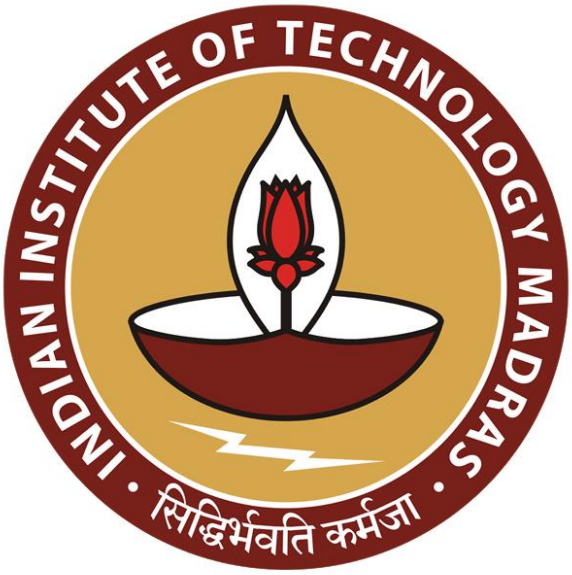
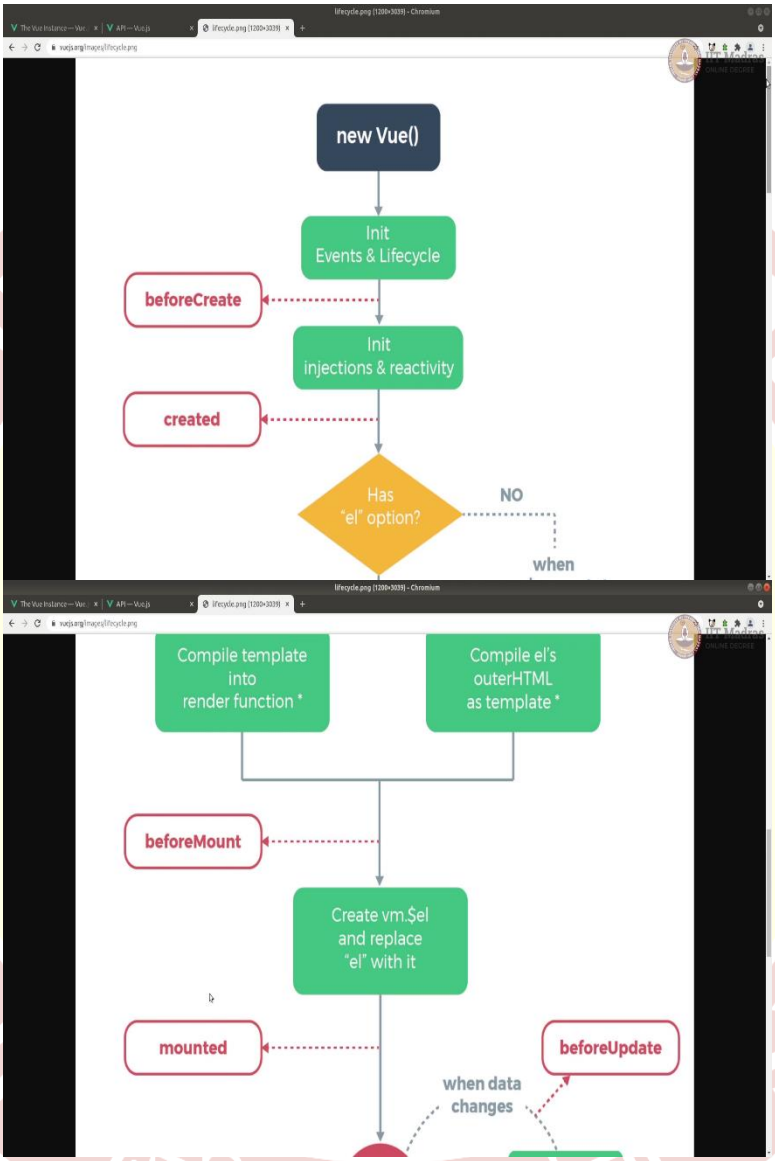


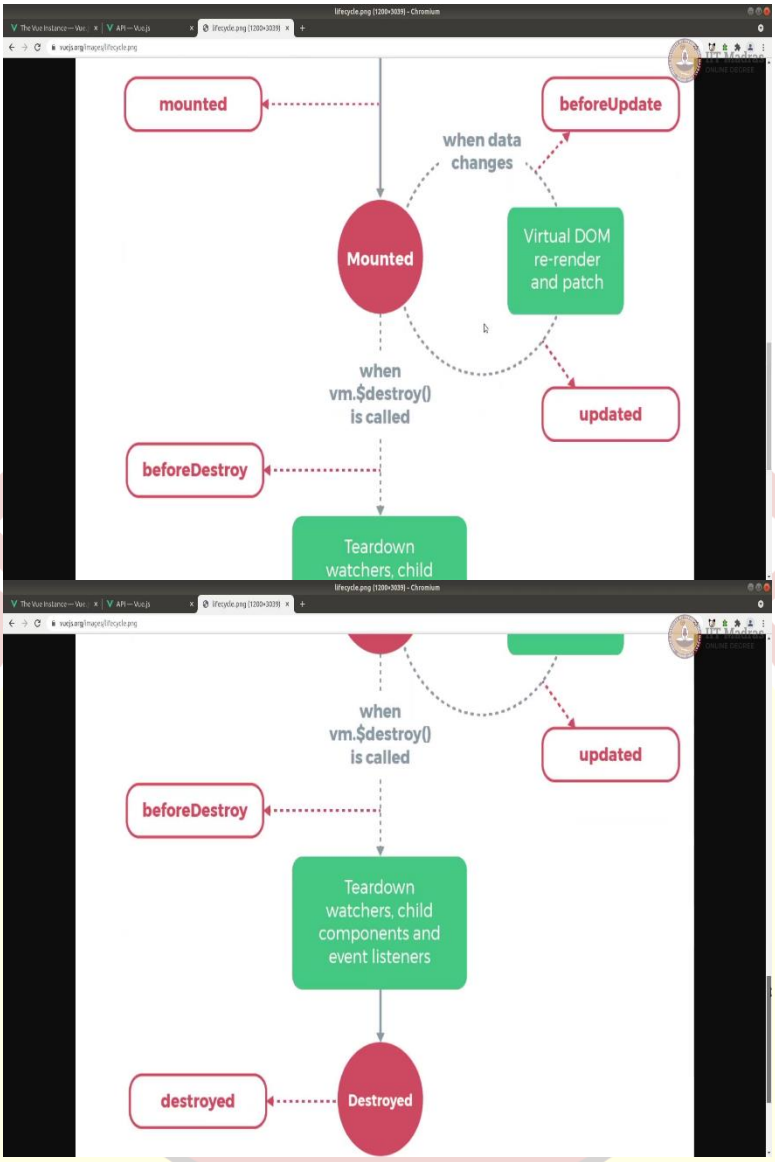
IIT Madras

ONLINE DEGREE

Modern Application Development - II
Professor Thejesh G N
Software Consultant
Indian Institute of Technology, Madras
VueJS Lifecycle

(Refer Slide Time: 00:14)





Welcome to the Modern Application Development II screencast. In this screencast, we will look at lifecycle of Vue app and hooks that we can use. For this you will need editor like Vs Odium or Sublime Text, browser, like Firefox or Chrome, you can install VueJS dev tools, if you think it makes it easy for you to debug.

Here on this screen, you will see an image of Vue lifecycle given to us by the VueJS documentation site. Vue lifecycle, a Vue app or a component goes through a series of steps throughout its life from the initialization or that is creation to getting destroyed or remote. Along the way, it also runs functions called lifecycle hooks, giving us the user's opportunity to add our own code at specific stages of the app or component.

