2/7/2021 elementool - thai :: thai

Issue #1

Title: React-Script v3.4.1 fails to start in Docker Product: **Description:** After updating React-Script from v3.3.0 to v3.4.1, the React-Web Status: Fixed application fails to start in the Docker environment. Priority: Mediu Severity: Mediu Attempted to do: Version#: v3.4.1 1. Run the application with 'npm start': successfully start 2. Pruned docker containers and images and restart with 'build' and 'up': Frequency: 3. Downgraded to React-Script v3.3.0: successfully start with Docker and Assigned to: thai npm Reported by: thai CC: Console Logs: Recreating core-ui web 1 ... done **Time Spent:** Attaching to core-ui web 1 web 1| Submitted on: 0 web 1 | > core-ui@0.1.0 start /app web 1 | > react-app-rewired start 0 web 1| 7/ web 1 | i [wds]: Project is running at http://172.24.0.6/ 2 web_1 | i \(\text{wds} \): webpack output is served from 0 web 1 | i [wds]: Content not from webpack is served from /app/public 2 web_1 | i [wds]: 404s will fallback to / 1 web_1 | Starting the development server... web_1 | Fixed in version#: v3.4.2 core-ui web 1 exited with code 0 forwa Steps to Reproduce: Docker environment and dependencies: and all dc run web npx create-react-app --info previc npx: installed 99 in 7.075s 3.4.x**Environment Info:** Fixed by: current version of create-react-app: 3.4.1 running from /root/.npm/_npx/6/lib/node_modules/create-react-app System: OS: Linux 4.9 Debian GNU/Linux 9 (stretch) 9 (stretch) CPU: (4) x64 Intel(R) Core(TM) i7-4870HQ CPU @ 2.50GHz Binaries: Node: 12.13.1 - /usr/local/bin/node Yarn: 1.19.1 - /usr/local/bin/yarn npm: 6.12.1 - /usr/local/bin/npm Browsers: Chrome: Not Found Firefox: Not Found npmPackages: react: ^16.12.0 => 16.12.0 react-dom: ^16.12.0 => 16.12.0 react-scripts: ^3.4.1 => 3.4.1 npmGlobalPackages:

Reproducing steps:

create-react-app: Not Found

- 1. Install CRA with RS@3.4.1
- 2. Run in docker container
- 3. Exits with code 0



2/7/2021



thai 02/07/2021 6:57:54 PM

Reproducible demo

```Dockerfile

```
create base image with all packages, proper entrypoint, and directories created
FROM node:erbium AS base
install any packages we need from apt here
RUN apt-get update \
 && apt-get install dumb-init
set entrypoint to `dumb-init` as it handles being pid 1 and forwarding signals
so that you dont need to bake that logic into your node app
ENTRYPOINT ["dumb-init", "--"]
all of our code will live in `/app`
WORKDIR /app
using the base image, create an image containing all of our files
and dependencies installed, devDeps and test directory included
FROM base AS dependencies
COPY package*.json ./
RUN npm set progress=false \
 && npm config set depth 0 \
 && npm i
COPY ./config-overrides.js ./
COPY ./public ./public
COPY ./src ./src
if you have any build scripts to run, like for the `templated-site` flavor
uncomment and possibly modify the following RUN command:
RUN npm run build
keeping all of the bash commands you can within a single RUN is generally import
but for this case it's likely that we want to use the cache from the prune which
change infrequently.
test running image using all of the files and devDeps
FROM dependencies AS test
USER node
ENV NODE ENV=test
use `sh -c` so we can chain test commands using `&&`
CMD ["npm", "test"]
FROM dependencies AS development
expose port 3000 from in the container to the outside world
this is the default port defined in server/manifest.js, and
will need to be updated if you change the default port
EXPOSE 3000
CMD ["npm", "start"]
release ready image, devDeps are pruned and tests removed for size control
```

```
FROM development AS release
```

RUN npm prune --production \
 rm -rf lib/\*\*/\*.spec.js

...

**Expecting Behavior**: Server start **Actual Behavior**: Exit with status 0

## Linked Records:

