



Sitecore Headless Development with Next.js

Sebastian Tecsı

- Sitecore Architect
- First contact Sitecore 4.3
- Sitecore MVP 2018-2021
- Certified in Sitecore 5, 6, 8 and 9
- Working with Sitecore for 15+ years



Twitter: @sebastiantecsi
Blog: <https://xtremdev.com/>

Intro



.NET Core Rendering SDK
NuGet packages



JavaScript Rendering SDK (JSS)
npm packages

NEXT.js



Sitecore Headless Services

Used By .NET Core Rendering SDK

Used By JavaScript Rendering SDK (JSS)

Rendering
Engine

Media
Handler

GraphQL

Layout
Service

Dictionary
Service

Import
Service



SITECORE®

(XM/XP)

Layout Service



- The Sitecore Layout Service is a Sitecore Headless Services endpoint that exposes Sitecore layout information as structured Json

Dictionary Service

- Sitecore Headless Services provides an API endpoint for retrieving application-specific Dictionary data.



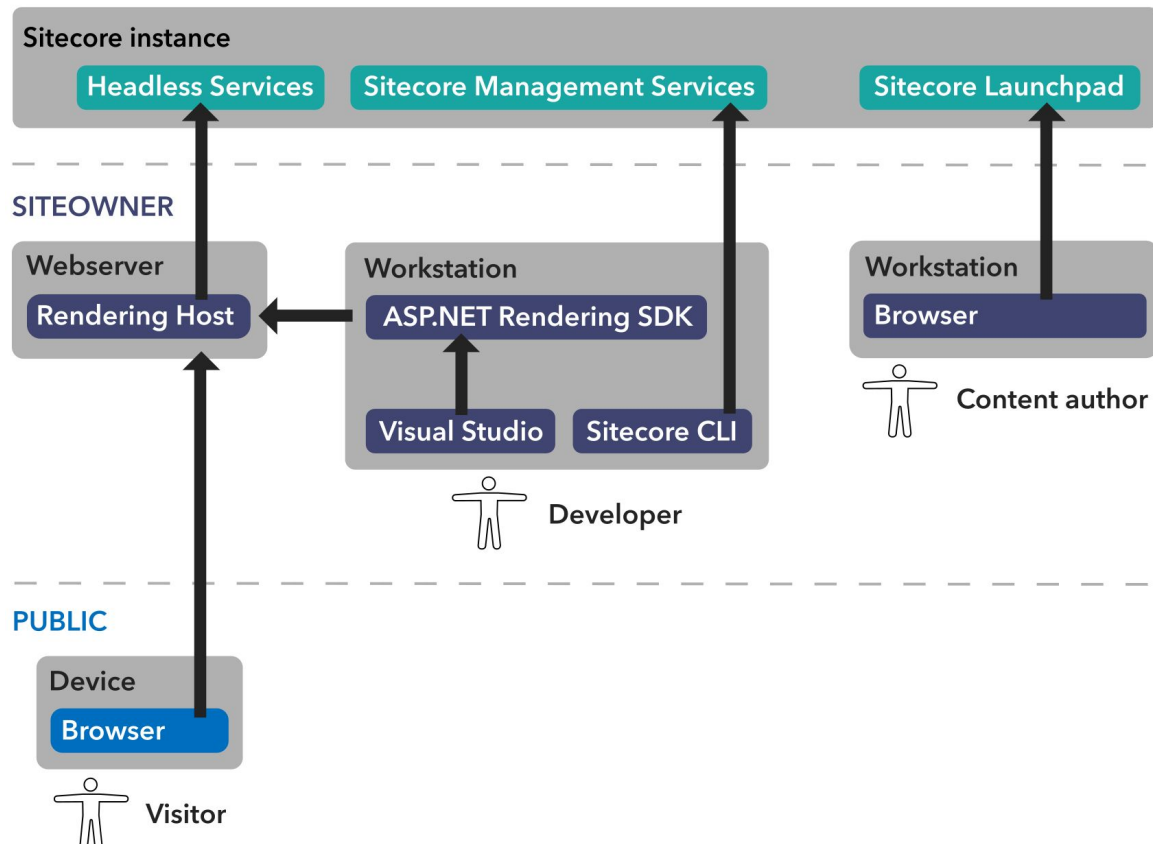
GraphQL



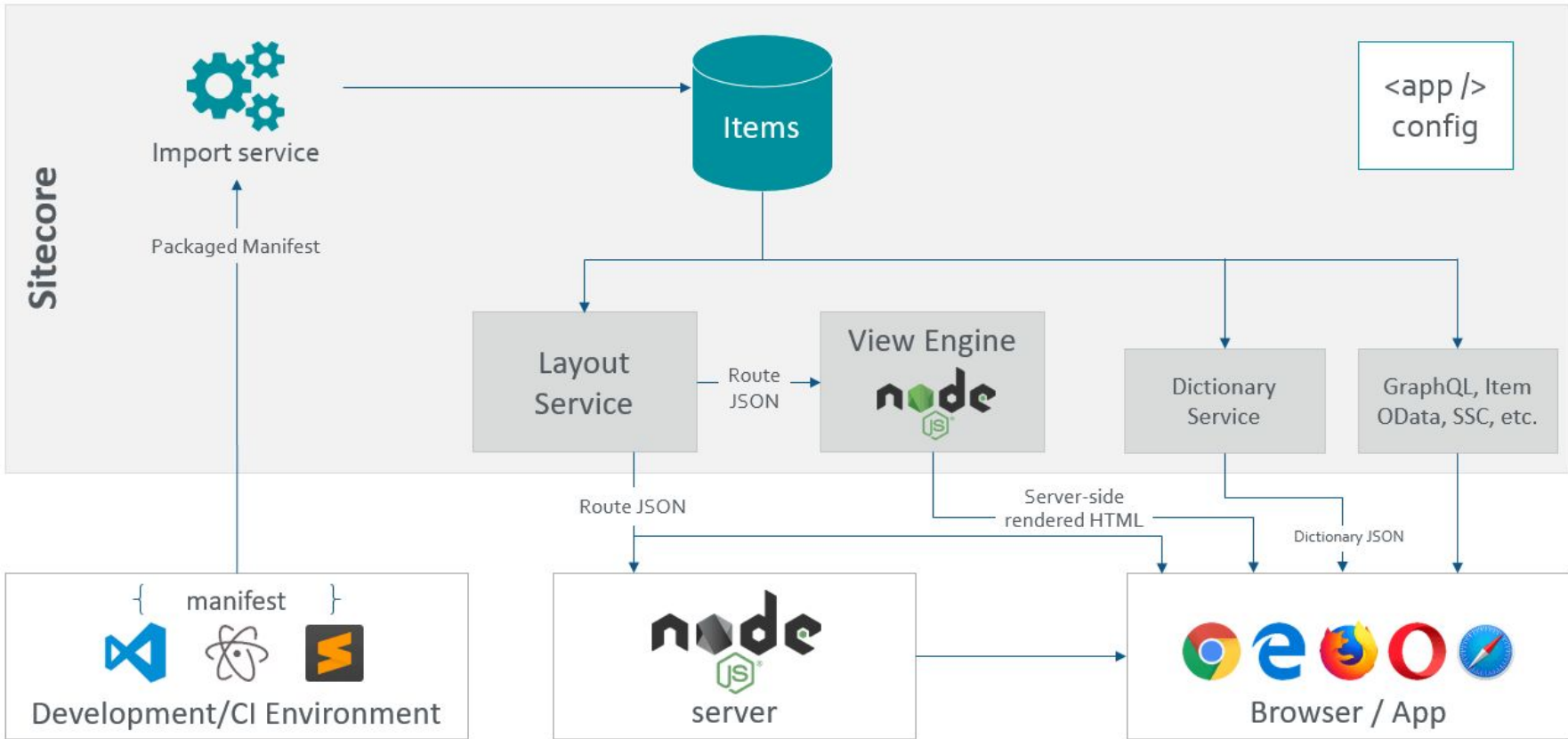
- Sitecore GraphQL API - a GraphQL service platform on top of Sitecore. You can use this to create custom endpoints.
- Sitecore Experience Edge for XM Preview GraphQL endpoint.

Sitecore ASP.NET Rendering SDK

SITECORE



Sitecore JavaScript Rendering SDKs (JSS)





Pre-requisites

- Sitecore License file that supports Headless Services
- Current LTS for NodeJS, other versions may work but you'll be on your own for any error messages
