



لقمه!

هدف اولیه از این پروژه، آشنایی با CORS^۱، استانداردسازی API ها و تبدیل خروجی آن‌ها به فرمت JSON و آشنایی با Postman است. دقت کنید از این فاز به بعد، دیگر نیازی به صفحات JSP و Server Side Rendering ندارید.

هدف نهایی این پروژه این است به کمک چارچوب^۲ React کدهای سمت کاربر خود را با معماری RIA به سرور متصل کنید. جهت آشنایی اولیه با React در ابتدا پیشنهاد می‌شود که به مطالبی که در کلاس آموزش داده شده‌اند رجوع کنید و سپس [این منبع](#) نیز می‌تواند کمک شایانی کند.

بخش اول (مقدمات RIA)

آشنایی با CORS

هنگامی که از یک آدرس بخواهیم به آدرس دیگری (تفاوت می‌تواند در آدرس یا پورت باشد) درخواست HTTP ارسال کنیم، نیاز است که سایت مقصد به ما اجازه دسترسی به منابعش را داده باشد. این امر با مقداردهی به چند پارامتر در Header جواب ارسالی سایت مقصد مشخص می‌شود و به این مکانیزم CORS گفته می‌شود. در این بخش، لازم است تا ابتدا با [CORS](#) آشنا شوید و سپس هدرهای مورد نیاز را در کدهای خود قرار دهید. برای قراردادن هدرهای مربوطه در هر درخواست، از [Filter](#) استفاده کنید. Filter یک شیء است که قبل از API های مورد نظر و همچنین بعد از پاسخ آن‌ها قرار می‌گیرد و امکانات بسیاری از قبیل امکان ایجاد تغییرات در درخواست‌ها، پاسخ‌ها و... را در اختیار قرار می‌دهد.

استانداردسازی API ها

در این بخش لازم است تا API های خود را استاندارد کنید. توصیه می‌کنیم از [این کتاب](#) و مطالب گفته شده در کلاس استفاده کنید. برای این کار لازم است تا علاوه بر استانداردسازی URL های API های خود، متد HTTP درخواست‌های خود را نیز استاندارد کنید. (مثلاً برای حذف یک غذا از سبد خرید از متد DELETE استفاده کنید).

تبدیل خروجی API ها به فرمت JSON

در این قسمت لازم است تا فرمت خروجی های API های خود را یکسان کنید. برای این کار، لازم است تا خروجی آن‌ها را به فرمت JSON تبدیل کنید. API هایی که از نوع GET هستند، باید خروجی JSON برگردانند و API های متدهای

^۱ Cross-Origin Resource Sharing

^۲ Framework

دیگر باید پاسخی به فرمت JSON با ³Status Code مناسب برگردانند. برای اینکار کافیت از سرویس‌های اسپرینگ در بخش backend برنامه خود استفاده کنید؛ چرا که این سرویس‌ها خروجی را به JSON تبدیل می‌کنند.

آشنایی و استفاده از Postman

[Postman](#) ابزاری برای ارسال درخواست با HTTP Method و فرمت دلخواه به یک سرور است. برای آزمایش و مشاهده خروجی Method هایتان می‌توانید از این ابزار استفاده کنید. لازم است تا این ابزار را در هنگام تحویل روی لپ‌تاپ خود نصب داشته باشید. در هنگام تحویل، یکی از API هایتان به صورت تصادفی با Postman آزموده خواهد شد. توجه کنید که در بخش بعدی (React) تمامی سرویس‌های گفته شده باید در frontend انجام پذیر باشند و این ابزار تنها برای آزمایش خروجی سرویس هایتان می‌باشد.

بخش دوم (React)

- در این فاز نیاز است به طراحی دو صفحه باقی‌مانده، یعنی خانه و ورود پردازید (صفحه ورود با خلاقیت خودتان). سپس تمام صفحات را با توجه به دانشی که از چارچوب React به دست می‌آورید، مجدداً با استفاده از این چارچوب پیاده‌سازی کنید.

- صفحه ورود باید شامل ایمیل و رمز عبور باشد.

- دقت کنید نیازی نیست که بخش‌های زیر در این فاز به سرور متصل شوند و پیاده‌سازی منطق سمت سرور و اتصال به آن‌ها، مربوط به فازهای بعدی است:

- ورود
- ثبت نام
- جست‌وجو در رستوران‌ها و غذاها
- خروج از حساب کاربری

- توجه داشته باشید به جز موارد بالا، سایر امکانات نرم‌افزار برای «کاربر وارد شده» باید قابل انجام باشد. به عنوان مثال تمامی قابلیت‌های زیر باید از طریق ارسال درخواست از سمت کاربر (Client) و دریافت پاسخ از سمت سرور انجام‌پذیر باشند:

- مشاهده جشن غذا و رستوران‌ها
- سفارش از جشن غذا
- مشاهده منوی رستوران‌ها و اضافه کردن غذا به سبد خرید از طریق آن
- مشاهده پروفایل کاربر
- مشاهده سفارش‌های پیشین و افزایش اعتبار
- فاکتور یک سفارش
- سبد خرید

نکات تکمیلی

- پیشنهاد می‌کنیم قبل از شروع پروژه، پروژه دست‌گرمی سایت React را که در [این](#) لینک قابل مشاهده است پیاده‌سازی کنید تا با مفاهیم کلی این چارچوب به خوبی آشنا شوید.

