

Прекратите бояться SEO

Серверный рендеринг JavaScript приложений



БИБОСС TeamLead Салих Фахрутдинов

Поговорим

- SEO и JavaScript
- Переход с CSR к SSR в JS
- Существующие решения
- SSR и VueJS
- Итоги

Хватит бояться SEO



Основные страхи

- JS это про интерактив, JS это не про SEO
- JS нельзя использовать в тех местах, где нужно это показывать поисковику
- Нельзя писать сайты на Vue Angular React, чтобы они индексировались
- SPA на контентный сайт? не нельзя
- Так не делают, js не для этого и т.п.

Хватит это терпеть

и писать только бэк-офис* на spa

*** бэк-офис - это админки, дэшборды и тп, что не видит поисковая машина**

JavaScript & SEO



Поисковые машины



2014

Яндекс

2015

Google

- Googlebot использует для рендеринга Google Chrome 41 (ему три года)
- Googlebot не настоящий браузер
- Google пообещали в 3 квартале этого года дать полноценнее поддержку js - подтверждений не нашел

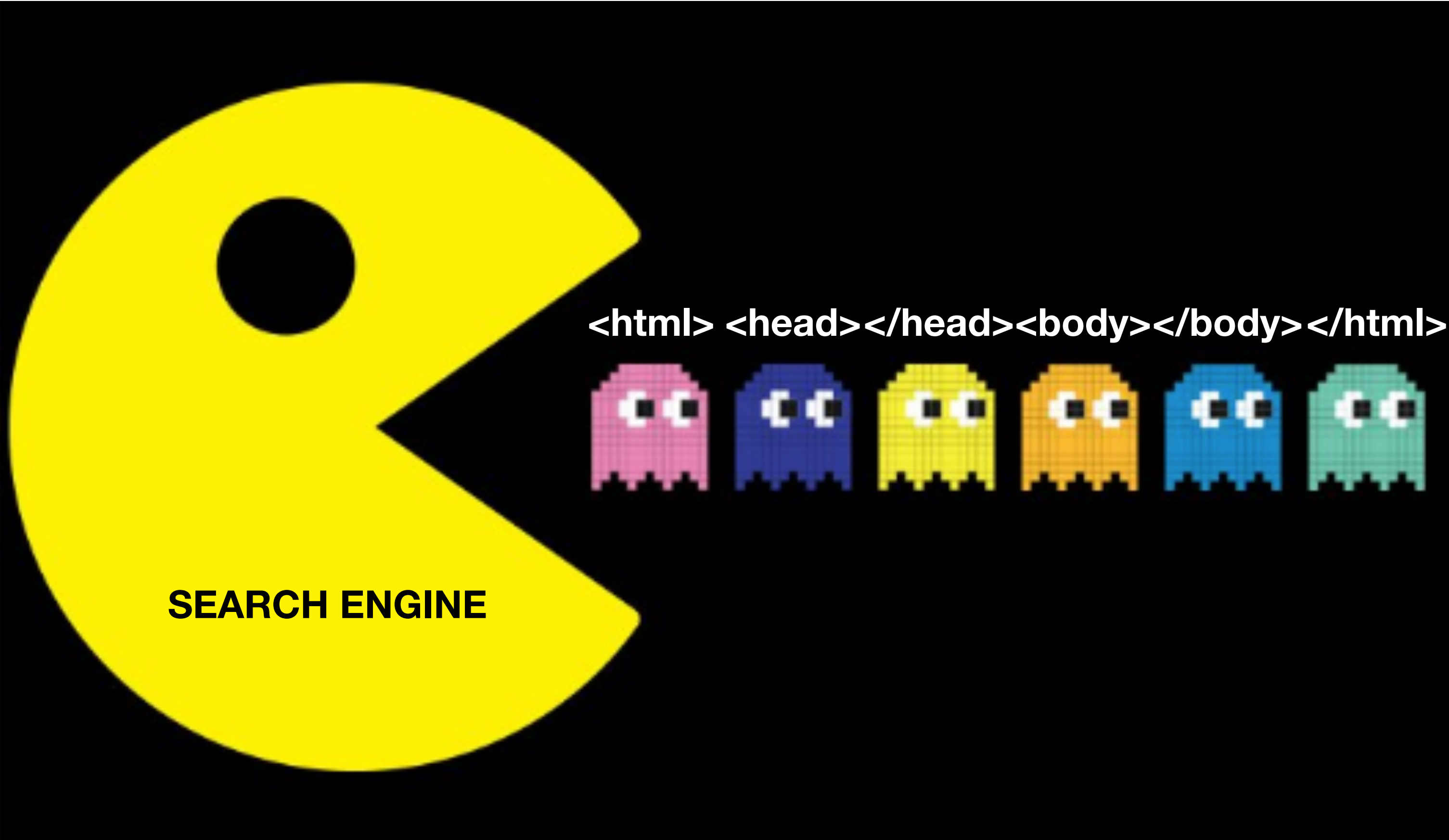
Yandex

 **Яндекс** Вебмастер

[Вебмастер](#) / Индексирование AJAX-сайтов

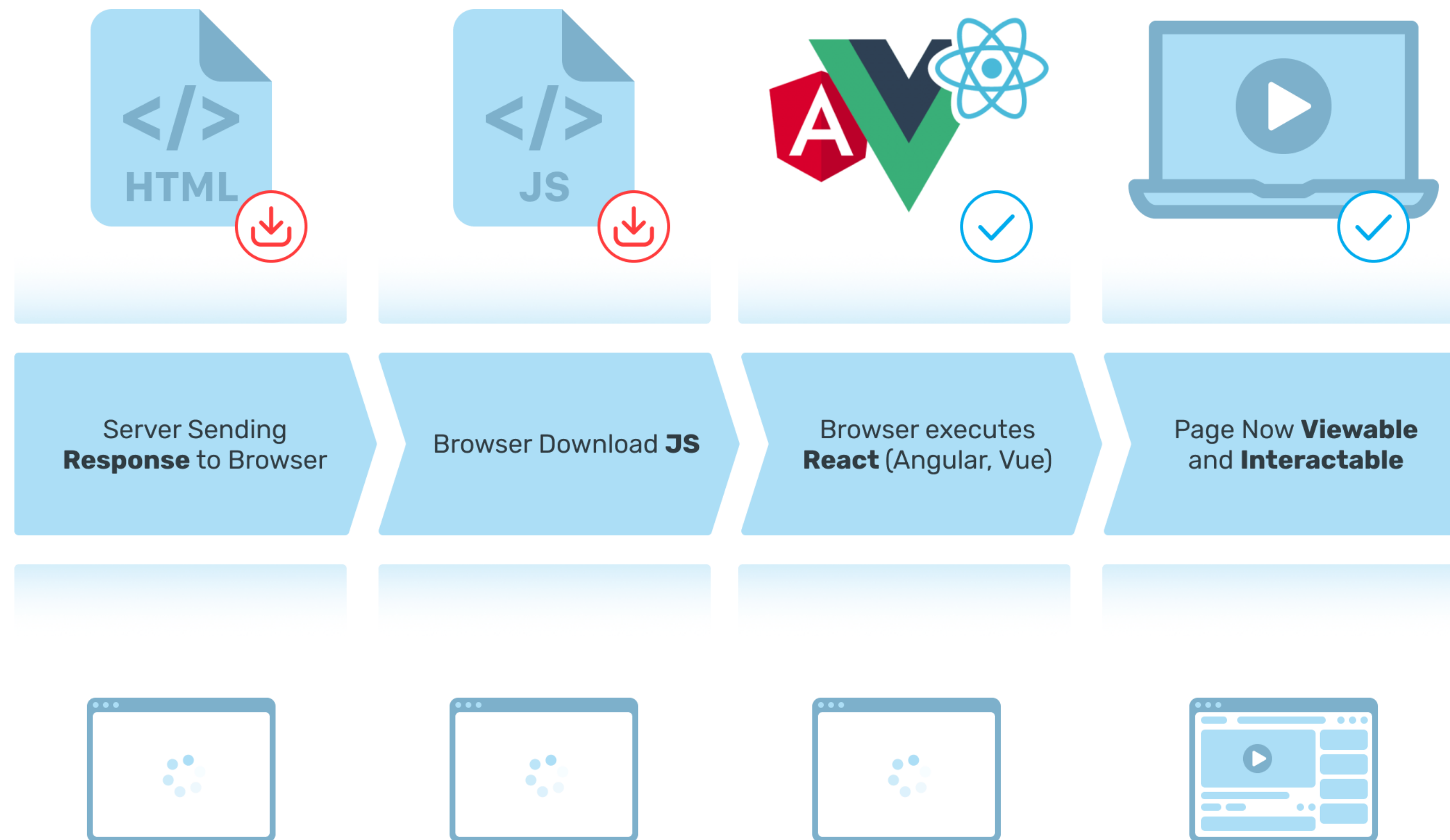
Индексирование AJAX-сайтов

Робот Яндекса может проиндексировать AJAX-сайт, если у каждой страницы сайта есть HTML-версия.

