Locations:** Fields for Address, P/U, Notes, Airport, Flight #, and Dep. time.
 - Payment Type:** Fields for Loyalty card and Electronic Voucher.
- Search / edit address:** A pop-up window with a map of Beirut. It includes fields for Required details (Full address, Arabic address, Region, Latitude, Longitude) and Optional details (Building, Company name, Street, Town/Neighbourhood, Postcode, District/City, Country). The map shows the location of the BLOM Bank in Beirut.
- Details / Notes:** A section showing details for a specific job, including the number, type, date, course, and price.

2.

**Все нормальные пишут
на JavaFX !**

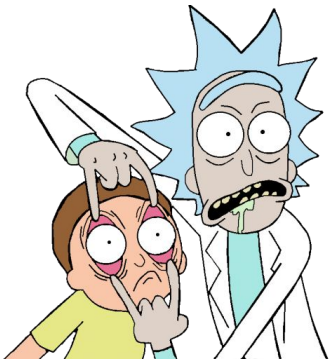
Ну есть же готовые UI библиотеки для Java на Desktop



Ну а если серьёзно?

- » JavaFX движется в сторону Web.
- » Swing умирает, а SWT не развивается.
- » Тяжело писать responsive UI и анимации.

! Тонны наработок для HTML/CSS не могут быть использованы в Java фреймворках для настольных систем.





3.

Electron



Фреймворк для разработки настольных приложений на JS/HTML/CSS.



Electron в двух словах

Круто

Полноценный Chromium, интеграция с ОС, нативные меню, уведомления, все новые фишки HTML5 и CSS3.

Не круто

Node.js в довесок.

Пишем всё приложение на JavaScript.

Целый огромный Chromium!



Какие варианты?

- » Пишем фронтенд на JS
- » Компилим свой фронтенд в JS
- » Берём какой-нибудь server-side фреймворк

Если у вас уже есть фронтенд - остаётся только адаптировать приложение.