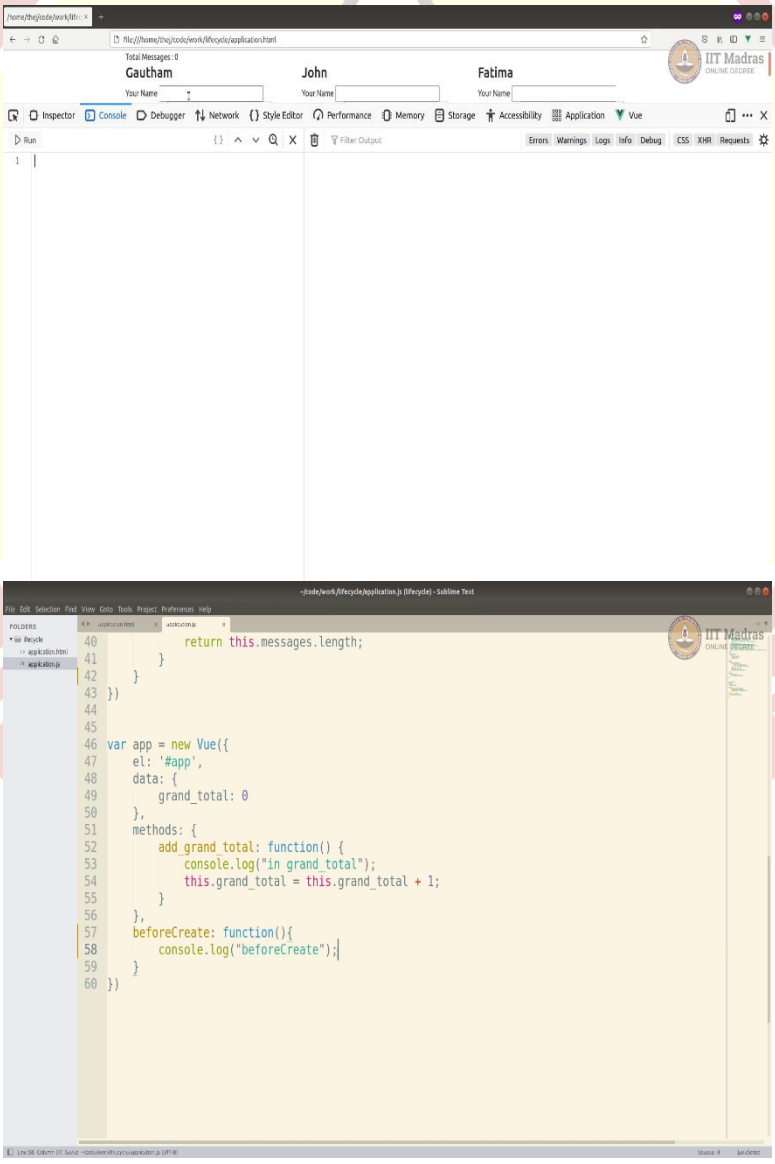
Now let us look at some of the Vue's lifecycle hooks. On the screen, you can see a new Vue app being initialized, before initialization it calls create after that it calls created. All the red

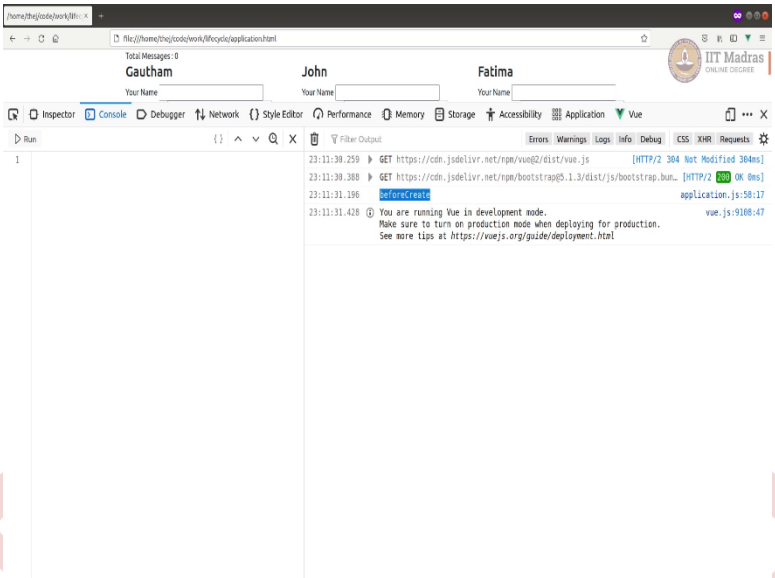
ones are the lifecycle hooks. And then before mounting, it calls before mount after it gets mounted it calls mounted.

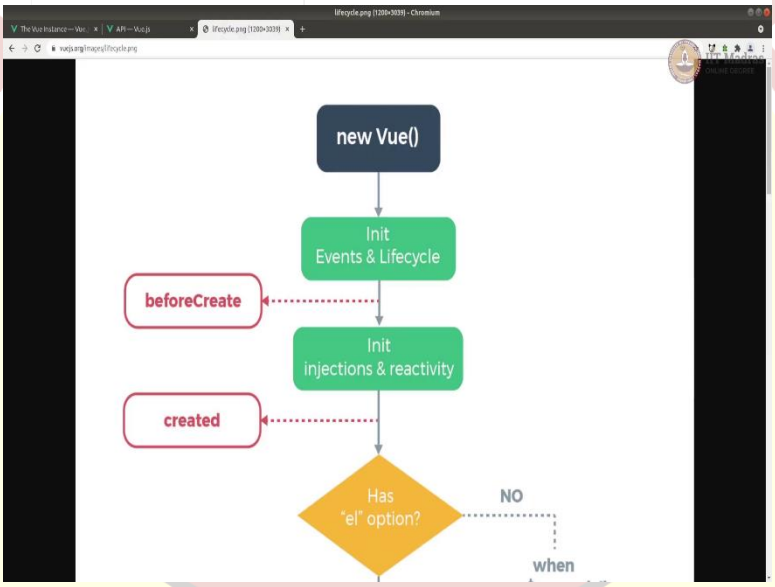
Then, once it is mounted, whenever there is a change in data state, the Virtual DOM gets updated and patched. before it happens before update is called post that updated is called. Now when you exit the page or hide the Vue component or destroy it, before it gets destroyed before destroy is called after it get destroyed, destroyed is called.

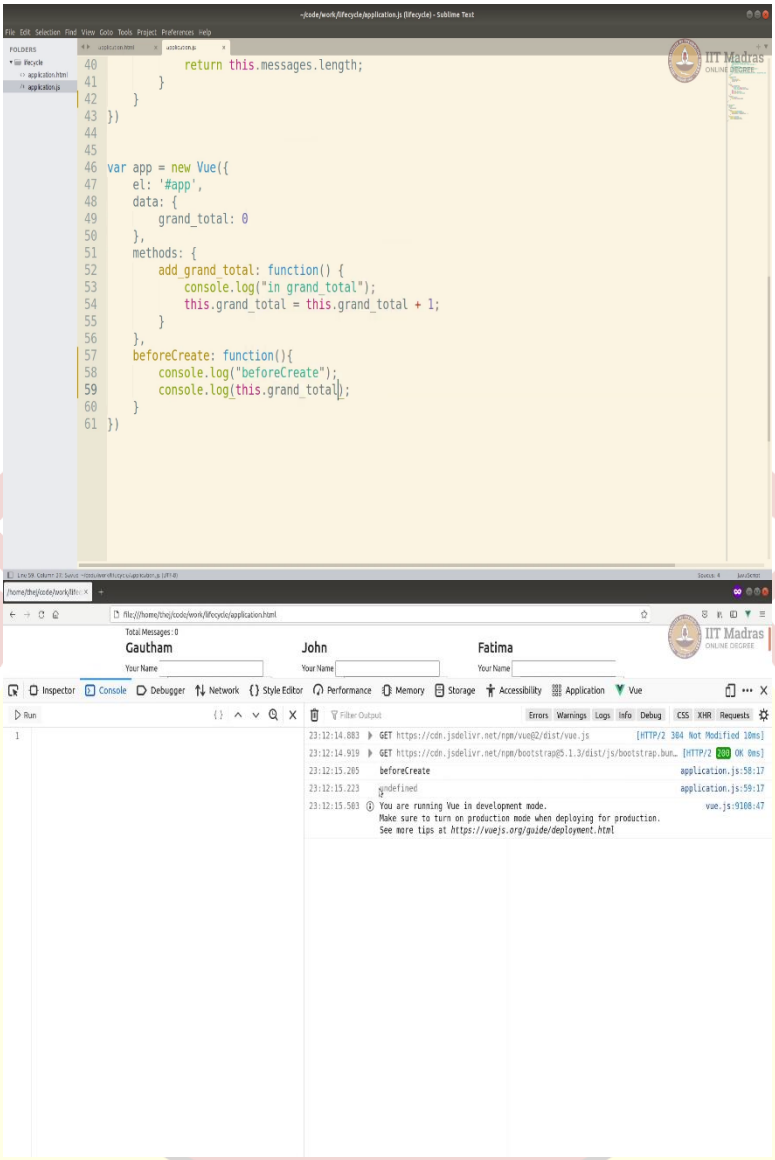
So, you can see, even though 10s of the hooks change. Like for example, before create and then after creation is complete created. So now let us look at one by one, let us add some code to look at all of them, actually, at least four of them and then I will explain to you.

