



مدرس: دکتر خامس پناه

طراحان: کیمیا خبیری، طاهای شعبانی

مهلت تحویل: شنبه ۴ اردیبهشت ۱۴۰۰، ساعت ۲۳:۵۵



بلبلستان!

مقدمه

اهداف پروژه پنجم درس مهندسی اینترنت، در دو بخش کلی قابل بیان است: هدف اولیه آشنایی با مکانیزم CORS¹، استانداردسازی API و تبدیل خروجی آن به فرمت JSON و همینطور آشنایی با ابزار postman است. دقت کنید که از این فاز به بعد، دیگر نیازی به صفحات JSP و Server Side Rendering ندارید.

هدف بعدی استفاده از چارچوب React² برای پیاده‌سازی سمت کاربر پروژه با استفاده از معماری RIA³ و اتصال آن به سرور است. جهت آشنایی اولیه با React پیشنهاد می‌شود ابتدا به مطالبی که در کلاس آموزش داده شده است مراجعه کنید و سپس از [این منبع](#) نیز در صورت نیاز کمک بگیرید.

¹ Cross-Origin Resource Sharing

² Framework

³ Rich Internet Applications

بخش اول (مقدمات RIA)

آشنایی با CORS

هنگامیکه از یک آدرس بخواهیم به آدرس یا پورت دیگری درخواست HTTP ارسال کنیم، نیاز است که سایت مقصد به ما اجازه دسترسی به منابع را داده باشد. این امر با مقداردهی چند پارامتر در Header پاسخ ارسال شده از سوی سایت مقصد مشخص میشود و به این مکانیزم CORS گفته می شود.

در این بخش لازم است تا ابتدا با [CORS](#) آشنا شوید و سپس Header های مورد نیاز را در کدهای خود قرار دهید. برای قرار دادن هدرهای مربوطه در هر درخواست از [Filter](#) استفاده کنید. Filter یک شیء است که قبل از درخواست به API های مورد نظر و همچنین بعد از پاسخ ارسالی آنها قرار می گیرد و امکانات بسیاری از قبیل امکان ایجاد تغییرات در درخواست ها، پاسخ ها و ... را در اختیار قرار می دهد.

استانداردسازی API ها

در این بخش لازم است تا API های خود را استاندارد کنید. توصیه می کنیم از [این کتاب](#) و منابع گفته شده در کلاس استفاده کنید. برای این کار لازم است تا علاوه بر استانداردسازی URL های API خود، متد HTTP درخواست های خود را نیز استاندارد کنید. (مثلا برای حذف یک درس از لیست دروس از متد DELETE استفاده کنید.)

تبدیل خروجی API ها به فرمت JSON

در این قسمت لازم است تا فرمت خروجی API های خود را یکسان کنید. برای اینکار لازم است تا خروجی آنها را به فرمت JSON تبدیل کنید. API ها باید خروجی خود را به فرمت JSON و به همراه Status code مناسب برگردانند. برای اینکار کافیهست از سرویس های اسپرینگ در بخش Backend برنامه ی خود استفاده کنید چرا که این سرویس ها خروجی را به JSON تبدیل می کنند.

آشنایی و استفاده از Postman

[Postman](#) ابزاری برای ارسال درخواست‌هایت با متدهای HTTP و فرمت دلخواه به یک سرور است. برای آزمایش و مشاهده‌ی خروجی متدهایتان می‌توانید از این ابزار استفاده کنید. لازم است تا این ابزار را در هنگام تحویل روی لپ تاپ خود نصب داشته باشید. در هنگام تحویل یکی از API های شما به صورت تصادفی با Postman آزموده خواهد شد. توجه کنید در بخش بعدی (React) تمامی سرویس‌های گفته‌شده باید در Frontend انجام‌پذیر باشد و این ابزار تنها برای آزمایش خروجی سرویس‌هایتان می‌باشد.

بخش دوم (React)

در این بخش ابتدا نیاز است که تمامی صفحات را با استفاده از دانشی که در زمینه چارچوب React کسب کرده اید، پیاده‌سازی کنید (صفحه ورود/ثبت‌نام را به سلیقه خودتان طراحی کنید). دقت داشته باشید که صفحه ورود باید شامل ایمیل و رمز عبور باشد. صفحه ثبت‌نام نیز حداقل شامل نام و نام خانوادگی، ایمیل و رمز عبور است. (فیلدهای تاریخ تولد، دانشکده، رشته و مقطع را نیز در صورت تمایل می‌توانید به صورت اختیاری دریافت کنید) برای این فاز و مشابه پیاده‌سازی فازهای قبل، تنها با شماره دانشجویی لاگین ابتدایی داشته باشید. (طراحی صفحه ورود را با توجه به ورودی‌های ایمیل و رمز عبور پیاده‌سازی کنید ولی در این فاز برای لاگین فقط یک ورودی و با مقدار شماره دانشجویی از کاربر دریافت کنید. برای مثال می‌توانید در هنگام لاگین مقدار رمز عبور را خالی بگذارید و در یکی از inputها که در آینده ایمیل ورود خواهد بود، مقدار شماره دانشجویی را وارد کنید).