³ https://en.wikipedia.org/wiki/List_of_HTTP_status_codes , <https://http.cat/>

- مدیریت حالت⁴ در این پروژه و چارچوب React اهمیت شایانی دارد. بنابراین سعی کنید به این موضوع اهمیت زیادی دهید و از مفاهیم آن به خوبی استفاده کنید.
- حالت «در حال بارگذاری» را باید در تمامی صفحات و بخش‌هایی که نیاز است پیاده‌سازی کنید؛ یعنی برنامه شما باید هنگامی که درخواست را از سمت کاربر به سمت سرور ارسال کرده است و منتظر پاسخ است، با نشان دادن پیام یا نشانگر مناسب، این حالت را به درستی نشان دهد. می‌توانید از [Spinner](#)ها استفاده کنید. توجه داشته باشید که با توجه به این‌که سرور شما نیز روی همان سیستم در حال اجراست (localhost)، پیشنهاد می‌شود از یک فیلتر برای ایجاد تأخیر در ارسال پاسخ‌ها در سمت سرور استفاده کنید. این لینک می‌تواند در این امر به شما کمک کند.
- توجه کنید که شما در حال استفاده از یک چارچوب هستید. تمیزی کد و استفاده چندباره از کامپوننت‌ها اهمیت دارد و هیچ توجیهی برای داشتن کدهای تکراری⁵ وجود ندارد.
- برای ساخت پروژه‌ی React و شروع آن می‌توانید از [این](#) لینک استفاده کنید.
- پوشه‌بندی و نحوه مدیریت فایل‌های شما اهمیت دارد. برای پوشه‌بندی مناسب می‌توانید از [این](#) لینک و بخش "Grouping by features or routes" استفاده کنید. اگر تمایل به آشنایی بیشتر در مورد نحوه پوشه‌بندی مناسب دارید نیز [این](#) لینک مناسب است.
- ورودی‌ها و فیلدهایی که برای سرور ارسال می‌کنید مانند فیلد جست‌وجو یا ورودی‌های صفحه ثبت نام باید قبل از ارسال در سمت کاربر Validate شوند.
- در صورت عدم موفقیت در انجام یک درخواست، نمایش مناسب پیام خطا حائز اهمیت است؛ برای مثال می‌توانید از [این](#) لینک و یا [این](#) لینک استفاده کنید.
- برای برقراری ارتباط با سرور و ارسال یا دریافت اطلاعات استفاده از ابزاری مانند [axios](#) پیشنهاد می‌شود. این کتابخانه با قراردادن یک Interface برای ارسال و دریافت درخواست‌ها کار شما را بسیار راحت می‌کند.
- استفاده از TypeScript برای انجام پروژه مانعی ندارد. (دقت کنید که برای این قسمت نمره امتیازی در نظر گرفته نمی‌شود)
- **امتیازی:** مقادیر Props کامپوننت‌های شما باید توسط PropTypesها نوع‌سنجی⁶ شوند. همچنین در صورت نیاز مقادیر defaultProps را باید تعیین کنید. [این](#) لینک راهنمای مناسبی برای آشنایی است. توجه کنید در صورتی‌که از TypeScript استفاده می‌کنید، به دلیل این‌که Propهای شما نوع مشخصی دارند نیازی به پیاده‌سازی این بخش نیست، اما باید Propهای اختیاری و اصلی را مشخص کنید.

نکات پایانی

کافی است که یکی از اعضای گروه در دو خط آدرس مخزن‌های پروژه (back-end, front-end) را در گیت‌لب و در خط‌های بعدی Hash مربوط به آخرین کامیت‌های این مخزن‌ها را در سایت درس آپلود کند. در هنگام تحویل، پروژه روی این کامیت‌ها مورد ارزیابی قرار می‌گیرد.

ساختار صحیح و تمیزی کد برنامه بخشی از نمره این فاز پروژه شما خواهد بود. بنابراین در طراحی ساختار برنامه دقت به خرج دهید.

هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت مشاهده شباهت بین کدهای دو گروه، ۵۰٪ نمره کل پروژه‌ها از گروه متقلب و تقلب‌دهنده کسر خواهد شد و در صورت تکرار، این روند ادامه خواهد داشت.

⁴ State Management

⁵ duplicate code

⁶ type-check

سؤالات 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 خاص‌تری داشتید، از طریق ایمیل با طراحان این فاز پروژه ارتباط برقرار کنید. توجه داشته باشید که دیگر شبکه‌های اجتماعی مانند تلگرام راه ارتباطی رسمی با دستیاران آموزشی نیست و دستیاران آموزشی موظف به پاسخگویی در محیط‌های غیررسمی نیستند.
ایمیل طراحان پروژه:

farzadhabibii98@gmail.com , hossein.soltanloo@gmail.com