

Integrating React View Layers with Persistent Data



Daniel Stern

CODE WHISPERER

<http://github.com/danielstern>



Integrating Front End With Persistent Data Roadmap



Add NPM script to initialize front end and server simultaneously

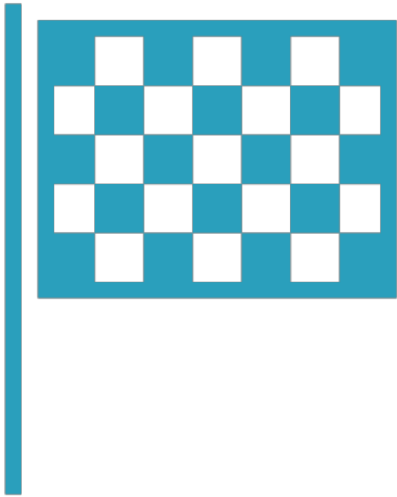
Update existing sagas to communicate with server created in previous module



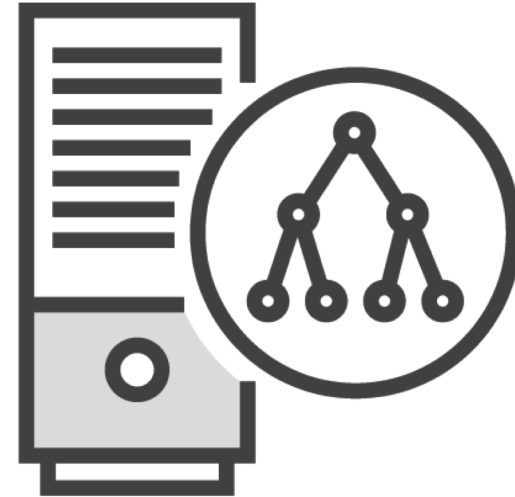
Initializing the Server and Client Simultaneously



Initializing the Server and Client Simultaneously



Script will initialize Webpack, which serves our application



Script will also initialize Express, which will wait for HTTP requests from our application and modify DB

Demo



Add start-dev NPM script

Verify that both front and back end are running when script is executed



Using Client-originated HTTP Requests to Modify Persistent Data



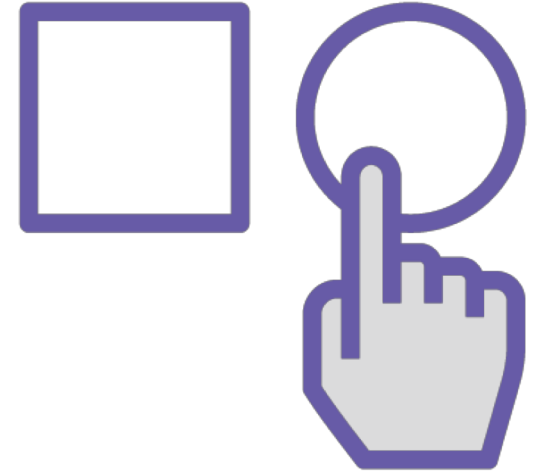
HTTP Requests



Used to communicate
with your server or
other authorized
domains



Cross-origin request
security (CORS)
prevents requests to
unauthorized domains



Many front-end tools
to make HTTP
requests: Axios,
jQuery, fetch, etc.



Demo



Install Axios using NPM

Create actual sagas to replace mock sagas

- Task creation saga will send HTTP request to *new task* route
- Task update saga will send HTTP request to *update task* route

Verify our data is now persisting



Summary



Changes to data made on client are usually relayed to server via HTTP and REST

Axios can be used to send request, Express can be used to receive request

Sagas or other side-effect management tools let our application work with or without server component