- دقت داشته باشید که ثبت‌نام در این فاز نیازی نیست که به سرور متصل شود و پیاده‌سازی منطقی و اتصال آن مربوط به فازهای بعدی می‌شود.

به جز موردی که در بالا ذکر شده‌است، سایر امکانات نرم‌افزار می‌بایست قابل اجرا بوده و تعامل سمت کاربر و سمت سرور در آن به درستی برقرار باشد. برای مثال:

- احراز هویت اولیه و خروج از حساب کاربری مطابق با فازهای قبلی
- مشاهده اطلاعات پروفایل (صفحه خانه)
- مشاهده اطلاعات نمرات کارنامه (صفحه خانه)
- مشاهده اطلاعات برنامه هفتگی (صفحه برنامه هفتگی)

- افزودن (چه در ظرفیت اصلی و چه در لیست انتظار) و حذف کردن درس از برنامه هفتگی پیش از ثبت نهایی (صفحه انتخاب واحد)
- افزودن (چه در ظرفیت اصلی و چه در لیست انتظار) و حذف کردن درس پس از ثبت نهایی (صفحه انتخاب واحد)
- مشاهده دروس برداشته شده و در انتظار (صفحه انتخاب واحد)
- مشاهده دروس موجود و اطلاعات آن‌ها (صفحه انتخاب واحد)
- جست‌وجوی دروس مورد نظر با اسم درس (صفحه انتخاب واحد)
- اعمال تغییرات زمان‌بند دروس لیست انتظار (صفحه انتخاب واحد)
- نهایی کردن برنامه درسی ترم (صفحه انتخاب واحد)

نکات تکمیلی و راهنمایی

- قبل از شروع پروژه پیشنهاد می‌شود که [پروژه دست‌گرمی](#) سایت React را مشاهده و بررسی کنید تا با مفاهیم کلی این چارچوب به خوبی آشنا شوید.
- برای ساخت پروژه React و شروع آن می‌توانید از [این لینک](#) استفاده کنید. دقت کنید که برای این پروژه نیازی به استفاده از SSG⁴ ها نظیر Gatsby نیست.
- توجه داشته باشید که تمیزی کد و استفاده چندباره از کامپوننت‌ها اهمیت زیادی دارد. بنابراین با استفاده از قابلیت‌هایی که چارچوب React در اختیاران قرار می‌دهد، سعی در داشتن حداقل کد تکراری را داشته باشید.
- مدیریت حالت⁵ در این پروژه بسیار مهم است. بنابراین سعی کنید به این موضوع اهمیت زیادی دهید و در ساختار پروژه از آن بهره ببرید.
- با مفاهیم [life cycle](#) و انواع آن و همچنین [hook](#) ها آشنا باشید، زیرا در حین پیاده‌سازی به کارتان خواهد آمد.
- به پوشه‌بندی و نحوه مدیریت فایل‌ها دقت داشته باشید. برای مثال می‌توانید از [این لینک](#) و بخش Grouping by features or routes استفاده کنید. در صورت تمایل نیز می‌توانید درباره نحوه پوشه‌بندی مناسب جستجو کرده و با روش‌های متنوع آن آشنا شوید.

⁴ Static website generator

⁵ State management

- حالت (در حال بارگذاری) را برای 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 نمایش دهد. برای مثال می‌توانید از [spinner](#) استفاده کنید. توجه داشته باشید که با توجه به این که سرور و برنامه کلاینت شما هر دو بر روی localhost در حال اجرا هستند، پیشنهاد می‌شود برای تست حالت بارگذاری، با ایجاد تاخیر در سمت سرور و یا در سمت کلاینت، این امر را بررسی کنید.
- ورودی‌هایی که به سمت سرور ارسال می‌شوند، مانند: در فاز فعلی شماره دانشجویی برای ورود (در فازهای آینده: ایمیل) باید قبل از ارسال validate شوند.
- در صورت عدم موفقیت در انجام یک درخواست، نمایش پیام مناسب 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 می‌توانید از [fetch api](#) و یا ابزار [axios](#) استفاده کنید. پیشنهاد می‌شود تفاوت‌های این دو روش را بررسی کنید و بسته به نیاز خود از هر کدام که مایل بودید استفاده کنید.
- استفاده از typescript و scss برای انجام پروژه مانعی ندارد.

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.
- 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 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:

در صورت داشتن سوال از بخش مقدمات RIA به kimia.khabiri.92@gmail.com و در صورت داشتن سوال از بخش React به taha.shabani.m@gmail.com ایمیل بزنید.