(Refer Slide Time: 02:24)




Total Messages : 0
Gautham
John
Fatima
Your Name
Inspector
Console
Debugger
Network
Style Editor
Performance
Memory
Storage
Accessibility
Application
Vue
beforeCreate
You are running Vue in development mode.
Make sure to turn on production mode when deploying for production.
See more tips at https://vuejs.org/guide/deployment.html

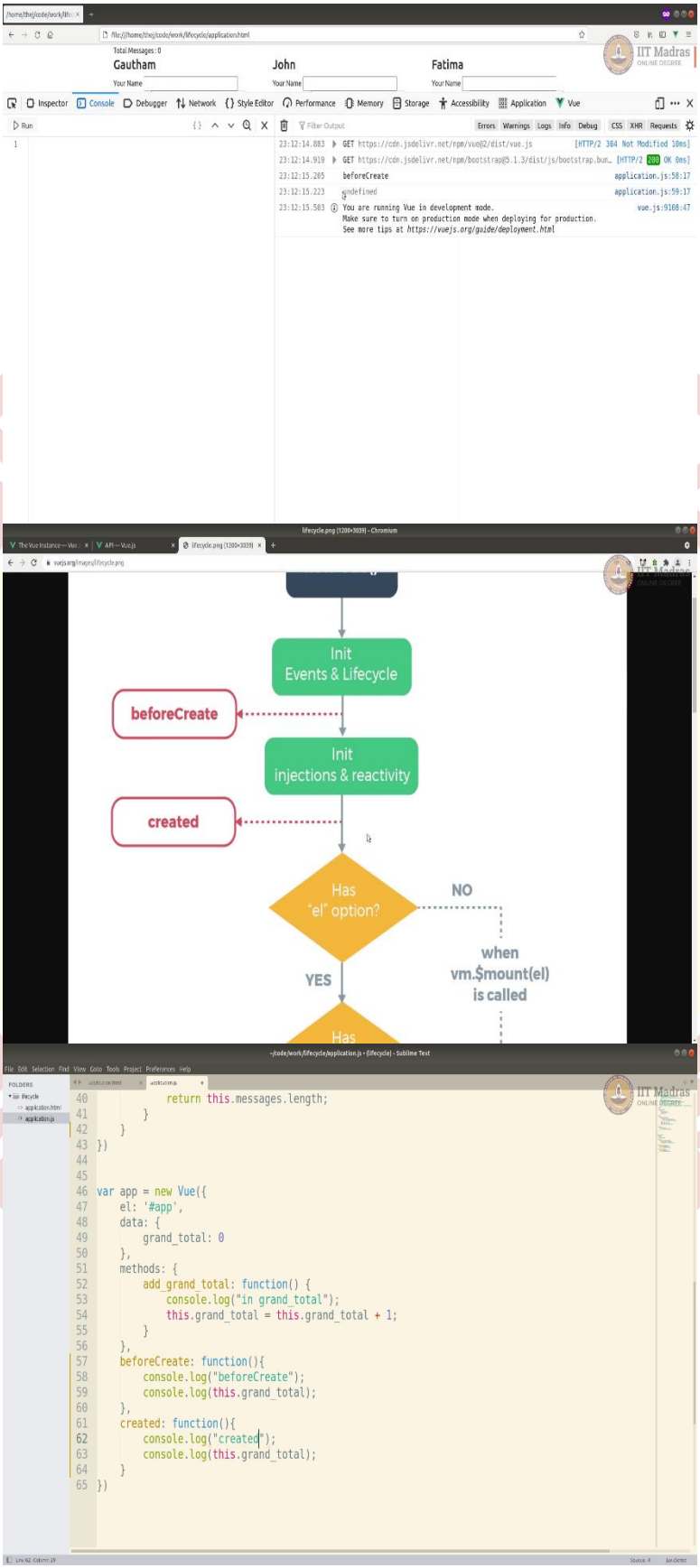

new Vue()
Init
Events & Lifecycle
beforeCreate
Init
injections & reactivity
created
Has
"el" option?
NO
when

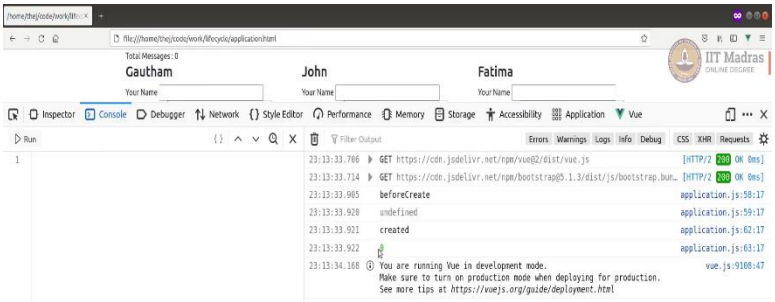


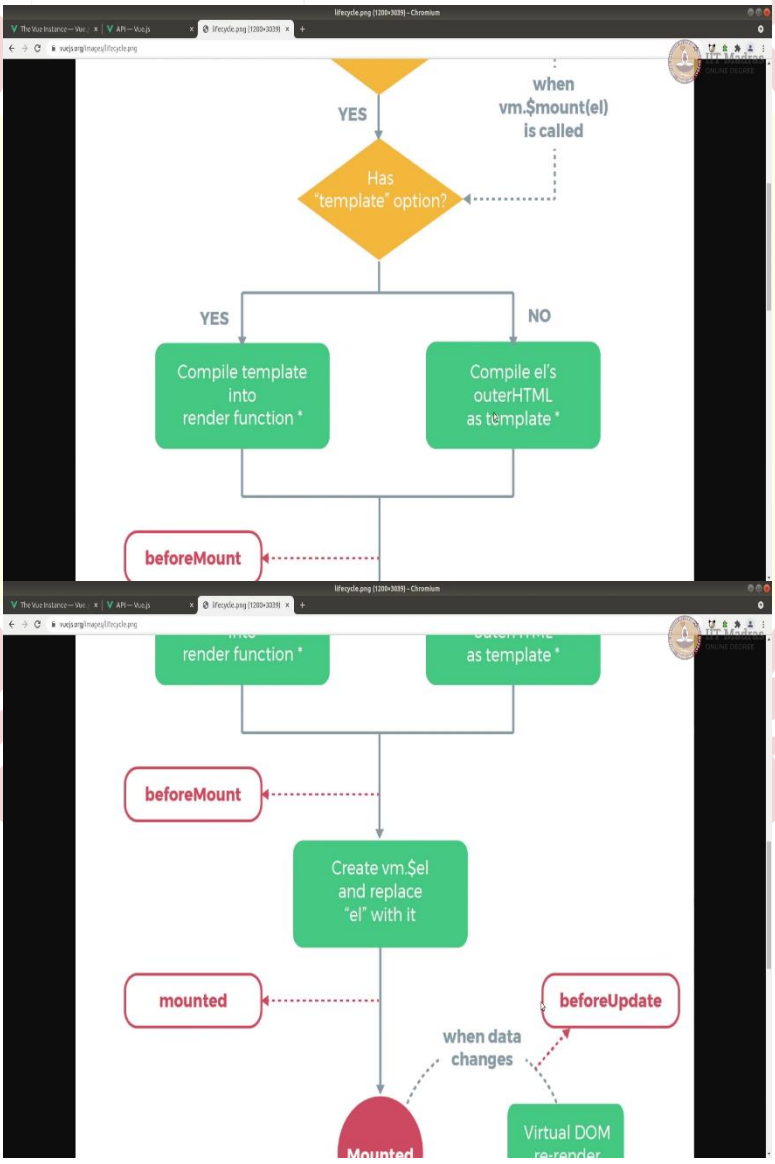
For the ease, I will just write first. Like, like any other function, here is the Vue app, I am using the same app that we had used before. It is a simple app, same no change. I am just going to add some lifecycle hooks to the app and not the component. So, you can create before create hook it is basically a function, which takes nothing currently, and you can do whatever you want to do inside. So, I will just add just a message called before create, and then, I will just refresh.

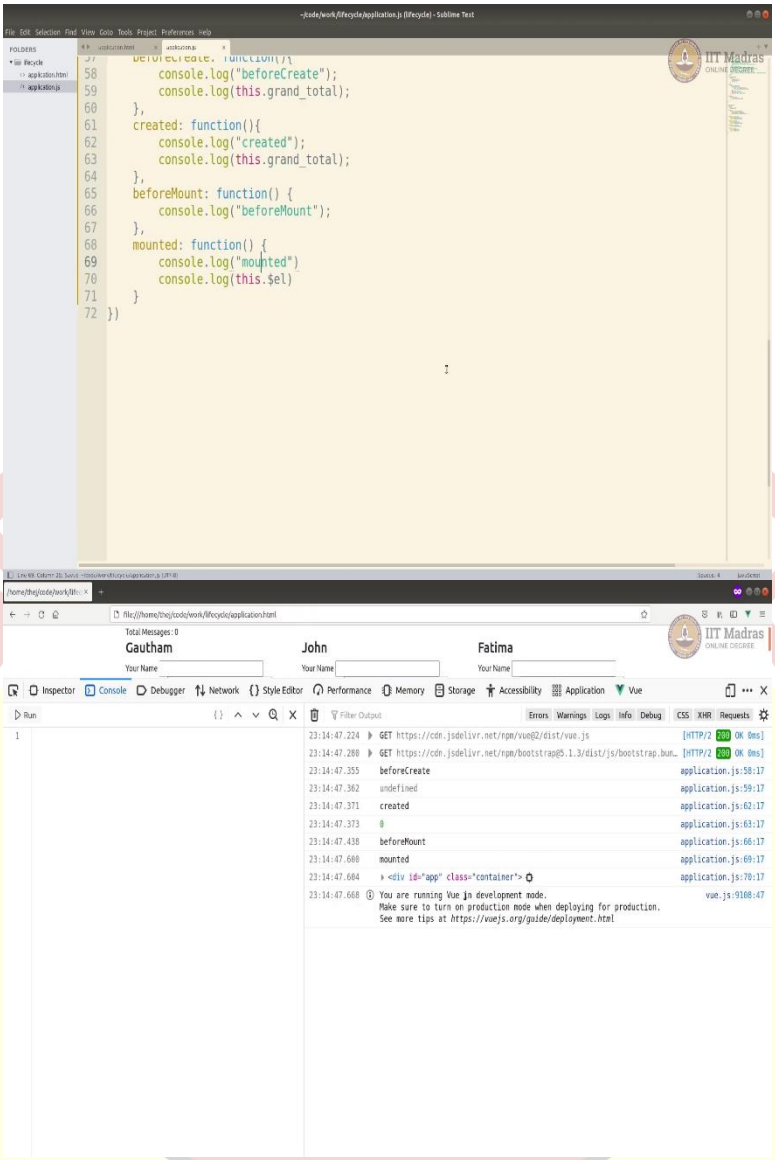
So, you can see that before create got called at the beginning itself. Now before create is actually called synchronously. Here, you can see it is called synchronously, it is called immediately after the instance being initiated. No data observation is set up, no event watcher is set up, hence there is no access to data. So, let us try and print and just check that. Wait I will just print this dot grand total, and I am going to save and refresh. You can see that it is undefined.

