

Lecture Notes - 07-Vue-Event-Listening

The focus of today's lecture is to add on to the Product Review component. You've already added a form to add a new review.

You'll see there's a new link on the page that says **Show Form**. That link has a **v-on** attached that sets a boolean flag on the component. No need for a method for a function that simple. We need to add the **.prevent** so that the browser doesn't perform its default action, which is to follow the href of the link. There is also a **v-if** that hides that link when the form is shown.

Under that is a form. The form also looks at that boolean flag, showing itself when the **showForm** variable is true. This sets up a toggle. When the link is clicked, it is hidden and the form is shown.

There is also a **Cancel** button at the bottom of that form that does the opposite. When it's clicked, it calls a **method** that sets the **showForm** to false. This hides the form and shows the link again.

You'll notice that the form is handled with a **v-on** for the submit. Like the link, you need to prevent the browser's default action with **.prevent**. This calls the **addNewReview** method, which takes all the data from the form that's saved in the **newReview** data property. Then, it adds it to the front of the **reviews** data property before calling the **resetForm** method.

Now, it's time to add another piece to this application. You'll change it so that if the user clicks on the counts of the star ratings, the reviews filter to only show those reviews. For example, if a user clicks on the number of one-star reviews, they should only see one-star reviews.

How would you do this? Right now, you can see all the reviews in the display. If you only want to show some of the reviews, you'll need another property to show that list of reviews instead of all of them. If you know what star reviews you want to show, show them. Otherwise, show them all.

First, create a new data variable that holds the filter for the reviews:

```
data() {  
  return {  
    name: 'Cigar Parties for Dummies',  
    ...  
    filter: 0,  
    ...  
  }  
}
```

If the filter is set to 0, you'll see all the reviews. If the filter is set to any other number, you'll only see the reviews for that rating.

Next, create a new **computed** property to return these filtered reviews:

```
filteredReviews(vm) {  
  return vm.reviews.filter((review) => {  
    return vm.filter === 0 ? true : vm.filter === review.rating;  
  })  
}
```

```
    });  
  },
```

You can now use this new property in the `v-for` for the `div`:

```
<div class="review" v-for="review in filteredReviews" v-  
  bind:key="review.id">
```

You should see no difference in the view now. Because the filter is set to 0 by default, it shows all the reviews. Now you can set up the text in the rating display to set the filter to the star rating you want. To do this, use a `v-on` on a click:

```
<div class="well">  
  <span class="amount" v-on:click="filter = 1">{{ numberOfOneStarReviews  
}}</span>  
  1 Star Review{{ numberOfOneStarReviews === 1 ? '' : 's' }}  
</div>
```

You can do this for all the filters. When you click on a number, you'll only see the reviews with that number of stars. When you add a new review, the review is automatically shown in either view.

If you want, you can also set the average review number to show all the reviews again. To do this, set the filter back to 0:

```
<div class="well">  
  <span class="amount" v-on:click="filter = 0">{{ averageRating }}  
</span>  
  Average Rating  
</div>
```

When do you use a method versus putting the code in the `v-on`?

`methods` are useful when you have code that you want to execute on an event. However, you can also put code directly in the `v-on`. Why use one over the other?

Putting code in the `v-on` is useful if you're only setting a variable. During Vue development, this makes sense.

You should use `methods` if you need to write more than one line of code or code used in multiple `v-on` bindings. `methods` help to consolidate code in one place and prevent code duplication, which makes maintenance easier.