مدرس: دکتر احسان خامس‌پناه پوریا تاج محرابی، میثاق محقق

MioBook

مقدمه

هدف از این پروژه، آشنایی با روش‌های احراز هویت (Authentication)، کسب اجازه (Authorization) و اطمینان از وجود برخی از پارامترهای امنیتی در برنامه شما می‌باشد.

ثبت نام

فرآیند ثبت نام کاربران در سیستم همانند قبل باقی می‌ماند. کاربران در صفحه ثبت نام سایت، با وارد کردن فیلدهای مورد نیاز (نام کاربری، رمز عبور، ایمیل، نقش، کشور و شهر) عضو می‌شوند.
در اینجا باید نکات زیر را رعایت کنید و تغییراتی را در کد خود اعمال کنید:

- فرمت ورودی‌ها (مانند فرمت ایمیل و کاراکترهای قابل قبول برای فیلدها) تا جای ممکن در سمت فرانت باید اعتبارسنجی شوند. این اتفاق باید در بک‌اند نیز رخ داده تا در برابر ریکوئست‌های دستی مقاوم باشیم.
- بررسی تکراری نبودن نام کاربری و ایمیل همچنان در سمت سرور الزامیست و در صورت تکراری بودن، خطای مناسب به کاربر نمایش داده می‌شود.
- پس از ارسال ریکوئست ثبت نام به سرور، رمز عبور کاربر به هیچ وجه نباید به صورت plain text در پایگاه‌داده ذخیره گردد. در پایگاه‌داده باید Hash رمز را ذخیره کنیم (از الگوریتم SHA256 استفاده کنید).
- در هنگام دریافت لیست اولیه کاربران در ابتدای اجرای سرور از طریق این endpoint، حتما رمزها را به صورت Hash شده ذخیره کنید.
- درباره Salt و دلیل استفاده از آن در هنگام ذخیره‌سازی رمز کاربران تحقیق کنید. شما باید در کنار Hash رمز هر کاربر، Salt آن را نیز نگه دارید.

JSON Web Tokens

مدیریت Session

هم‌اکنون شما جهت مدیریت نشست (Session) کاربران، به هر کاربر در هنگام لاگین یک Token تصادفی اختصاص می‌دهید و آن را در Redis ذخیره می‌کنید. استفاده از Redis به جای حافظه RAM به این دلیل است که در صورتی که چند سرور بک‌اند داشته باشیم، ریکوئست کاربر بتواند به هر سرور ارسال شود و داده نشست آنها یکسان باشد.
این کار ملزم به نگهداری یک سرور Redis و ارتباط با آن است که سربارهای خودش را دارد. در اینجا با یک روش بدون حالت (Stateless) برای مدیریت نشست کاربران آشنا می‌شویم.
JWT یا JSON Web Tokens یک استاندارد در این راستا است که به ما اجازه می‌دهد با استفاده از یک Token رمزنگاری‌شده، اطلاعات کاربر را در سمت کلاینت نگه داریم و در هر درخواست همانند قبل، آن را به سرور ارسال کنیم. نکته این روش این است که نیازی به ذخیره‌سازی این Token در سرور نداریم و به دلیل وجود رمزنگاری و داده‌های داخل آن، همچنان امنیت بالایی داریم.
JWT توسط کتابخانه‌ها در اکثر زبان‌های برنامه‌نویسی پشتیبانی شده و می‌توانیم به راحتی از آن استفاده کنیم.

ساختار JWT

یک توکن JWT از سه بخش تشکیل شده است:

- Header: شامل اطلاعات الگوریتم مورد استفاده برای Signature و نوع Token (معمولا JWT).
- Payload: شامل Claim-های JWT است. Claim-ها اطلاعاتی در مورد کاربر، سرور و یا Metadata درباره خود Token اند.
- Signature: شامل امضای دیجیتال جهت صحت‌سنجی JWT است.

