

تهیه کننده : شاهین قاسمی : 9413088028

سعی شده تمام فایلها و محتویات آنها به ترتیب توضیح داده شوند :

Front:

Index.html: این فایل html همان فایل صفحه اول (سرچ) میباشد که در مسیر "/" قرار دارد.

Models: این پوشه حاوی مدلها و مدلها (schema) برای پایگاه داده میباشد.

doc.js: این مدل برای ذخیره صفحات خزش شده مورد استفاده قرار میگیرد.

link.js: وقتی هر صفحه ای را خزش و ذخیره میکنیم ، یک فیلد به نام links دارد که شامل تمام لینکهای موجود در آن صفحه میباشد . این لینک ها هرکدام به تنهایی در این مدل ذخیره میشوند.

test.js: این داده برای کارهای آموزشی و توسعه مورد استفاده قرار گرفته است.

crawl.js: این فایل حاوی تابع **firstCrawl()** میباشد. همانطور که از اسمش معلوم میباشد این تابع عملیات مربوط به خزش را انجام میدهد. این تابع قبل ذخیره هر صفحه خزش شده، چک میکند که آیا این صفحه قبلا در پایگاه داده ذخیره شده است یا خیر تا از ذخیره سازی صفحات تکراری جلوگیری شود. این تابع علاوه بر ذخیره سازی صفحات خزش شده، لینکهای داخل این صفحات را در مدلهای **link.js** ذخیره میکند. ورودی این تابع یک url میباشد.

loop.js: این فایل تمام لینکهایی که تا به الان خزش نشده اند (یعنی فیلد visited آنها false میباشد) را از پایگاه داده میگیرد و برای هرکدامشان جداگانه تابع **firstCrawl()** را اجرا میکند.

parsing.js: این فایل یک شی برمیگرداند که دارای دو فیلد مهم میباشد. فیلد **baseURL** که برابر است با لینک سایت ویکیپدیای فارسی. از این فیلد برای ساختن لینک کامل (<https://fa.wikipedia.org/wiki/something>) استفاده میشود.

فیلد **gather** که یک تابع میباشد و از **DOM** برای خزش صفحه و بدست آوردن اطلاعات مربوط استفاده میکند. برای خزش صفحات از کتابخانه **cheerio** استفاده شده است. این کتابخانه کار با **DOM** را در سمت سرور سهولت میبخشد. این تابع بخشهای مهم مانند **h3, h2, links, paragraphs** و **title** صفحات را استخراج میکند و ذخیره میکند.

server.js: این فایل نمونه (instance) سرور را اجرا میکند و مسیر (route) های مختلف را هندل میکند.

مسیرهای مهم سرور عبارتند از :

"/query": این مسیر برای یافتن نتایج بعد از ارسال درخواست (query) کاربر مورد استفاده قرار میگیرد. در این مسیر ابتدا درخواست فارسی کاربر دریافت میشود و بعد از بهینه کردن درخواست و حذف حروف زائد (مثل به، و،

یا، و...) آن را سمت به پایگاه داده ارسال میکند. توجه شود که برای ارسال نتایج در نسخه وجود دارد یکی استفاده از stream ها و یکی استفاده از http

"/aggregate": این مسیر برای بدست آوردن داده های تکراری در پایگاه داده مورد استفاده قرار میگیرد. Aggregate معادل Group در پایگاه داده های رابطه ای SQL میباشد.

"/": فایل روت میباشد که فایل index.html را برای رندر سمت کلاینت میفرستد تا درخواست کاربر را دریافت و به مسیر **"/query"** بفرستد.

utilize.js: این فایل شامل یک آبجکت با فیلدهای کاربردی میباشد که در سراسر پروژه مکررا از آن استفاده شده است. لینک اتصال به پایگاه داده در این فایل قرار دارد. تابع حذف حروف اضافه و بهینه کردن درخواست (query) در این فایل قرار دارد. تابع ساختن شکل ظاهری نتایج جستجو نیز در این فایل قرار دارد. که هرکدام به راحتی باتوجه به اسمشان قابل تشخیص و بررسی هستند.

کتابخانه های مورد استفاده در پروژه :

Axios.js : این کتابخانه برای فرستادن ریکوئست در سمت سرور استفاده شده است. اینترفیس ساده آن باعث شده که به یکی از انتخاب های خوب برای این کار تبدیل شود.

Cheerio.js : این کتابخانه برای کار با DOM در سمت سرور استفاده شده است. این کتابخانه از کتابخانه jQuery الهام گرفته است. API آن نیز تا حد زیادی به jQuery شباهت دارد.

Express.js : این کتابخانه برای ساختن سرور در node.js مورد استفاده قرار میگیرد و خیلی از کارها را ساده تر میکند.

Moment.js : این کتابخانه برای استفاده از تاریخ و ساعت مورد استفاده قرار گرفته است.

Mongoose.js : این کتابخانه یک ODM برای پایگاه داده MongoDB میباشد. درواقع اینترفیس استفاده از پایگاه داده مونگو را خیلی ساده کرده است و علاوه بر آن کلی امکانات اضافه برای کار با مدلها را در اختیار برنامه نویس قرار میدهد.

با تشکر، شاهین قاسمی : 9413088028