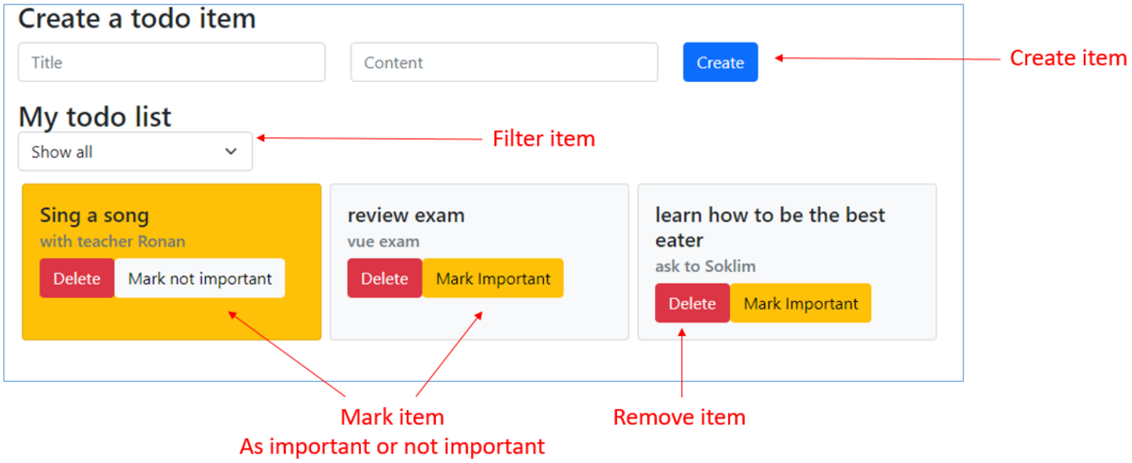
VUE C4-S7 –*TODO list*

We want to create a TODO app:

- ✓ Create items
- ✓ Mask item as important
- ✓ Remove items
- ✓ Filter items



Create a todo item

My todo list

Sing a song
with teacher Ronan

review exam
vue exam

learn how to be the best eater
ask to Soklim

Create item

Filter item

Mark item
As important or not important

Remove item

#1 – CRUD on client only

For now, let’s add, remove, and update the data **locally**, on the client, using our **VueJS data**.

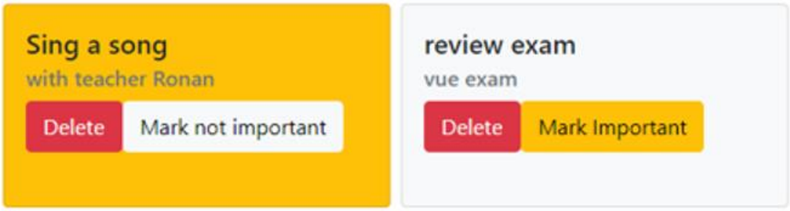
A TODO item is composed of the following fields:

Attribute	Type
Id	Number
Title	String
Content	String
important	boolean

You need to **manage a list of items in the VueJS data**

- ✓ When clicking on **create button**, add a new item to the list :
  - content = the input value
  - title = the input value
  - id = generated (for now it's fake)
  - important = false (by default)

- ✓ **Display the list of items** as follows :
  - If item is important it should be displayed in YELLOW



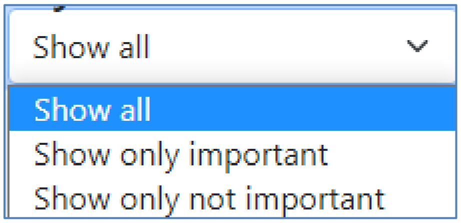
Item is important *Item is not important*

- ✓ When clicking on "**Mark Important" / "Mark not Important**" it should change the status of the item

Note. : To style the card as important, you can use the following CSS class: ***important***

- ✓ When clicking on **Delete** it should change the right item (corresponding to the right ID in the list of items)

- ✓ Now **manage the filters** : to show only the important or not important, or to show all :



Small tip: Can you use a computed property to filter the list?

#2 – Create an REST API on back end

The first part is on the BACK side:

- ✓ Your back end API should manage the below REQUEST
- ✓ Your back end should store the items in a SQL database (for example)

You should also test your API using THUNDER client.

URL	<YOUR ROOT API> /api/todos/
METHOD	GET
INPUT	Nothing
OUTPUT	The list of todos.

URL	<YOUR ROOT API> /api/todos
METHOD	POST

INPUT	{
	title: 'New todo',
	content: 'This is my first todo'
	}
	
	Id should not be sent
	Important is optional

OUTPUT	The updated list of todos. You can use it to update the list in your page.

URL	<YOUR ROOT API> /api/todos/id
METHOD	DELETE
INPUT	Nothing
OUTPUT	The updated list of todos. You can use it to update the list in your page.

URL	<YOUR ROOT API> /api/todos/id
METHOD	PUT

INPUT	{
	title: 'Do my homework',
	content: 'Any text you want, it will update all'
	important: true
	}

OUTPUT	The updated list of todos. You can use it to update the list in your page.

#3 – Connect front and back ends

Now, it’s time to connect the **front end** with the **back end** (why not?)

Display item   when front end is loaded:

- ✓ When front end application is loaded (i.e. “mounted”) you need **to fetch the list of TODO from the back end**
- ✓ Use AXIOS to send requests to the back end

Send request when Adding, removing, updating an item

- ✓ Complete the existing code to send requests to the back end whenever you add, remove, update the items
- ✓ Make sure the VueJS data is **always updated** with the new list provide by the back end