

ایژکس

وظیفه این ماژول استفاده از فناوری Ajax برای ارسال فرم ها و درخواست های بدون ورودی (لینک) به سرور است.

نحوه کار

نحوه کار این ماژول نسبتا ساده است و نیاز به تنظیمات خاصی از سمت شما ندارد و تنها کاری که شما باید انجام دهید اضافه کردن تگ های `*-data` به عناصر HTML برای تعیین تنظیمات است.

رویداد های این ماژول روی عنصری که رویداد روی آن اجرا شده جرقه می خورند.

مراحل کاری

مراحل کاری این ماژول برای لینک ها و فرم ها متفاوت است:

در ابتدای اجرای این متد رویداد `ajaxify:init` و سپس `ajaxify:send:before` جرقه می خورند.

فرم

برای فرم ها، متد ارسال درخواست از مشخصه `method` و آدرس هدف از مشخصه `action` فرم خوانده می شود.

سپس اطلاعات فرم به صورت `FormData` در می آید و درخواست ایژکس ارسال می شود. پس از ارسال درخواست، تمامی ورودی های از نوع `input` و عناصر `[contenteditable]` غیر فعال می شوند.

لینک

برای لینک ها اطلاعات درخواست از جمله متد و آدرس از مشخصه های `data-method` و `href` خوانده می شوند.

اطلاعات درخواست با اجرای `JSON.parse` بر روی مشخصه `data-data` لینک خوانده شده و همراه با داده ها، این تنظیمات به `jQuery.ajax` داده می شوند:

```
{
  data: JSON.parse(el.attr('data-data')),
  contentType: 'application/x-www-form-urlencoded; charset=UTF-8',
  processData: true
}
```

قبل از ارسال اطلاعات عناصر `[data-ajaxify]` غیر فعال می‌شوند.

مشترک

بعد از جمع‌آوری اطلاعات در مورد درخواست رویداد `ajaxify:send:ajax:start` جرقه خورده و `jQuery.ajax` اجرا می‌شود.

پس از دریافت پاسخ از سمت سرور رویداد `ajaxify:complete` جرقه می‌خورد، عناصر غیر فعال شده به حالت خود بازمی‌گردند و کلاس `loading-form` از بادی حذف می‌شود.

پس از این مراحل، بسته به پاسخ دریافتی از سرور دو حالت وجود دارد:

موفقیت

در صورت موفق بودن درخواست متد `ajaxify.showResults` اجرا می‌شود که نوتیفیکیشن مناسب را نمایش می‌دهد.

سپس در صورت وجود مشخصه `msg.redirect` در جواب، کاربر به صفحه مورد نظر منتقل می‌شود.

در صورت وجود مشخصه `refresh` هنگام اجرای متد و یا وجود مشخصه `data-refresh` روی عنصر صفحه رفرش می‌شود.

رویداد `ajaxify:success` جرقه می‌خورد.

اختلال

رویداد `ajaxify:fail` جرقه می‌خورد.

استفاده

برای استفاده از این ماژول شما از متد `ajaxify` اضافه شده به جی‌کوری استفاده می‌کنید که خواندن و ارسال داده‌ها و در نهایت نمایش نتیجه را بر عهده دارد.

این متد یک شی به عنوان آرگومان قبول می‌کند که شامل تنظیمات می‌شود. تنظیمات این متد در زیر توضیح داده شده‌اند:

```
$('#myform').ajaxify({
  ajax: {
    cache: true,
    ...
  },
  noLoading: false,
  link: false,
  refresh: false
});
```

ajax

این شی با تنظیمات پیش فرض ایترکسی ادغام شده و به `jQuery.ajax` داده می‌شود. دقت کنید که مشخصه `data` ی این شی در آخر با اطلاعات گرفته شده از عنصر جایگزین می‌شود پس استفاده از این مشخصه بی‌فایده و نادرست است. بجای اینکار باید اطلاعات مورد نظرتان را به عنصر اضافه کنید (برای مثال اضافه کردن ورودی‌های مخفی به فرم).

noLoading

در صورت مثبت بودن این مشخصه از اضافه کردن کلاس `loading-form` به بدنه صفحه خودداری می‌شود.

link

این مشخصه تعیین می‌کند که رفتار متد باید مانند لینک باشد یا خیر.

refresh

در صورت مثبت بودن این مشخصه بعد از دریافت پاسخ برای درخواست، صفحه با استفاده از دستور `Navigate` رفرش می‌شود.

متدها

showResults

(data: Object, form: DOMNode, jq: jQuery Object)

این متد وظیفه‌ی نمایش نتیجه‌ی درخواست را دارد. در اجرای این متد رویداد `ajaxify:render:start` جرقه می‌خورد.

در ابتدا این متد کلاس‌های `error` و `warn` را از روی تمامی ورودی‌های فرم حذف می‌کند.

سپس یک عنصر `ul` ساخته می‌شود که با اطلاعات `data.messages` پر می‌شود. برای اطلاع از نحوه رندر نوتیفیکشن به فایل `includes/mvc/display.html` مراجعه کنید.

```

<div id="formError" class="error_{{debug.status}} unselectable">
{%for key, mydebug in debug.messages if mydebug is iterable %}
  <ul class="{{key}}">
    {%for key, err in mydebug%}
      {%if err.title is defined%}
        <li class="{{err.group}} {{err.redirect}}">{{err.title}}</li>
      {%else%}
        <li class="n_{{key}}">{{err}}</li>
      {%endif%}
    {%endfor%}
  </ul>
{%endfor%}
</div>

```

پس از اتمام پر کردن عناصر رویداد `ajaxify:render:done` جرقه می‌خورد.

در صورتی که خطایی در فرم وجود نداشته باشد و مشخصه `data-clear` روی عنصر مشخص شده باشد، مقدار تمام عناصری که مشخصه `data-unclear` را نداشته باشند خالی می‌شود. رویداد `ajaxify:render:clear` اجرا می‌شود.

سپس در صورت نبودن خطا، اولین `input` در فرم دارای فوکوس می‌شود. رویداد `ajaxify:render:focus` اجرا می‌شود.

سپس با استفاده از `notify` پیغام نمایش داده می‌شود و رویداد `ajaxify:notify` جرقه می‌خورد.

در صورت مشخص کردن `data-delay` این تنظیم به `notify` داده می‌شود که تعیین کننده‌ی زمان ماندن روی صفحه‌ی نوتیفیکیشن است.