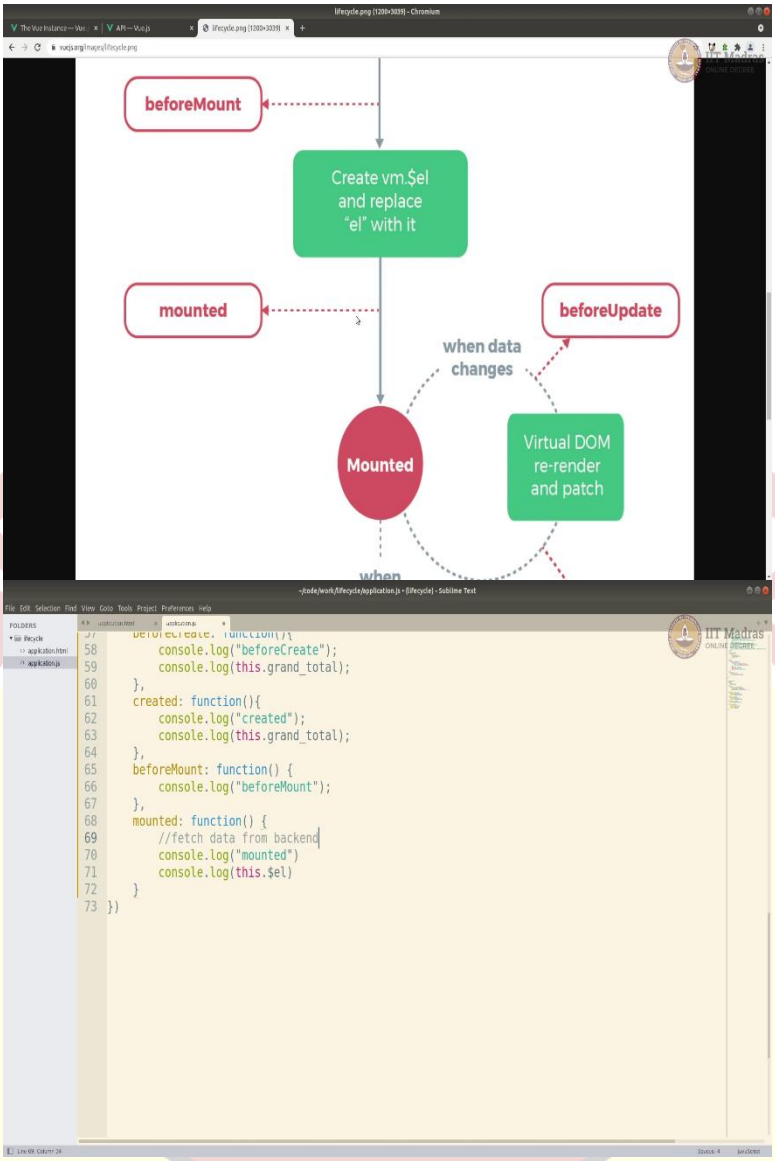
(Refer Slide Time: 03:56)



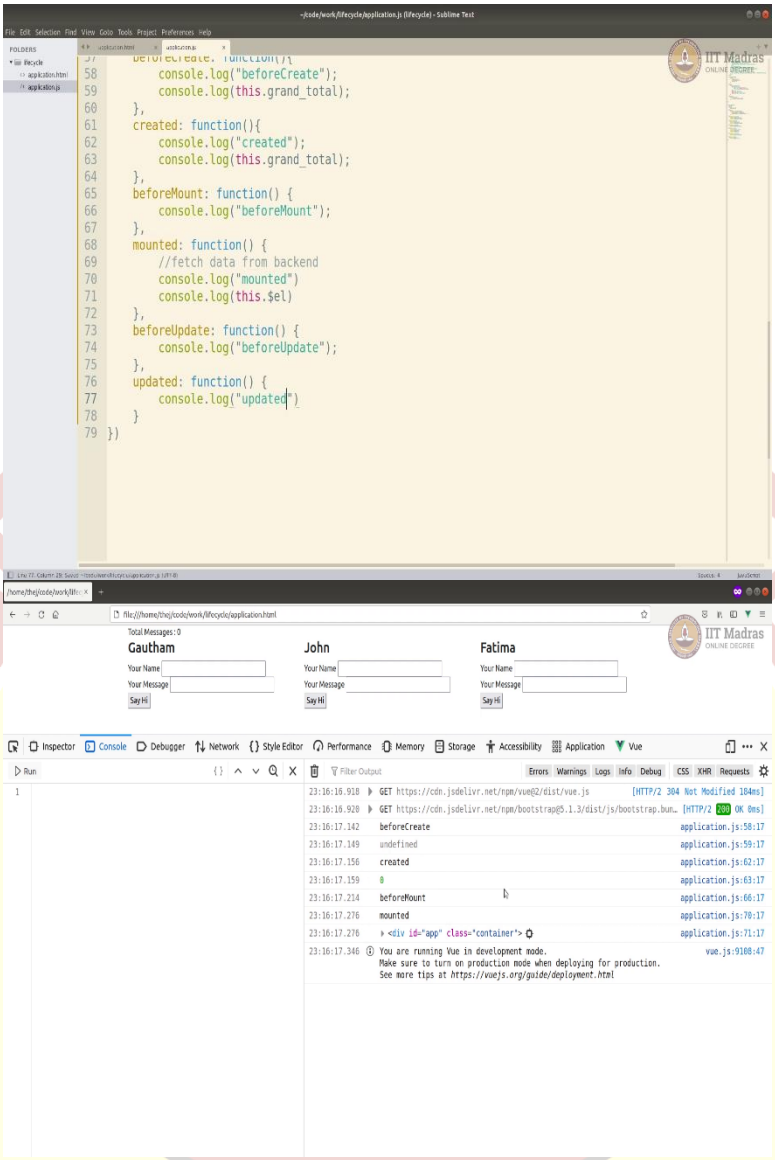

Total Messages : 0
Gautham
John
Fatima
Your Name
Inspector
Console
Debugger
Network
Style Editor
Performance
Memory
Storage
Accessibility
Application
Vue
GET https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js
GET https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bun...
beforeCreate
undefined
created
You are running Vue in development mode.
Make sure to turn on production mode when deploying for production.
See more tips at https://vuejs.org/guide/deployment.html