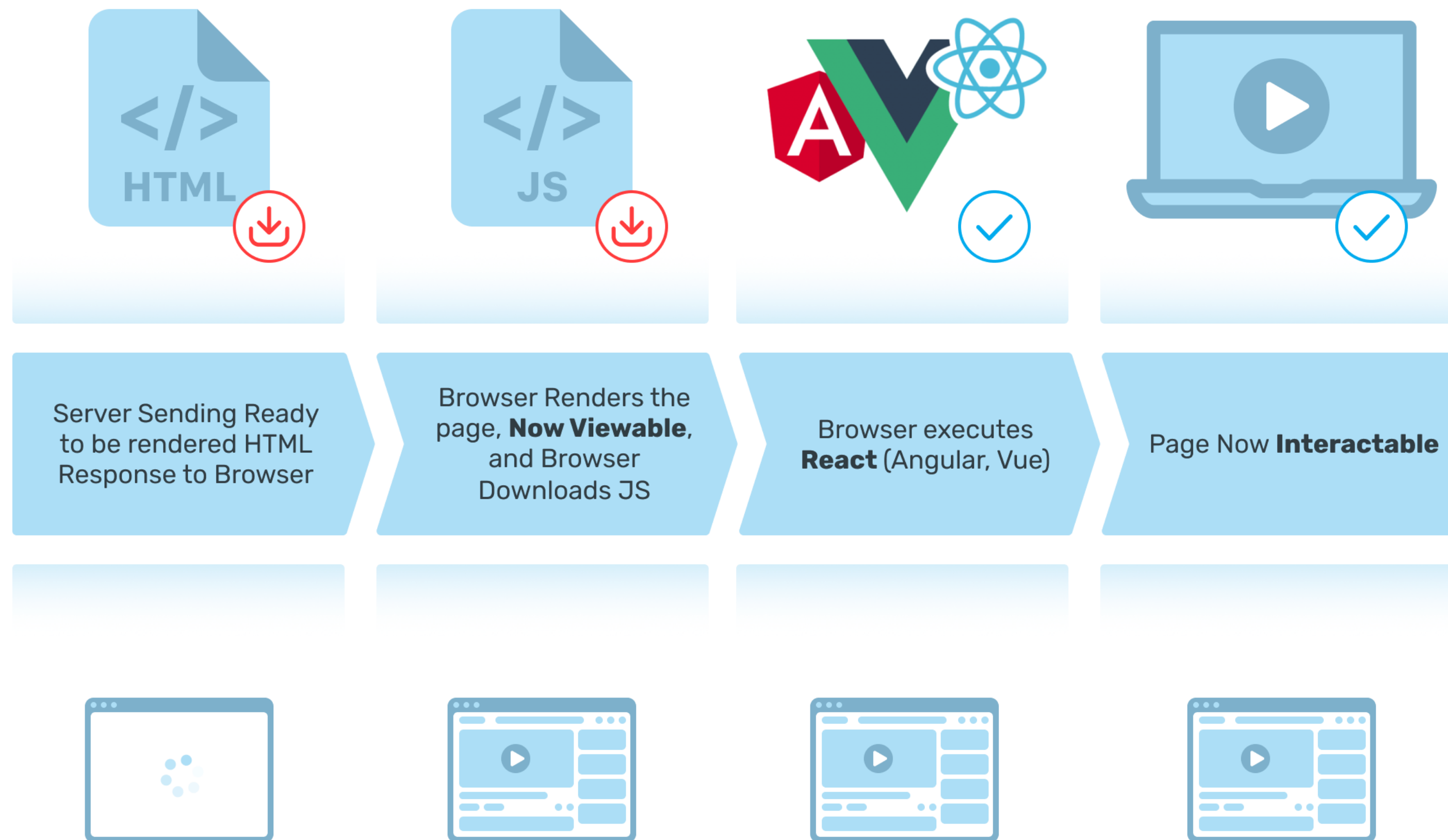


CSR -> SSR

CSR (Client Side Rendering)