Ну мы же тут Java разработчики

Vaadin
JSF
ZK

ИЛИ

Kotlin.js
Scala.js
JSweet

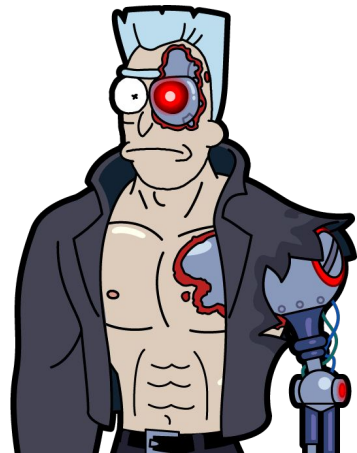


Как это всё завести

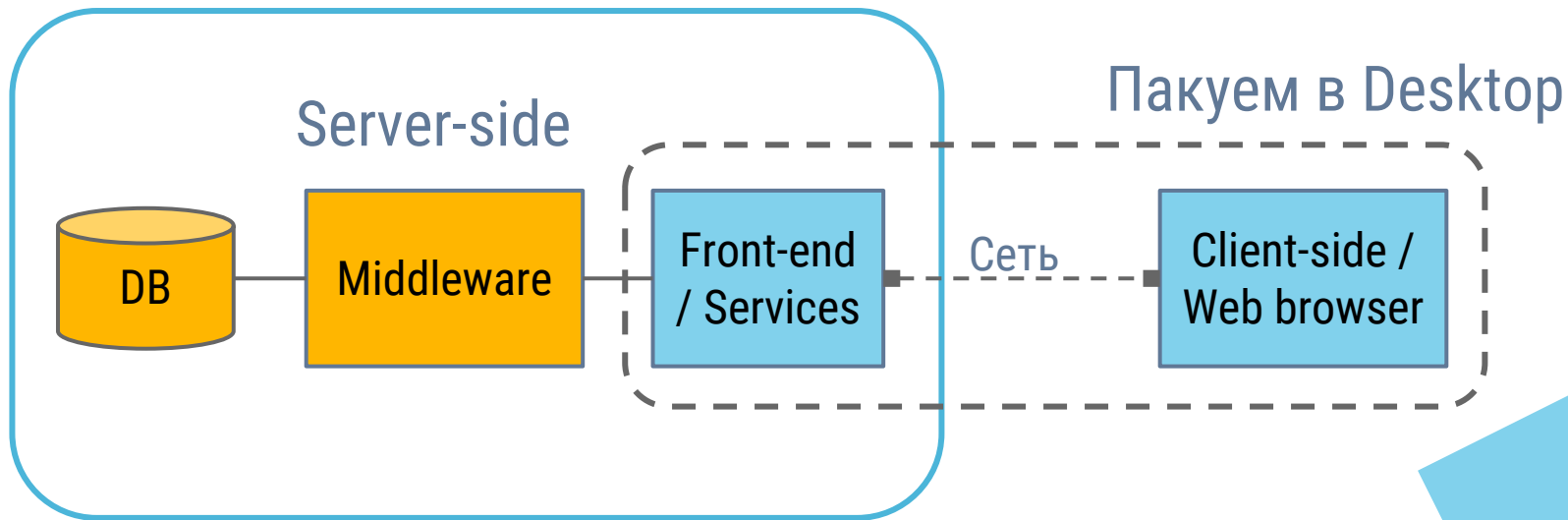
Собираем Fat
JAR вместе с
веб-сервером

Стартуем
его из JS

Пакуем
всё
вместе



Архитектура приложения



Vaadin Framework, что ты такое?

- » Сервер-ориентированный Java framework
- » Хранит состояние UI на сервере
- » В браузере бежит JS часть виджетов
- » Можно писать на Java без JavaScript
- » OpenSource, Apache 2
- » Легко разрабатывать RIA / enterprise UI

Тут будет
демо

Какие проблемы?

- » Стартует медленно
- » Качает статику по сети
- » Использует сеть для логики
- » Передаёт куки и заголовки
- » Неудобный процесс разработки

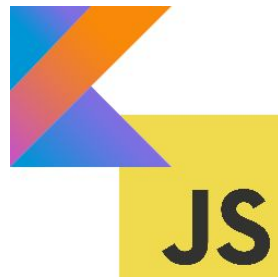
Улучшаем коммуникации: двусторонний WebSocket

Тут будет
демо

Если у вас уже есть приложение:

- » Заменяем AJAX на WebSocket
- » Избавляемся от HTTP заголовков
- » Снижаем время на транспорт до 1мс !
- » Чрезвычайно легко в server-side фреймворках

<https://vaadin.com/docs/framework/advanced/advanced-push.html>



Kotlin.JS за 60 секунд

- » Транспайлер Kotlin в JavaScript (ECMAScript 5.1)
- » Можно переиспользовать код на Kotlin на backend / frontend (с версии 1.2)
- » Лёгкий interop с JavaScript
- » Куча типизированных JS библиотек даром с <https://github.com/Kotlin/ts2kt>

Улучшаем коммуникации радикально: IPC

Если приложение пишется с нуля:

- » Открываем *nix socket / named pipe
- » Используем IPC вместо AJAX
- » Работает мгновенно!

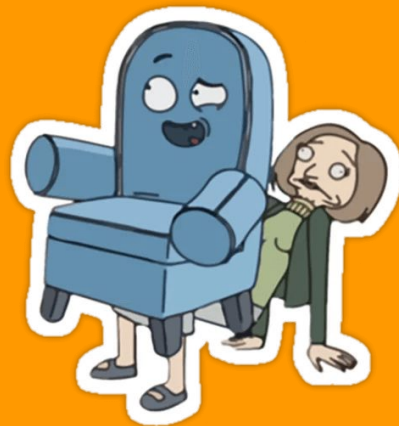
Тут будет
демо

<https://nodejs.org/api/net.html>

4.

Ерунда какая-то получается!

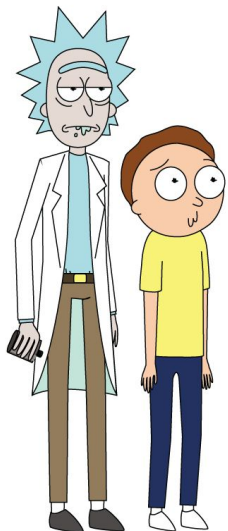
Это ваше поделие небось медленное и глючное



Память на примере Vaadin Demo

	Min	Демо- Приложение	Небольшой Dev Tool
Java	10 MB	120 MB	450 MB
Electron	85 MB	100 MB	150 MB
Всего	95 MB	220 MB	600 MB