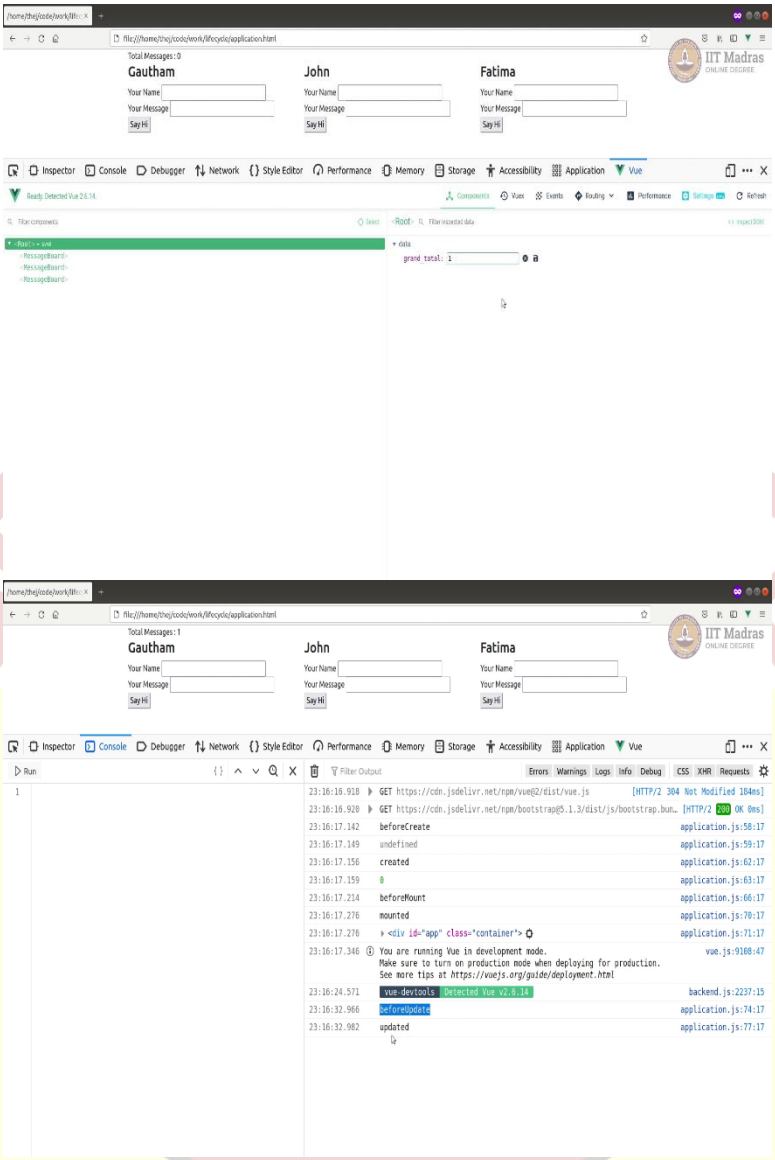

when
vm.$mount(el)
is called
YES
Has
"template" option?
YES
NO
Compile template
into
render function *
Compile el's
outerHTML
as template *
beforeMount
render function *
as template *
beforeMount
Create vm.$el
and replace
"el" with it
mounted
beforeUpdate
when data
changes
Virtual DOM


~/code/work/lifecycle/application.js (lifecycle) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
lifecycle
application.html
application.js
58 console.log("beforeCreate");
59 console.log(this.grand_total);
60 },
61 created: function(){
62 console.log("created");
63 console.log(this.grand_total);
64 },
65 beforeMount: function() {
66 console.log("beforeMount");
67 },
68 mounted: function() {
69 console.log("mounted")
70 console.log(this.$el)
71 }
72 })
IIT Madras
ONLINE DEGREE
file:///home/theji/code/work/lifecycle/application.html
Total Messages : 0
Gautham
Your Name
John
Your Name
Fatima
Your Name
Inspector
Console
Debugger
Network
Style Editor
Performance
Memory
Storage
Accessibility
Application
Vue
Run
Filter Output
Errors
Warnings
Logs
Info
Debug
CSS
XHR
Requests
23:14:47.224 GET https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js [HTTP/2 200 OK 0ms]
23:14:47.280 GET https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bun... [HTTP/2 200 OK 0ms]
23:14:47.355 beforeCreate application.js:58:17
23:14:47.362 undefined application.js:59:17
23:14:47.371 created application.js:62:17
23:14:47.373 0 application.js:63:17
23:14:47.438 beforeMount application.js:66:17
23:14:47.600 mounted application.js:69:17
23:14:47.604 application.js:70:17
23:14:47.668 You are running Vue in development mode.
Make sure to turn on production mode when deploying for production.
See more tips at https://vuejs.org/guide/deployment.html vue.js:9108:47


lifecycle.png (1200×3039) - Chromium
beforeMount
Create vm.$el
and replace
"el" with it
mounted
beforeUpdate
when data
changes
Mounted
Virtual DOM
re-render
and patch
when

~/code/work/lifecycle/application.js - (lifecycle) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
lifecycle
application.html
application.js
58 console.log("beforeCreate");
59 console.log(this.grand_total);
60 },
61 created: function(){
62 console.log("created");
63 console.log(this.grand_total);
64 },
65 beforeMount: function() {
66 console.log("beforeMount");
67 },
68 mounted: function() {
69 //fetch data from backend
70 console.log("mounted")
71 console.log(this.$el)
72 }
73 })


~/code/work/lifecycle/application.js (lifecycle) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
lifecycle
application.html
application.js
beforeCreate: function(){
console.log("beforeCreate");
console.log(this.grand_total);
},
created: function(){
console.log("created");
console.log(this.grand_total);
},
beforeMount: function() {
console.log("beforeMount");
},
mounted: function() {
//fetch data from backend
console.log("mounted")
console.log(this.$el)
},
beforeUpdate: function() {
console.log("beforeUpdate");
},
updated: function() {
console.log("updated")
}
})
IIT Madras
ONLINE DEGREE
file:///home/thej/code/work/lifecycle/application.html
Total Messages : 0
Gautham
Your Name
Your Message
Say Hi
John
Your Name
Your Message
Say Hi
Fatima
Your Name
Your Message
Say Hi
Inspector
Console
Debugger
Network
Style Editor
Performance
Memory
Storage
Accessibility
Application
Vue
Run
Filter Output
Errors
Warnings
Logs
Info
Debug
CSS
XHR
Requests
23:16:16.918 GET https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js [HTTP/2 304 Not Modified 184ms]
23:16:16.920 GET https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bun… [HTTP/2 200 OK 0ms]
23:16:17.142 beforeCreate application.js:58:17
23:16:17.149 undefined application.js:59:17
23:16:17.156 created application.js:62:17
23:16:17.159 0 application.js:63:17
23:16:17.214 beforeMount application.js:66:17
23:16:17.276 mounted application.js:70:17
23:16:17.276 application.js:71:17
23:16:17.346 You are running Vue in development mode.
Make sure to turn on production mode when deploying for production.
See more tips at https://vuejs.org/guide/deployment.html vue.js:9108:47



Next event hook or the lifecycle hook is created. It happens after the initialization of injections, and the activity is complete. This is also called in synchronous mode after the application or the component has been created. The, at this point data observation, computed properties, method, watch, events, call backs are set up. But it is not yet mounted. At this point, you would not have access to El, but you will have access to data, so you can use the data.

This is also good point or a place where you can fetch some data from the back. So, you can show some data to the user before the app gets shown to the user. So, let us just check that whether we have access to the data. So, I am just going to copy paste the same thing here and call it created and refresh this.

You can see that, you know, it has been initialized to 0, and we have access to that grand total. So, this is also one of the places where you can call, you know, like maybe fetch to get

the data from backend. We will look at Mount as well later where you can do that. But yeah, you this is one of the step.

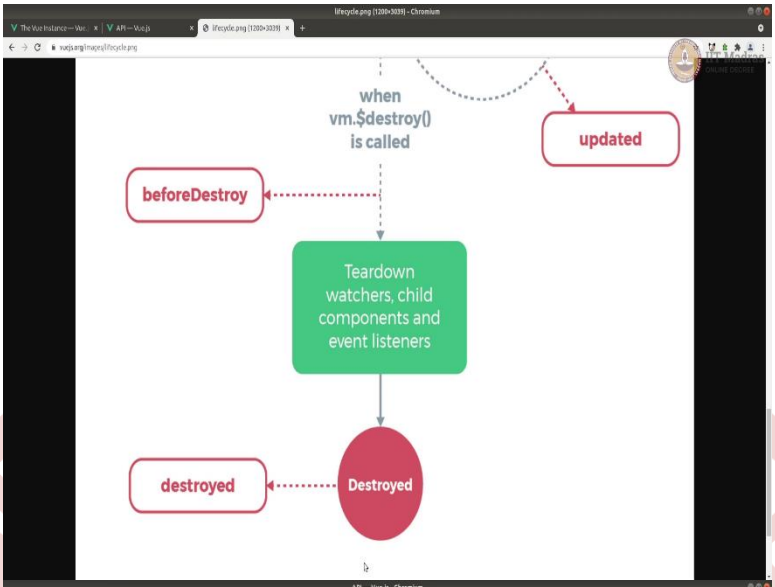
After create, we are going to get another lifecycle event called beforemount. So, here beforemount. This happens, as soon as there is a compilation and template is ready to render, but it is not completely ready to replace the el with. So, at this point before mount is called, once it is mounted, the mounted lifecycle hook is called, so I will just add both of them, so it is quick for us to check it, beforemount and mount mounted. Same tents before mount and mounted.

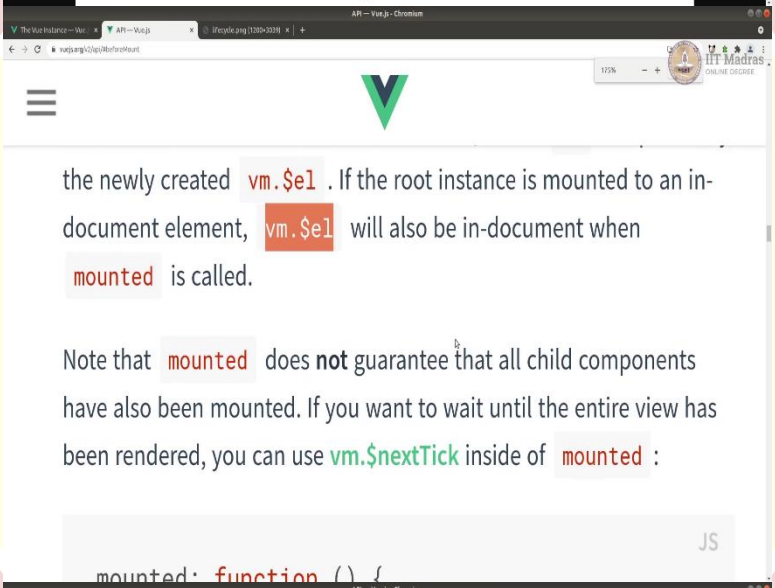
So, you can see it has been mounted, and we have access to el. At this point more than access to el the el is replaced by our virtual el. So that way, we have the control on it, now we can manipulate it or modify it as we want. So mounted is another place where you can do fetch, like you can also do fetch, fetch data from backend for example and do necessary things and show it to the user.