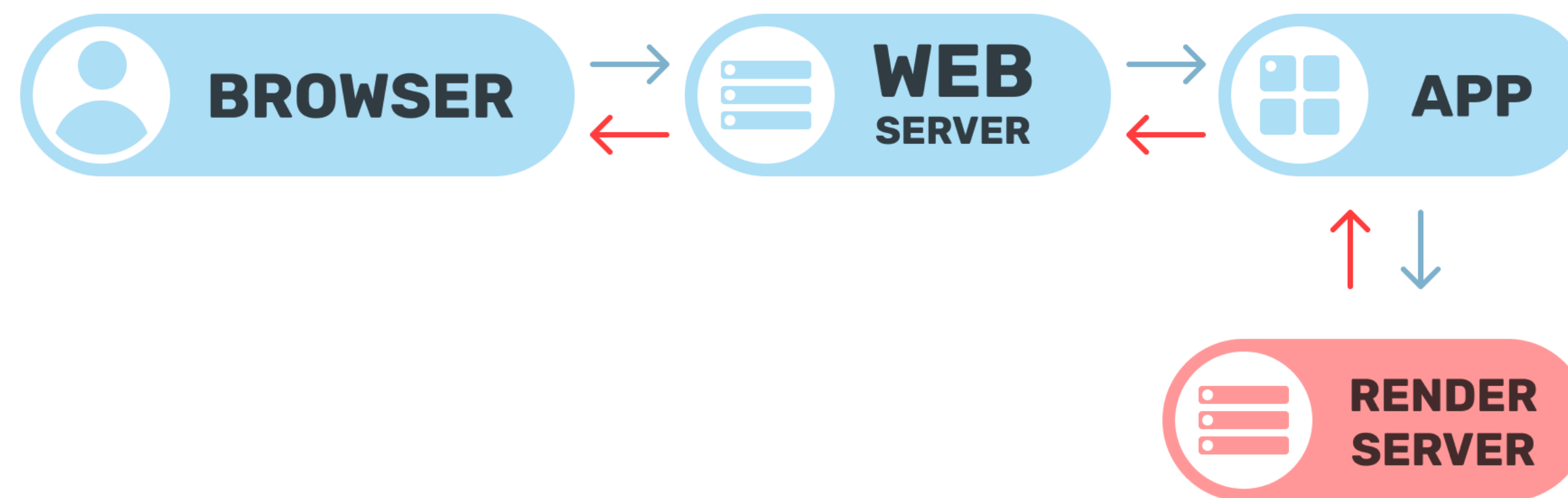
SSR (Server Side Rendering)



