# 125.https://stackoverflow.com/questions/72763463/is-frontend-better-or-backend-for-integration-with-smart-contract-erc721

**T:**Is frontend better or backend for integration with smart contract (erc721)?

**Q:**I have developed my ethereum smart contract and I want to integrate it with my web application. But I don't know that is integration with ReactJS a better option or integration with nodejs a better one. Please give suggestions . Also I don't know how to do integration with backend through web3 library so please guide me about that.

3 **Answer**

**A1:**When you are creating web3 app , most of the time you are not going to need a backend for that (Although you need a server to host your website) .Working with and managing smart contracts are very easy at the front-end and there is no need for backend, but if your working on a large scale project , you are definitely going to need a backend for handling complex logics .  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Next js is best option for both of that , containing node and react , also no need for manually configuring web server.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

**C1:**the problem is I don't have much time to adopt a new technology for my app :

**C2:**use React with serverelss functions

**A2:**Better is next.js. I explained it here: https://ethereum.stackexchange.com/questions/129547/next-js-versus-react-which-to-use-when-for-your-dapp/130040#130040  
  
Not every browser has metamask extension. By using next.js, when ourcode is taken and rendered on the server, on the next server we canreach out to the Ethereum network and do some initial calls like datafetching, or alist of items in your smart contract. we execute all ofthose requests on the server. That means when next.js produces Htmldocuments to send down to the user browser, it does not matter whetheror not users are using metamask. It does not matter whether or notthey have access to an Ethereum network. Because we already take careof the data fetching for them. So all the users out there who are notusing metamask are going to see some information on the screen.

**C1:**Yes but my app is developed through react and node js and no time left to adopt new tech

**A3:**You've mentioned that you don't want to switch technologies, and while I agree with NextJS being a good platform to develop dApps on, I suggest you just use your current NodeJS server for anything that isn't web3 related and you use the web3js library from the frontend (your React) which would be very similar to a NextJS app anyway.  
  
This way you don't have to switch technologies.

**C1:**Ok so using frontend for integration, right? Also, whats the best approach. Like normally wallets are used to sign transactions but my application is web based that would be fine for me but what if i have to use it on mobile wont it be a problem to use wallet extension or should I include private keys in my code and dont use wallets, What is the best approach?

**C2:**You'd most likely use a wallet that can interact with the browser, like MetaMask. It can then also work through the metamask app, and many, if not nearly all other wallets support dApps in the same way, often by injecting a window.ethereum object that you can access in JS in the frontend. Also never store private keys in the frontend. You don't need private keys for a dApp anyway, only to deploy a smart contract.