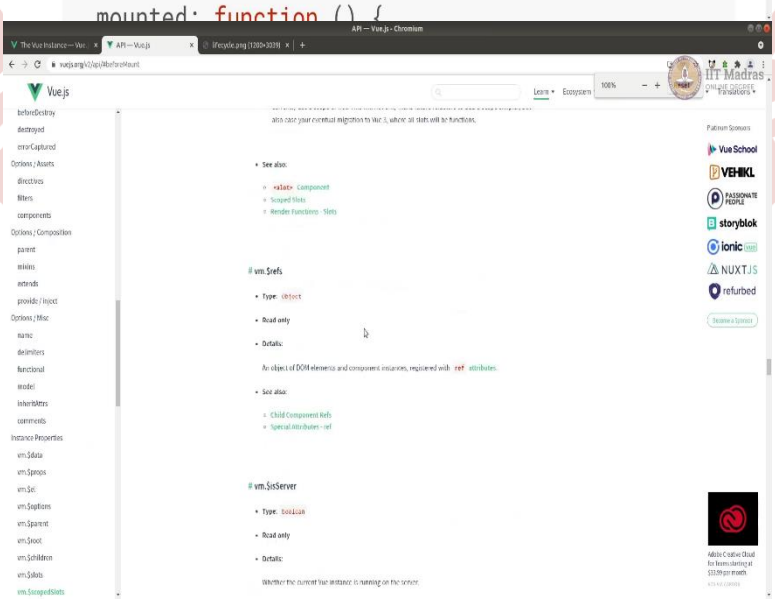
And now before update and update happens every time there is a change to virtual DOM. For example, if I change the data, then virtual DOM changes, and then these two get called. Let us do that also. Before update, no this is not required updated.

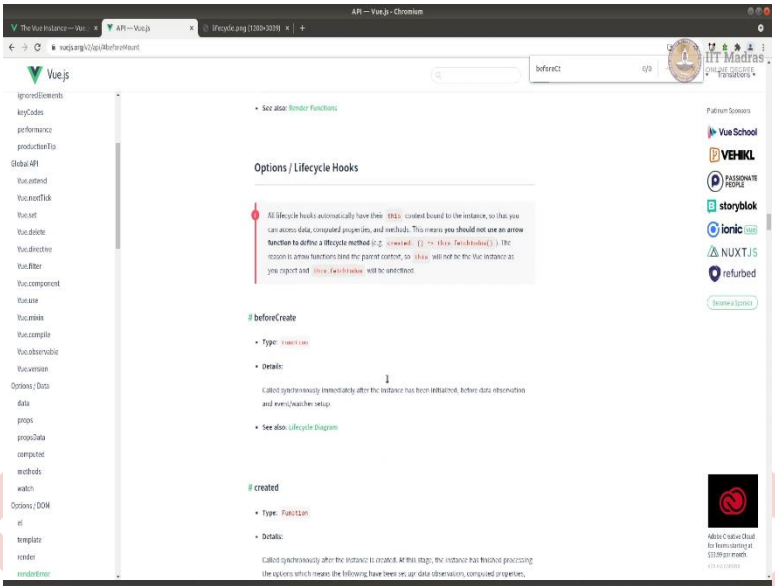
Now let us go here. I will clear it and run it. Now it is not called yet. Now let us go to Vue and make it 1. So, it got changed here. Now if you go to console, you can see that before update and update got called that is because we changed the data, and the data in turn changed the virtual DOM and it is before update and updated is called.

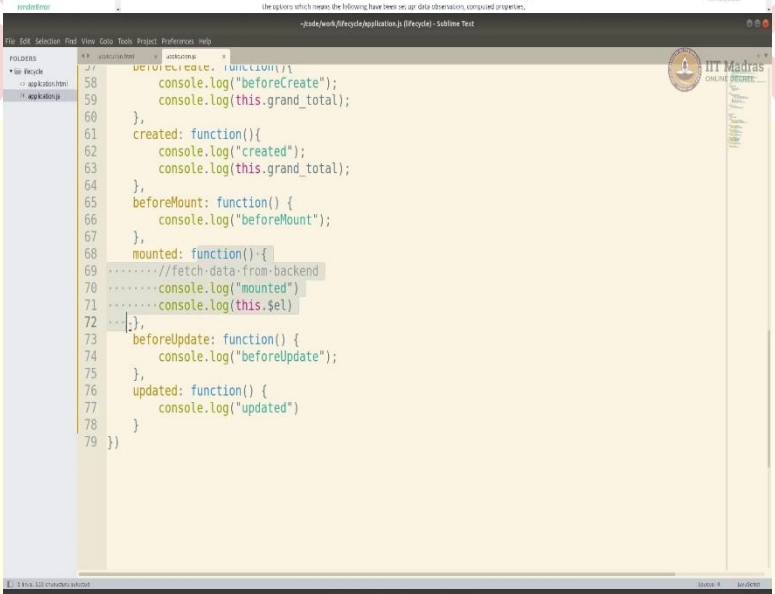
(Refer Slide Time: 08:27)

