

جابجایی بین صفحات

این ماژول برای جابجایی بین صفحه ها بدون رفرش صفحه با استفاده از HTML5 History API استفاده می‌شود.

[HTML5 History](#)

نحوه کار

با اجرای `Navigate` ابتدا رویداد `navigate:start` روی پنجره جرقه می‌خورد، سپس درخواستی به آدرس تعیین شده ارسال می‌شود که شامل مقدار دریافتی `MD5` صفحه از سرور نیز می‌شود. این درخواست با استفاده از تابع `fetch` در فایل این ماژول ارسال می‌شود، این تابع در ادامه توضیح داده شده است.

در مرحله بعد `pushState` و یا `replaceState` روی شی `history` اجرا می‌شوند تا در حقیقت مرورگر رکورد جدیدی به تاریخ خود اضافه کند. سپس صفحه با توجه به پاسخ دریافتی رندر می‌شود که خود مرحله‌ای دارد و در ادامه، در بخش `render` توضیح داده شده است.

سپس رویداد `statechange` بر روی پنجره جرقه می‌خورد.

استفاده

استفاده از این ماژول ساده است ولی تنظیمات زیادی وجود دارند که امکان استفاده در شرایط مختلف را به شما می‌دهند. این تابع یک `Promise` باز می‌گرداند که در اتمام عملیات حل می‌شود.

```
Navigate({
  url: '',
  html: '',
  title: null,
  replace: false,
  fake: false,
  data: false,
  filter: null,
  ajax: {
    type: 'get'
  }
})
```

url

این مشخصه تعیین کننده آدرس مورد نظر برای درخواست و ایجاد رکورد در تاریخ مرورگر است.

در صورتی که این مشخصه تعیین شده باشد، مرحله ارسال درخواست به سرور رد شده و به صورت مستقیم مرحله `render` روی محتویات این مشخصه انجام می‌گیرد.

title

در صورتی که می‌خواهید تیتتر صفحه مقصد را به صورت دستی تغییر دهید از این مشخصه استفاده کنید.

replace

در صورتی که این مشخصه مقدار مثبت داشته باشد، بجای `pushState` از `replaceState` استفاده می‌شود.

fake

در صورت مثبت بودن این مشخصه نه درخواستی ارسال می‌شود و نه رندری صورت می‌گیرد، بلکه فقط آدرس صفحه حاضر تغییر می‌کند.

data

در صورتی که درخواستی را فقط برای دریافت اطلاعات از آدرس مورد نظر اجرا می‌کنید، و نمی‌خواهید مرحله رندر انجام پذیرد از این مشخصه استفاده کنید.

filter

در صورت تعیین این مشخصه، در هنگام رندر عناصری که مقدار `data-xhr` آنها یکی از مقادیر تعیین شده در این مشخصه (می‌تواند آرایه یا رشته باشد) رندر می‌شوند و از رندر بقیه صرف نظر می‌شود.

ajax

این شی با تنظیمات پیش‌فرض ادغام می‌شود و به `jQuery.ajax` داده می‌شود.

روال‌ها

fetch → Promise

(properties: Object, md5: String)

این روال درخواستی به صفحه مورد نظر می‌فرستد و سپس در صورت امکان، اطلاعات `JSON` را از صفحه استخراج می‌کند. همچنین در صورت دریافت پاسخی از سرور مبنی بر نبود تغییری در صفحه مورد نظر (از روی MD5) صفحه را از `localStorage` می‌خواند.

در شروع کار این روال رویداد `navigate:fetch:start`، پس از اجرای `jQuery.ajax` رویداد `navigate:fetch:ajax:start` و در پایان کار درخواست `ایژکس` رویداد

`navigate:fetch:done` و در آخر، رویداد `navigate:fetch:ajax:done` جرقه می‌خورند.

دو رویداد پایانی اطلاعات استخراج شده از صفحه را به عنوان آرگومانی دریافت می‌کنند.

render

(data: Object)

این روال وظیفه‌ی رندر صفحه را دارد که شامل قرار دادن عناصر در محل مناسب خود می‌شود.

در ابتدای اجرا، رویداد `navigate:render:start` جرقه می‌خورد. سپس آی‌دی بادی با توجه به مقدار `id` دریافتی از سرور تغییر می‌کند.

در مرحله بعد، قبل از فیلتر کردن عناصر رویداد `navigate:render:filter:before` اجرا می‌شود که شی فیلترهای داده‌شده به عنصر را به عنوان آرگومان می‌دهد. در پایان فیلتر رویداد `navigate:render:filter:done` اجرا می‌شود.

در مرحله بعد مقدار تیتل صفحه تغییر می‌کند.

سپس در صورت وجود شی `js` در پاسخ سرور، عناصر اسکریپت مورد نیاز ساخته و به صفحه اضافه می‌شوند. قبل از شروع این عملیات رویداد `navigate:render:scripts:before` را آرگومان همان مقدار دریافتی از سرور اجرا می‌شوند.

سپس در ساخت هر عنصر اسکریپت رویداد `navigate:render:script:created` با عنصر ساخته شده اجرا می‌شود. پس از اضافه کردن این اسکریپت به بدنه `navigate:render:script:appended` اجرا می‌شود.

پس از اتمام اضافه کردن تمام اسکریپت‌ها رویداد `navigate:render:scripts:done` جرقه می‌خورد.

در پایان نیز `sRoute` اجرا شده و رویداد `navigate:render:done` جرقه می‌خورد.