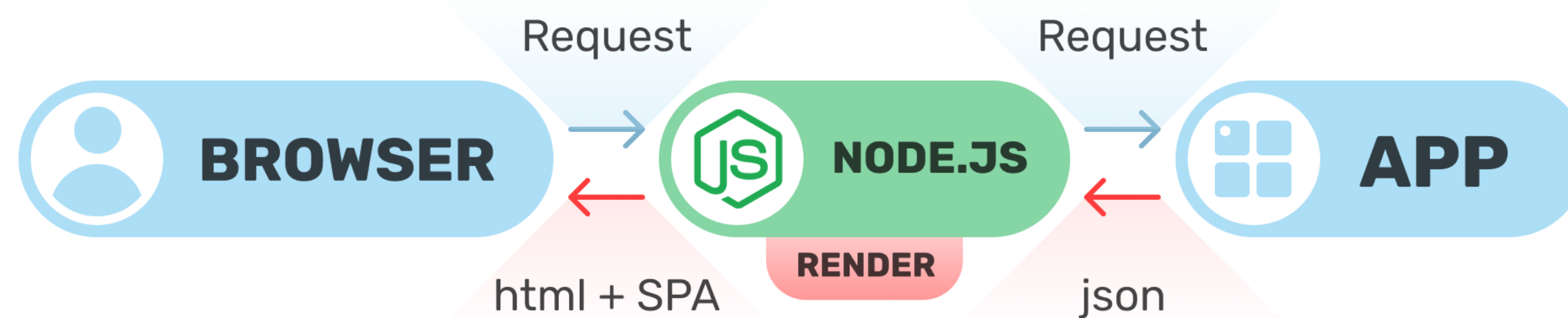
Рабочая схема при CSR



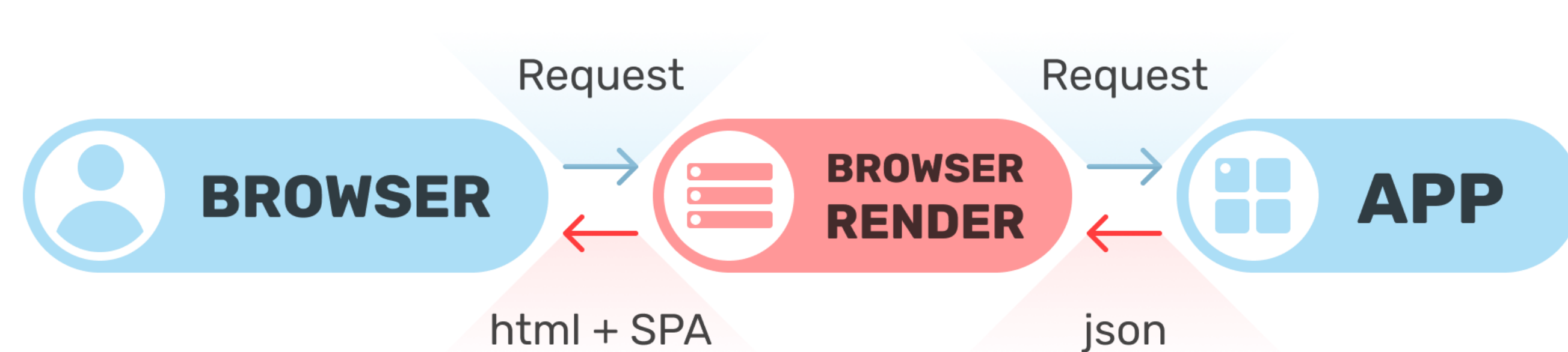
Рабочая схема при SSR #1



Рабочая схема при SSR #2



Рабочая схема при SSR #3

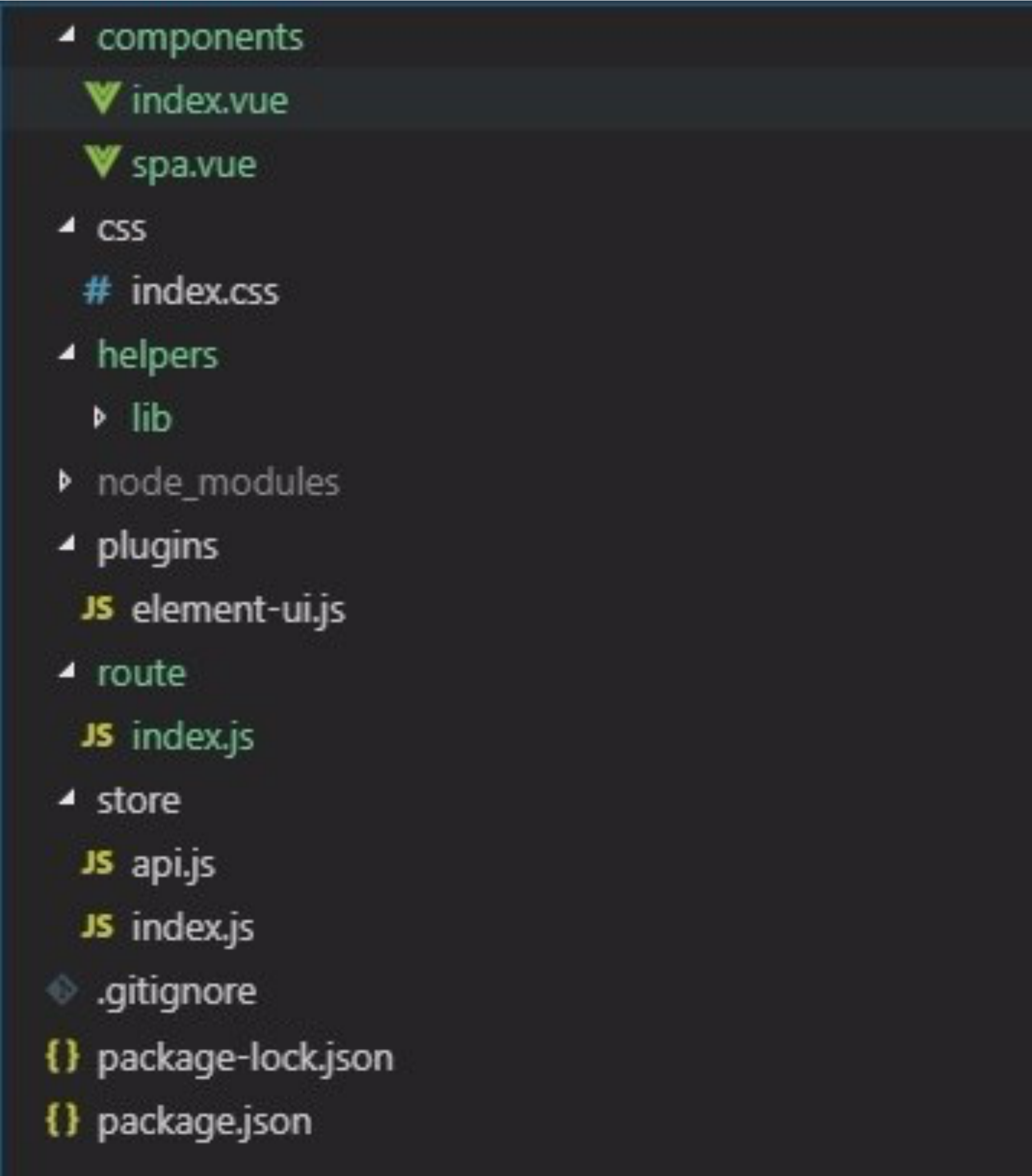


Решения

- angular - angular-cli
- react redux - next.js
- vuejs - nuxt.js
- prerender.io renderjs.io -
- <http://phantomjs.org/> и безголовые браузеры
- Самописное изоморфное приложение

Пример на VueJs и Nuxt

Как обычно выглядит SPA на VueJs



```
└─ components
  │   └─ index.vue
  │   └─ spa.vue
  │
  └─ css
      └─ # index.css
  │
  └─ helpers
      └─ lib
  │
  └─ node_modules
  │
  └─ plugins
      └─ JS element-ui.js
  │
  └─ route
      └─ JS index.js
  │
  └─ store
      └─ JS api.js
      └─ JS index.js
  │
  └─ .gitignore
  │
  └─ {} package-lock.json
  └─ {} package.json
```

A screenshot of a file explorer showing the directory structure of a Vue.js SPA project. The structure is as follows:

- components
 - index.vue
 - spa.vue
- css
 - # index.css
- helpers
 - lib
- node_modules
- plugins
 - JS element-ui.js
- route
 - JS index.js
- store
 - JS api.js
 - JS index.js
- .gitignore
- package-lock.json
- package.json