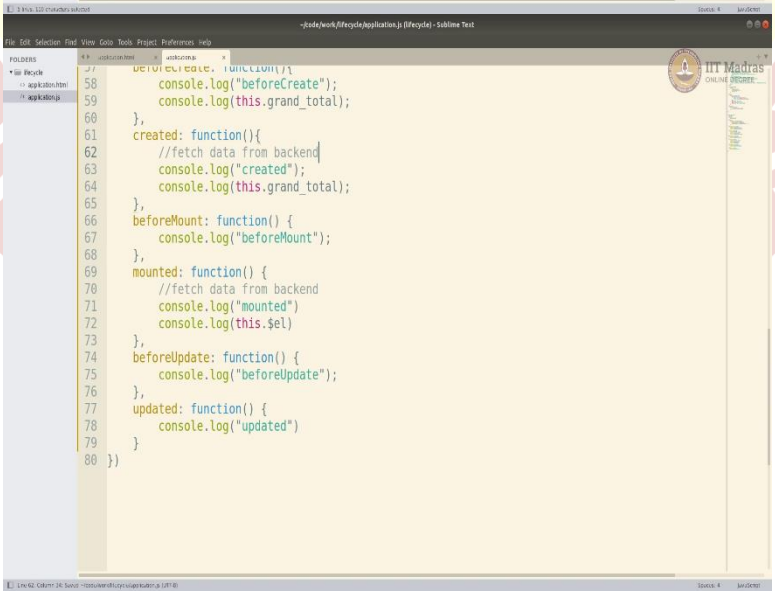




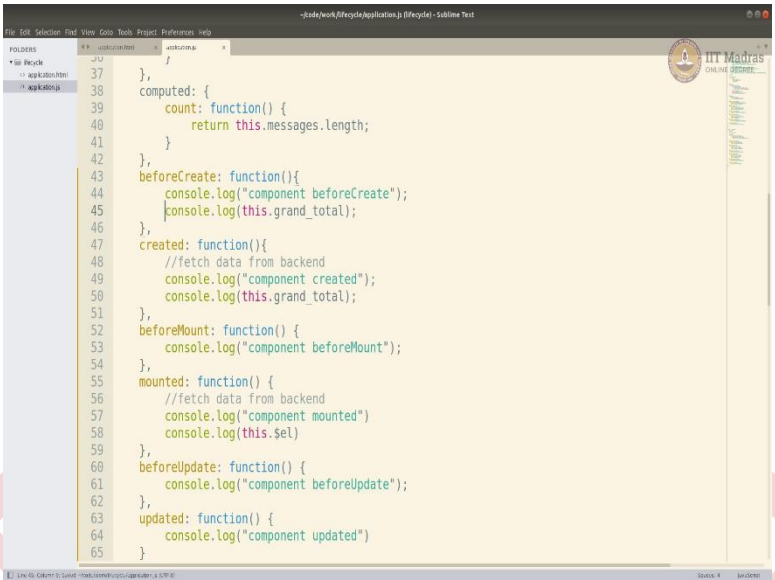


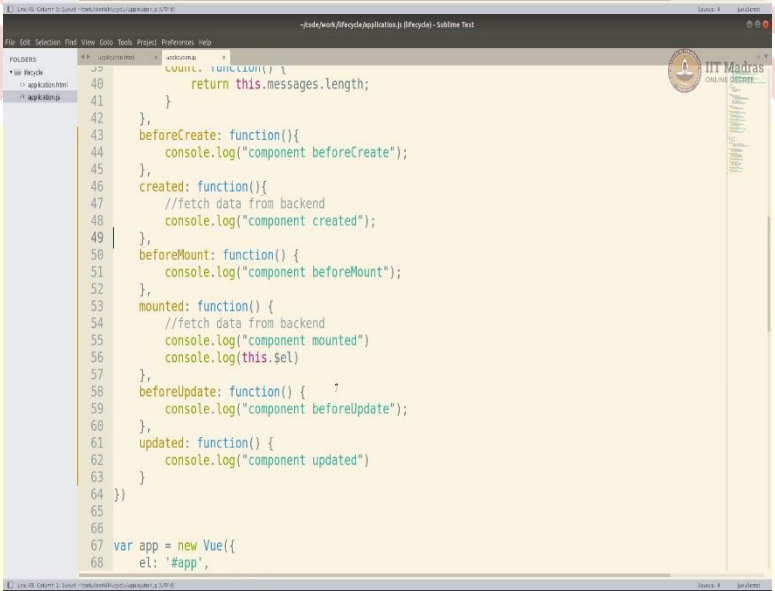

API — Vue.js - Chromium
Vue.js
Options / Lifecycle Hooks
beforeCreate
created

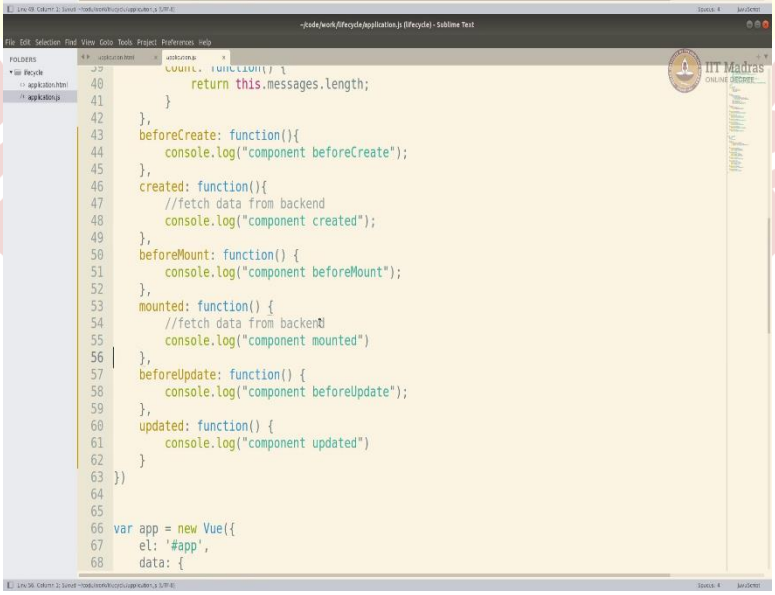

~/code/work/lifecycle/application.js (lifecycle) - Sublime Text
FOLDERS
lifecycle
application.html
application.js
beforeCreate: function(){
console.log("beforeCreate");
console.log(this.grand_total);
},
created: function(){
console.log("created");
console.log(this.grand_total);
},
beforeMount: function() {
console.log("beforeMount");
},
mounted: function() {
//fetch data from backend
console.log("mounted")
console.log(this.$el)
},
beforeUpdate: function() {
console.log("beforeUpdate");
},
updated: function() {
console.log("updated")
}
})

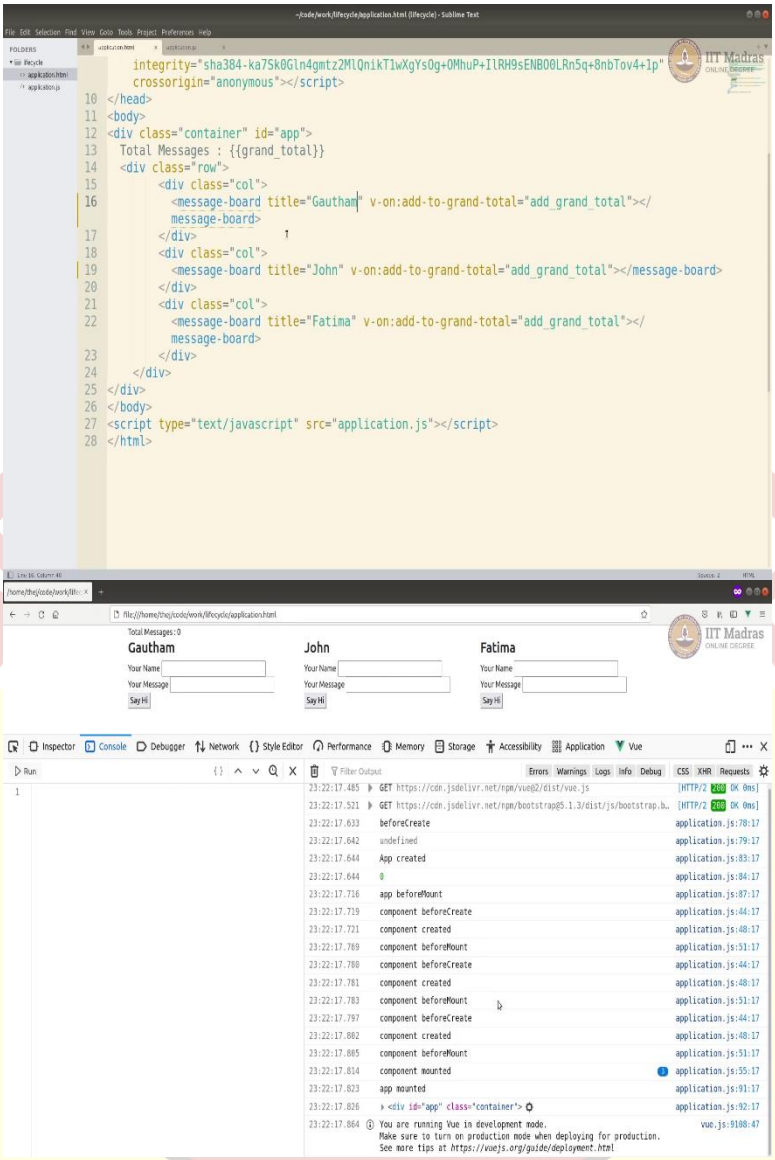

~/code/work/lifecycle/application.js (lifecycle) - Sublime Text
FOLDERS
lifecycle
application.html
application.js
beforeCreate: function(){
console.log("beforeCreate");
console.log(this.grand_total);
},
created: function(){
//fetch data from backend
console.log("created");
console.log(this.grand_total);
},
beforeMount: function() {
console.log("beforeMount");
},
mounted: function() {
//fetch data from backend
console.log("mounted")
console.log(this.$el)
},
beforeUpdate: function() {
console.log("beforeUpdate");
},
updated: function() {
console.log("updated")
}
})


lifecycle.png (1200×3039) - Chromium
Events & Lifecycle
beforeCreate
Init
injections & reactivity
created
Has
"el" option?
NO
when
vm.$mount(el)
is called
YES
Has
"template" option?
YES
NO
Compile template
into
render function *
Compile el's
outerHTML
as template *
beforeMount
~/code/work/lifecycle/application.js (lifecycle) - Sublime Text
FOLDERS
lifecycle
application.html
application.js
61 created: function(){
62 //fetch data from backend
63 console.log("App created");
64 console.log(this.grand_total);
65 },
66 beforeMount: function() {
67 console.log("app beforeMount");
68 },
69 mounted: function() {
70 //fetch data from backend
71 console.log("app mounted")
72 console.log(this.$el)
73 },
74 beforeUpdate: function() {
75 console.log("app beforeUpdate");
76 },
77 updated: function() {
78 console.log("app updated")
79 }
80 })
~/code/work/lifecycle/application.js - (lifecycle) - Sublime Text
47 created: function(){
48 //fetch data from backend
49 console.log("App created");
50 console.log(this.grand_total);
51 },
52 beforeMount: function() {
53 console.log("app beforeMount");
54 },
55 mounted: function() {
56 //fetch data from backend
57 console.log("app mounted")
58 console.log(this.$el)
59 },
60 beforeUpdate: function() {
61 console.log("app beforeUpdate");
62 },
63 updated: function() {
64 console.log("component updated")
65 }
66 })
69 var app = new Vue({
70 el: '#app',
71 data: {
72 grand_total: 0
73 },
74 methods: {


~/code/work/lifecycle/application.js (lifecycle) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
lifecycle
application.html
application.js
IIT Madras
ONLINE DEGREE
37 },
38 computed: {
39 count: function() {
40 return this.messages.length;
41 }
42 },
43 beforeCreate: function(){
44 console.log("component beforeCreate");
45 console.log(this.grand_total);
46 },
47 created: function(){
48 //fetch data from backend
49 console.log("component created");
50 console.log(this.grand_total);
51 },
52 beforeMount: function() {
53 console.log("component beforeMount");
54 },
55 mounted: function() {
56 //fetch data from backend
57 console.log("component mounted")
58 console.log(this.$el)
59 },
60 beforeUpdate: function() {
61 console.log("component beforeUpdate");
62 },
63 updated: function() {
64 console.log("component updated")
65 }