هر کدام از این سه بخش در ساختار یک JSON نوشته می‌شوند. سپس هر بخش با Base64-URL انکد شده و با کاراکتر نقطه (.) به هم چسبانده می‌شوند تا یک JWT کامل تولید شود.
برای اطلاعات بیشتر درباره ساختار JWT، این لینک را مطالعه کنید. همچنین در صفحه اصلی آن لینک، می‌توانید بخش‌های مختلف JWT را ادیت کرده و تغییرات را مشاهده کنید.

اطلاعات JWT

در بخش Payload، شما باید حتما Claim-های زیر را داشته باشید:

- iss یا issuer: کسی که توکن را ساخته یا همان سایت MioBook است.
- sub یا subject: کسی که توکن را دریافت کرده یا همان آیدی کاربر است.
- iat یا issued at: زمان تولید توکن است.

- exp یا expiration: تاریخ انقضای توکن است. یک کانستنت برای این تعریف کنید و مقدار پیش‌فرض آن را 1 روز قرار دهید.
- username: یک custom claim که حاوی نام کاربر است.
- email: یک custom claim که حاوی ایمیل کاربر است.

در بخش Signature، از الگوریتم HS256 که ترکیب HMAC و SHA256 است استفاده کنید. کلید استفاده شده جهت رمزنگاری در این الگوریتم باید حداقل 256 بیت باشد. بنابراین با استفاده از یک استرینگ پایه که حداقل 32 کاراکتر دارد، کلید HS256 را تولید کنید.

استفاده از JWT

در ابتدا، وابستگی سرور خود را به Redis از بین برده و آن را حذف کنید.
حال به ازای هر لاگین، با استفاده از کتابخانه `io.jsonwebtoken` برای کاربر یک JWT با مشخصات گفته شده بسازید و در بدنه Response خود به ریکوئست لاگین قرار دهید.
در سمت فرانت، باید مقدار JWT را در Local Storage خود نگه داشته (تا بعد از ریفرش صفحه از بین نرود) و از این به بعد در همه ریکوئست‌ها، در هدر Authorization مقدار آن را به صورت زیر قرار دهید:

```
Authorization: Bearer <TOKEN>
```

در سرور، با خواندن این هدر از ریکوئست، ابتدا درست بودن توکن را صحت‌سنجی کنید (صحت امضا، وجود فیلدها و درستی تایپ آنها، نبودن iat در آینده و همچنین نگذشتن تاریخ انقضای آن را بررسی کنید). در صورت درست نبودن توکن، کاربر را به صفحه لاگین هدایت کنید.
برای این کار می‌توانید از Filter-ها یا Interceptor-ها استفاده کنید تا وجود و صحت توکن را به ازای هر ریکوئست بررسی کنید.

حال می‌توانید با عبور از فیلتر اولیه JWT، کاربر کنونی را از Payload آن دریافت کنید.
در صورتی که کاربر JWT-ای ذخیره‌شده ندارد و یا اینکه به دلیل گذشتن انقضای آن، آن را از حافظه خود پاک کرده است، نباید دسترسی به صفحه‌ای به جز لاگین و ثبت نام داشته باشد.
جهت Logout شدن کاربر، از آنجا که JWT بدون حالت است، کافیست که صرفا توکن مربوطه را از حافظه مرورگر کاربر پاک کنیم.
توجه کنید که محتوای Header و Payload یک JWT توسط هر کسی قابل رویت می‌باشد و نباید اطلاعات محرمانه داخل آنها قرار گیرد. Signature فقط توسط سرور که کلید را دارد قابل تولید است و کاربر با تغییر Payload نمی‌تواند امضا را نیز تغییر دهد و بنابراین، توکن او نامعتبر می‌شود.

Google OAuth

در این بخش، شما باید امکان ورود کاربر با استفاده از سرویس Google را پیاده‌سازی نمایید.
این امکان با استفاده از استاندارد OAuth 2.0 توسط بسیاری از شرکت‌های بزرگ از جمله گوگل ارائه می‌شود.
جهت آشنایی بیشتر با این استاندارد این لینک را مطالعه کنید.

