



## Baloot

### مقدمه

داکر امکانی را فراهم کرده است تا پروژه‌ها و نرم‌افزارها بصورت مجزا و در محیطی کاملاً ایزوله راه اندازی شوند. به این محیط ایزوله، container می‌گویند. container این امکان را برای توسعه دهندگان نرم‌افزارها فراهم می‌کند تا یک برنامه را با تمام ماژول‌ها و اجزای وابسته به آن (مانند کتابخانه‌ها، توابع و ...) در کنار هم قرار داده و در قالب یک پکیج آماده‌ی اجرا ارائه می‌کند. این پکیج می‌تواند در پلتفرم‌ها و سیستم‌های مختلف بدون نیاز به نصب مجدد نیازمندی‌ها، اجرا شود. بنابراین این برنامه در هر محیطی بدون نگرانی از تنظیمات و وابستگی‌ها به سایر برنامه‌ها اجرا خواهد شد.

هر کانتینر در واقع یک نمونه‌ی قابل اجرا از یک image است. image یک فایل غیر قابل تغییر است که اطلاعات و دستورات مورد نیاز برای اجرای container را نگهداری می‌کند.

برای ساخت یک image از Dockerfile استفاده می‌کنیم. Dockerfile یک اسکریپت است که دستورات آن مشخص می‌کنند image خود را چگونه بسازیم. در واقع نیازمندی‌هایی که برنامه‌ی ما برای اجرا به آن نیاز دارد را در Dockerfile مشخص کرده و سپس image را بر اساس آن می‌سازیم.

## فاز هشتم پروژه

---

### داکری کردن پروژهی Front-End

در محیط‌های عملیاتی برنامه‌های تحت REST، یک سرور وظیفه‌ی پاسخگویی به درخواست‌ها را دارد و یک سرور فایل‌های استاتیک (که پروژه React شما نیز جزو این دسته قرار می‌گیرد) را ارائه می‌کند. شما باید یک Dockerfile بنویسید که ابتدا از پروژهی React یک ساخت<sup>1</sup> برای محیط عملیاتی بگیرد و در یک سرور (مانند Nginx) فایل‌ها را ارائه دهد. این کار مسئله‌ای مرسوم است و اگر در انجام آن به مشکل خوردید، سعی کنید در اینترنت به جستجوی راه حل مشکل بپردازید.

---

### داکری کردن پروژهی Back-End

شما در این بخش باید با نوشتن یک Dockerfile، یک image داکر برای سیستم خود ارائه دهید و در انتها با ایجاد یک container از image ساخته شده، سرور Back-end شما آماده‌ی پاسخگویی به درخواست‌ها باشد. دستوراتی که در Dockerfile می‌نویسید تقریباً همان دستوراتی است که روی سیستم در زمان ساخت و اجرای پروژه استفاده می‌کنید.

---

### نکات تکمیلی و راهنمایی

- برای آشنایی با داکر و نوشتن Dockerfile از [این لینک](#) و برای Serve کردن پروژهی React توسط یک وب سرور (Nginx) می‌توانید [این لینک](#) را مطالعه کنید.
- اگر در نوشتن Dockerfile به مشکلی برخوردید حتماً به مستندات داکر مراجعه کنید و سعی کنید تا حد ممکن از راه حل‌هایی که برای مشکلاتتان در اینترنت وجود دارد استفاده کنید.

---

<sup>1</sup> Build

---

## نکات پایانی

- کافی است که یکی از اعضای گروه Hash مربوط به آخرین کامیت پروژه سمت سرور و سمت کاربر را در سایت درس آپلود کند. در هنگام تحویل، پروژه روی این کامیت مورد ارزیابی قرار می‌گیرد.
- در صورت مشکل با تحریم و عدم استفاده از vpn، می‌توانید از [docker.ir](https://docker.ir) استفاده کنید.
- ساختار صحیح و تمیزی کد برنامه، بخشی از نمره‌ی این فاز پروژه‌ی شما خواهد بود. بنابراین در طراحی ساختار برنامه دقت به خرج دهید.
- هدف این تمرین یادگیری شماسست. لطفاً تمرین را خودتان انجام دهید. در صورت مشاهده‌ی مشابهت بین کدهای دو گروه، از نمره هر دو گروه مطابق سیاستی که در کلاس گفته شده است کسر خواهد شد.
- سوالات خود را تا حد ممکن در فروم درس مطرح کنید تا سایر دانشجویان نیز از پاسخ آن‌ها بهره‌مند شوند. در صورتی که قصد مطرح کردن سوال خاصی داشتید، از طریق ایمیل با طراحان این فاز پروژه ارتباط برقرار کنید.
- ایمیل طراحان پروژه:

[rezaqavi1379@gmail.com](mailto:rezaqavi1379@gmail.com)

[aalizad79@gmail.com](mailto:aalizad79@gmail.com)

موفق باشید.