- dotnet CLI
- VS Code for editing
- VS



Features and capabilities of JSS

- JSS CLI - a Node-based command-line tool to help you maintain applications.
- A core JavaScript package for leveraging Sitecore Headless Services and Sitecore Experience Edge endpoints using REST or GraphQL.
- Dedicated SDKs for popular front-end frameworks:
 - Next.js (12.0+).
 - React (17.0+).
 - Angular (11.2+).
 - Vue.js (3.1+).
 - React Native (experimental).
- Support for Sitecore MVC applications, allowing developers to use JSS to generate MVC applications as static HTML.



Features and capabilities of Next.js

- Server-Side Rendering (SSR)
 - Fetch data on each request
- Static Generation (SSG)
 - `getStaticProps` Fetch data at build time
 - `getStaticPaths` Pre-render dynamic routes based on data

Demo Headless + Next.js + containers





Tips for working with Sitecore's JSS and Next/React.js apps:

- Clear delimitation of concerns between Sitecore components (defined in backend) and common ReactJS components
- Define early the generateComponentFactory script structure and make sure it only includes the Sitecore components
- Fallback for FE components if no data available
- Make use of React.Context in cases where you need to pass page level data to inner components
- Make sure to enforce linting (TS/JS) on commits/pushes to avoid breaking builds for releases
 - Check code for errors

Thank you!



Q&A