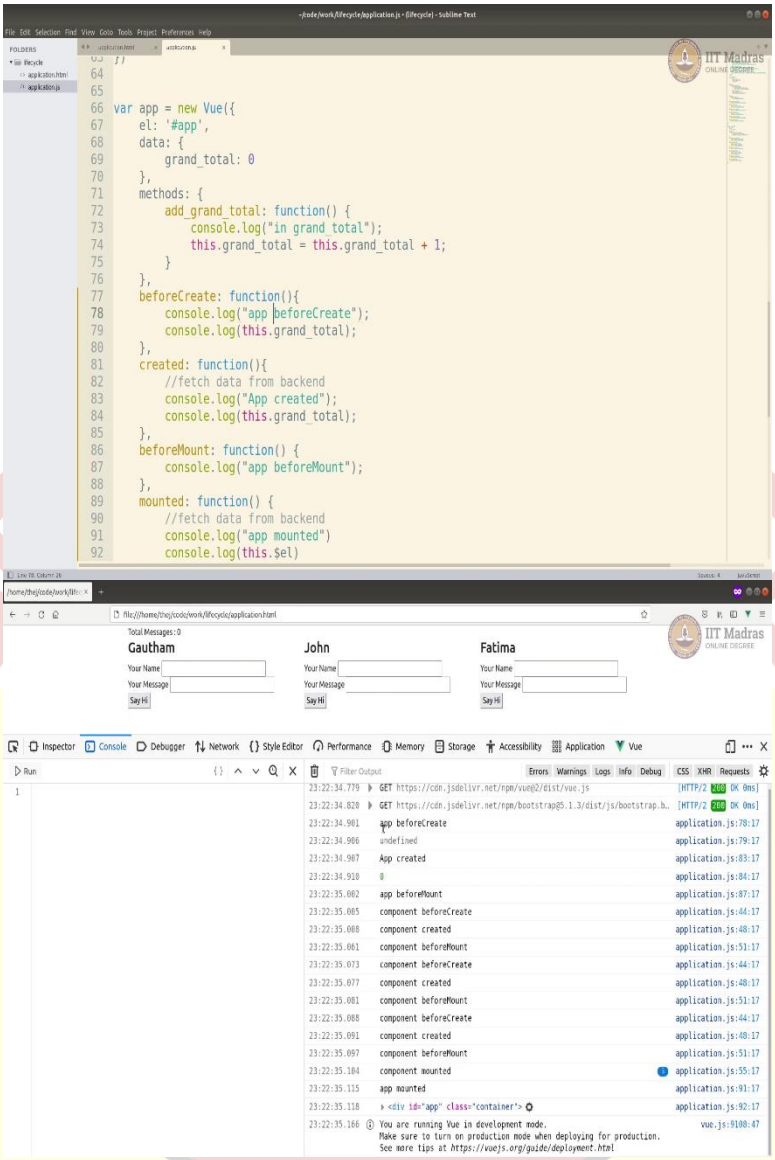

~/code/work/lifecycle/application.js (lifecycle) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
lifecycle
application.html
application.js
IIT Madras
ONLINE DEGREE
40 return this.messages.length;
41 }
42 },
43 beforeCreate: function(){
44 console.log("component beforeCreate");
45 },
46 created: function(){
47 //fetch data from backend
48 console.log("component created");
49 },
50 beforeMount: function() {
51 console.log("component beforeMount");
52 },
53 mounted: function() {
54 //fetch data from backend
55 console.log("component mounted")
56 console.log(this.$el)
57 },
58 beforeUpdate: function() {
59 console.log("component beforeUpdate");
60 },
61 updated: function() {
62 console.log("component updated")
63 }
64 })
65
66
67 var app = new Vue({
68 el: '#app',


~/code/work/lifecycle/application.js (lifecycle) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
lifecycle
application.html
application.js
IIT Madras
ONLINE DEGREE
40 return this.messages.length;
41 }
42 },
43 beforeCreate: function(){
44 console.log("component beforeCreate");
45 },
46 created: function(){
47 //fetch data from backend
48 console.log("component created");
49 },
50 beforeMount: function() {
51 console.log("component beforeMount");
52 },
53 mounted: function() {
54 //fetch data from backend
55 console.log("component mounted")
56 },
57 beforeUpdate: function() {
58 console.log("component beforeUpdate");
59 },
60 updated: function() {
61 console.log("component updated")
62 }
63 })
64
65
66 var app = new Vue({
67 el: '#app',
68 data: {


integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>
</head>
<body>

Total Messages : {{grand_total}}

<message-board title="Gautham" v-on:add-to-grand-total="add_grand_total"></message-board>
<message-board title="John" v-on:add-to-grand-total="add_grand_total"></message-board>
<message-board title="Fatima" v-on:add-to-grand-total="add_grand_total"></message-board>
</body>
<script type="text/javascript" src="application.js"></script>
</html>
Total Messages : 0
Gautham
John
Fatima
Your Name
Your Message
Say Hi
Inspector
Console
Debugger
Network
Style Editor
Performance
Memory
Storage
Accessibility
Application
Vue
Run
Filter Output
Errors
Warnings
Logs
Info
Debug
CSS
XHR
Requests
GET https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js
GET https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.b...
beforeCreate
undefined
App created
app beforeMount
component beforeCreate
component created
component beforeMount
component beforeCreate
component created
component beforeMount
component beforeCreate
component created
component beforeMount
component mounted
app mounted
You are running Vue in development mode.
Make sure to turn on production mode when deploying for production.
See more tips at https://vuejs.org/guide/deployment.html



Now let us look at the next set before destroy and destroy. I am not going to look at it, I will leave that to you to explore. Now where can you use destroy and before destroy. I think for example if you are using inside. If you are doing a single page application and if you hide certain things or move away certain things then before destroy and destroyed get gets called.

In before destroyed you still have access el property and all the other properties, so you can do things that you want to do before app or this component till it gets cleared. For example, you can switch off your event hooks or you can switch off a clear child component or clear watchers. You can turn off event listeners or event hooks, all of that you can do before destroy

And now the destroyed, this one, gets called at the end when the when the component or the app has been destroyed. At this point you do not have access to any of the data or el or any of those parameters. You usually use this to send an alert to the main app or other components

that this specific component or app has been destroyed and do something about it. Usually if you like one gets destroyed and the other component shows some message then you could use destroyed, so hook to do that. In before destroy you actually do all the shutdown things that you want to do.

There is also activated and deactivated event hooks or sorry lifecycle hooks rather, and they are not used a lot. You can learn about them by yourself. You can go to API site of UJS and they have all the information about all of the API's and event lifecycle hooks that you can look at and learn about them. You can go here, you can search for, for example, before, create.

Yeah, so you can see all the lifecycle object. Now there is one more thing to remember, all lifecycle hooks automatically have their own this context bound to the instance. So, you actually cannot write this arrow functions you have to write full fledged function. That is why we have written this full defined function and not arrow function.

It is important to remember, because sometimes you end up doing that and get stuck. Now there is also a debate as to where to fetch the data, should it fetch here or should we fetch it here? You can, it depends, actually and it is highly contested. It depends on your requirements and use cases where you want to fetch the data, but the most suggested one is to do it at a mounted thing, because everything is ready. And you know, it also useful when you are using server side rendering.

But you can also fetch create it before the user is shown the UI. But remember, most times, most people preferred to do fetch the fetching of the data and other things in the mounted lifecycle. webhook.

Another thing to remember is, if you call a sync await inside a mounted one, it would not block the rendering of the things because it is already mounted, so you can do that and reset it. So, you can also call async await without actually hindering by usability. And that is it, actually. So, you could you could actively use lifecycle hooks to do various things at various points in the life of a Vue app, and the same applies to view component as well.

Let me just add that, let me just add app here. Okay, I am just going to come make the copy of the same thing. Okay, I will add it to component, which is one of our component, message board. They also have similar life cycle, but here, I will just do comp, component instead of app. Going to remove this because this is not the data that is present here. You could actually

check for the data that is present in a component. This can be left, but I am just going to remove this for testing purposes. So, this all has happened, this all has component. So, you can see there is one app, and in our application, there are three components. So, it should do it for all of them. Let us clear that.

So, you can see that, before create an app created before create, yeah, so we have not given app here. Just to make it clear, and easy to understand I will just make that. So, you can see that app got created, and then app got beforemount, then the mounting starts to happen. As part of the mounting happening, you have all the components getting initialized and setting up and things like that, and mounted. And then once that is completed, the whole app gets mounted.

That is it. If you remove the component, then it will go through its own cycle of destroyed and before destroy. And then if you remove the whole app at that point app will also get the same hook calls. That is it actually. Thank you for watching. Hope you will end up using some of these hooks to update your components or app at specific times in the lifecycle of the Vue app or component. Thank you. Bye.