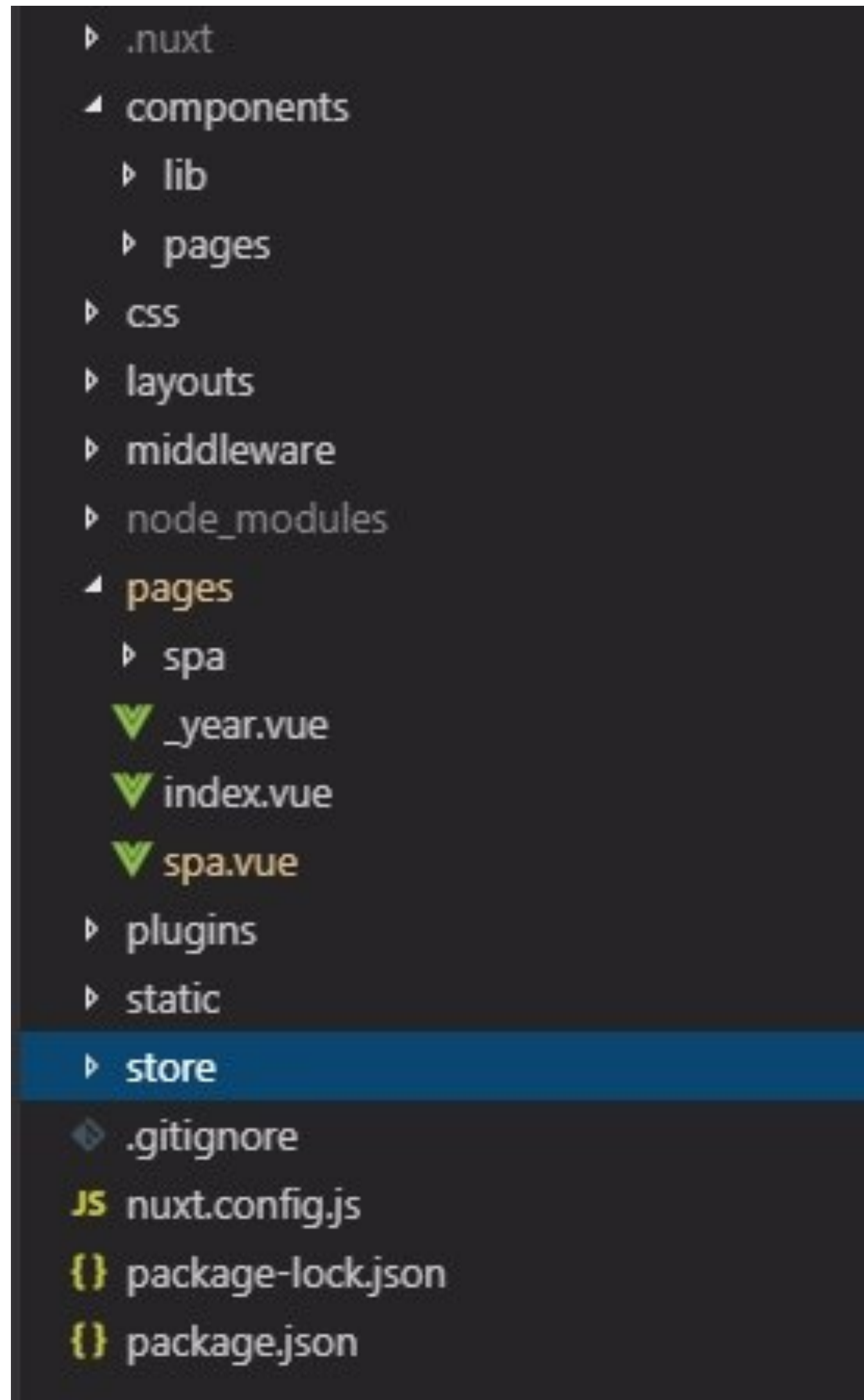
Nuxt.js

- Основные отличия от SPA приложения
 - static
 - Layouts
 - pages
 - middleware
 - nuxt.config.json

Ограничения Nuxt.js

- Нельзя platform-specific APIs браузера
- Нельзя глобальные переменные window, document.
 - beforeMount и mounted
- Нельзя код типа «side effect»
 - setInterval в beforeCreate и created

SPA на Nuxt.js



Примеры

- <https://vuenuxt.herokuapp.com/> - пример vue + nuxt
- <https://vuenuxt.herokuapp.com/spa> - пример классического spa
- <https://github.com/kiselevAlex/nuxt> - сам проект

Итоги

Плюсы SSR

- Пользователь видит страницу «быстрее»
- Для первоначальной отрисовки не требуется JS, **SEO**
- Увеличение скорости загрузки на мобильных устройствах, **SEO**
- Увеличение скорости загрузки на старых устройствах, **Охват, SEO**
- Возможность пользоваться sра без использования JS
(Изоморфность приложения)

Минусы SSR

- Нужен изоморфный код приложения (есть ограничения)
- Нужно рендерить на сервере (нагрузка с браузера уходит на сервер). Частично можно решить вопрос кэшем.
- Инфраструктура становится сложнее (одна из причин почему мало у кого есть SSR)
- Лучше всего рендерит nodejs на сервере либо безголовый браузер. Можно без nodejs, например v8js расширение на PHP. Но там уже есть ограничения.
- Разработка усложняется, в связи с изоморфным подходом.

Спасибо!

И не бойтесь серверного рендеринга