[!] Эти цифры сильно зависят от проекта

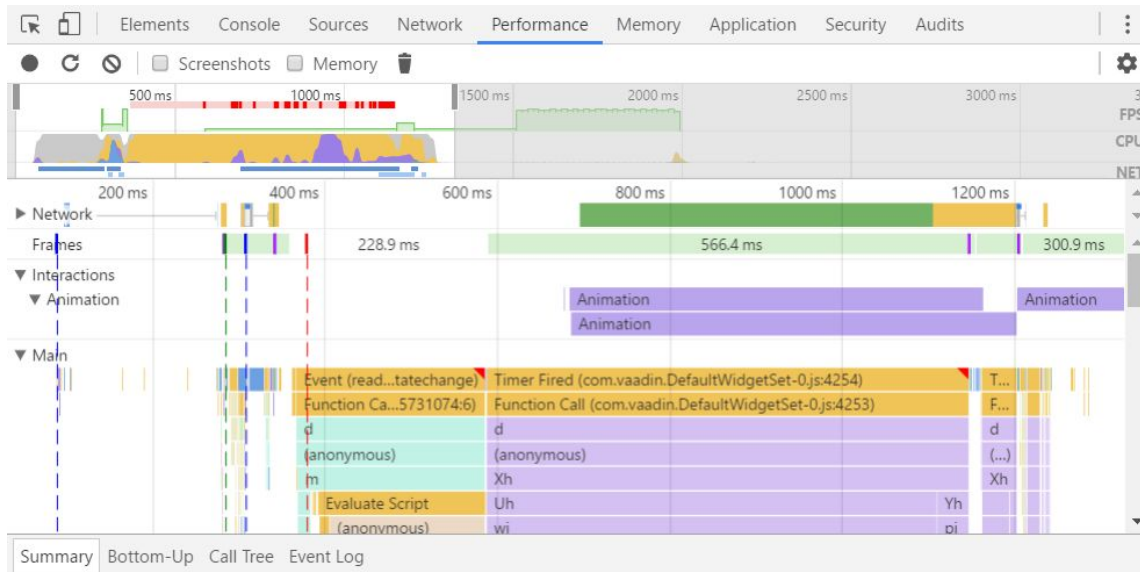


Рендеринг на примере Vaadin Demo

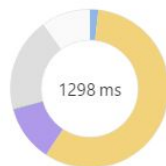
	Reload	XHR	Web Socket
Сеть	200 ms	15 ms	1-2 ms
Обновление DOM	1100 ms	90 ms	
Ждём на старте	1300 ms		

[!] Эти цифры сильно зависят от проекта





Range: 30 ms – 1.33 s



22.9 ms Loading
 746.9 ms Scripting
 148.4 ms Rendering
 2.4 ms Painting
 248.5 ms Other
 128.8 ms Idle

[!] Эти цифры сильно зависят от проекта



А реально это кто-то использует?



Да! Мы выпустили Development Tool на этом всём!

Тут будет
демо

Хаки и полезные наработки

Статические ресурсы

Отдаём статические ресурсы прямо с диска в обход сети при помощи перехватчика запросов.

```
const {protocol} =  
require('electron')
```

WebSocket

Используем WebSocket протокол для ускорения коммуникаций до 1ms плюс избавляемся от HTTP мусора.

IPC

Двусторонний обмен сообщениями (unix socket / named pipe). И даже без сети!

А что ещё посмотреть

Экосистема

Пакеты,
инсталляторы,
автообновление.

См. [electron-builder](#)

Dev Tools

Отличный
профайлер, UI
дебаггер и всё-всё-
всё как у веб-
разработчиков.

Gradle Node.js

Сборка всего
приложения из
уютного Gradle при
помощи плагина:

com.moowork.node

Ссылки

Minimal

<https://github.com/cuba-labs/java-electron-tutorial>

Playground

<http://github.com/jreznot/electron-java-app>

Kotlin Hybrid Desktop

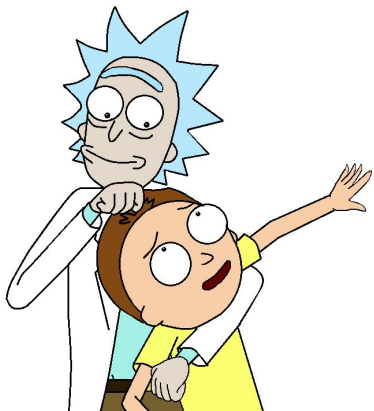
<https://github.com/jreznot/kotlin-hd>

Спасибо!

Вопросы?

Twitter

» @Yuriy_Artamonov



CREDITS

Special thanks to all the people who made and released these awesome resources for free:

- » Presentation template by [SlidesCarnival](#)