سیستم در سمت کلاینت باید به گونه‌ای برنامه‌ریزی شود که کاربر به گوگل ری‌دایرکت شود، و بعد از لاگین کاربر در گوگل، شما به سایت مبدأ ری‌دایرکت شده و مستقیماً توکن JWT را دریافت می‌کنید.
این یعنی شما در صفحه لاگین یک دکمه جدید Login with Google قرار می‌دهید که با کلیک روی آن، کاربر به گوگل ری‌دایرکت می‌شود و پروسه OAuth شروع می‌شود.
یک آدرس نمونه برای احراز هویت اولیه (لینک دکمه جدید در صفحه لاگین) به شکل زیر می‌باشد:

```
https://accounts.google.com/o/oauth2/auth?client_id=CLIENT_ID&response_type=code&
scope=SCOPE&redirect_uri=URI
```

در ابتدای کار، شما باید یک OAuth Application در گوگل ایجاد کنید. برای این کار این لینک را مطالعه کنید.
پس از ساختن آن، یک Client ID و یک Client Secret به شما داده می‌شود. توجه کنید که Client Secret نباید به هیچ وجه در فرانت پروژه قرار گیرد و به طور محرمانه فقط در دسترس بک‌اند می‌باشد.

حال لازم است که یک صفحه Callback در بخش فرانت‌اند و یک اندپوینت Callback در بخش بک‌اند پروژه خود پیاده‌سازی کنید.
صفحه Callback فرانت، یک صفحه خالی (مثلا با متن ...Redirecting) است. گوگل پس از احراز هویت، کاربر را به همراه یک سری Query Parameter به این صفحه ری‌دایرکت می‌کند. یکی از این پارامترها، پارامتر "code" است که باید بلافاصله پس از لود شدن صفحه Callback، از URL برداشته شده و به اندپوینت Callback در بک‌اند فرستاده شود.
مطمئن شوید که مسیر صفحه Callback فرانت را در تنظیمات OAuth Application خود تنظیم کرده‌اید.
در اندپوینت Callback بک‌اند، یک درخواست به گوگل زده می‌شود که شامل code دریافتی از فرانت، Client ID و Client Secret است. گوگل در جواب این ریکوئست یک Access Token برای کاربر ایجاد می‌کند.
این ریکوئست به صورت زیر می‌باشد:

```
HEADER: Accept: application/json
https://oauth2.googleapis.com/token?client_id=CLIENT_ID&client_secret=CLIENT_SECRET&c
ode=CODE&grant_type=authorization_code
```

پس از دریافت Access Token، بک‌اند می‌تواند ریکوئست‌های دیگری به گوگل بزند و اطلاعات کاربر را (چنانچه در scope توکن هست) به دست آورد. یک API برای اطلاعات کلی کاربر به صورت زیر است:

```
HEADER: Authorization: Bearer ACCESS_TOKEN
https://www.googleapis.com/oauth2/v2/userinfo
```

شما می‌توانید نام و ایمیل کاربر را از این endpoint به دست آورده و در صورتی که کاربر وجود ندارد، یک کاربر جدید در پایگاه‌داده ایجاد کنید و داده‌هایی که ندارید (از جمله رمز عبور) را null گذارید.
در صورتی که کاربری با نام یا ایمیل دریافتی از گوگل وجود داشت، همانند قبل یک JWT از روی کاربر بسازید و به فرانت ارسال کنید.
حال فرانت JWT را سیو کرده و از صفحه Callback خود خارج می‌شود و کاربر لاگین‌شده را به صفحه خانه هدایت می‌کند.

نکات پایانی

- این تمرین در گروه‌های حداکثر دو نفره انجام می‌شود. برای تحویل آن کافی است که یکی از اعضای گروه، لینک مخزن گیت‌هاب و Hash مربوط به آخرین کامیت پروژه را در سایت درس آپلود کند. پروژه شما بر روی این کامیت مورد ارزیابی قرار می‌گیرد.
- حتما کاربر IE-S04 را به پروژه خود اضافه کنید.
- ساختار مناسب و تمیزی کد برنامه، بخشی از نمره همه پروژه‌های شما خواهد بود. بنابراین در طراحی ساختار برنامه و همچنین خوانایی کد دقت زیادی به خرج دهید.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت مشاهده شباهت 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 گروه 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 تمرین ارتباط برقرار کنید.