

## VUE

Exercice : component communication (props) plus détaillé

Dans **UnAmi.vue** :

°Mettre en commentaire tout le tableau **props**, on va créer un objet **props** plutôt.

°Dans l'objet **props**, on va rajouter plusieurs objets (1 objet pour chaque **props** techniquement) :

```
°leNom : {}
°lePhone : {}
°leMail : {}
°premium : {}
```

°Dans chacun de ces objets on va avoir une propriété **type:String** (toujours Majuscule pour le type) et propriété **required** à true (sauf premium sa propriété **required** est à false)

° pour l'objet de la **props** premium, on rajoute une propriété **default** à '0', et une propriété **validator**

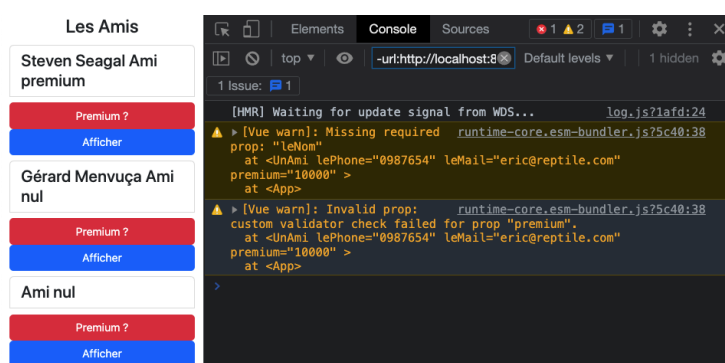
° à la propriété **validator** on lui attribue une fonction qui a une variable **value** en paramètre et qui **retourne** : cette condition `value==='1' || value==='0'`

Dans **App.vue** :

° Rajouter un composant `<un-ami>`, et nous allons créer des erreurs en injectant des props erronées par exemple :

° On oublie de renseigner la **props** leNom, et dans la **props** premium on va passer une mauvaise valeur « 1000 » par exemple.

°Ensuite dans le navigateur on va explorer la console pour voir les erreurs qui sont générées par cette nouvelle approche des **props** (plus sécurisé)



Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<b>Mathieu Mithridate</b>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

[GitHub](#)

Pour résumer avec l'approche objet des props :

```
props:{
  premium:{
    type:String,
    required:false,
    default:'0',
    validator: function(value){return value==='1' || value==='0'}
  }
}
```

Nous pouvons gérer le **type** des données qui seront injectées dans le composant enfant, les rendre obligatoire ou non avec **required**, mettre une valeur par **default**(on peut utiliser également une fonction dans default), enfin les **validator** permette d'avoir des contrôles très précis et customisé avec des conditions spécifiques (ici par exemple le **validator** n'autorise que la chaîne de caractère « 1 » ou « 0 »).

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<b><u>Mathieu Mithridate</u></b>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

### Props + v-for : composants dynamiques

A ce stade nous savons faire communiquer (injecter) des données d'un composant vers son composant enfant... Mais nous connaissons aussi la directive v-for... 😊

Afin de rendre notre composant encore plus dynamique et plus facilement réutilisable, dans le sens ou cela devient répétitif d'injecter une par une les props depuis le composant parent comme ceci :

```
<un-ami leNom="Gérard Menvuca" lePhone="0987654" leMail="gerard@menvuca.com"premium="0"></un-ami>
```

Dans l'exercice suivant nous allons gérer toutes les données dans les data de App.vue (composant parent) et optimiser leurs injections dans les **props** du composant enfant (UnAmi.vue).

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<b><u>Mathieu Mithridate</u></b>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

Exercice : component communication (props + v-for) Composants dynamiques

**Dans UnAmi.vue :**

- ° on va modifier l'objet de la **props** premium (le **type** sera booléen, le **required** et **default** seront à false)
- on peut mettre en commentaire le **validator** (plus besoin si c'est de type booléen)
- ° Au niveau des **data** on peut mettre en commentaire l'objet unPote
- ° Au niveau des methods, mettre en commentaire (ou supprimer) la partie IF ELSE, on va faire plus simple : on assigne premiumData à son inverse.
- ° Dans le template au niveau du <h2> modifier la condition ternaire : on vérifie juste si premiumData est définie (à voir on l'a ptet déjà fait 😊 cf. exo d'avant)

**Dans App.vue :**

- ° Mettre en commentaire (ou supprimer) les composants <un-ami> (on va en utiliser 1 seul).
- ° Sur un composant <un-ami> :
  - ° utiliser une directive v-for : avec une variable temporaire unAmi, parcourir le tableau lesAmis.
  - ° Ensuite on va faire des v-bind de :
    - key pour le lier à l'id de unAmi
    - leNom pour le lier à name de unAmi
    - lePhone pour le lier à phone de unAmi
    - leMail pour le lier à email de unAmi
    - premium pour lui assigner true

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<u>Mathieu Mithridate</u>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

### Les Amis

COCO L ASTICOT (Ami premium)

Premium ?

Masquer

01234 5678 991

coco@lasticot.com

Janine DeLavega (Ami nul)

Premium ?

Afficher

[GitHub](#)

### \$emit() (communication composant enfant vers composant parent)

Nous avons vu la notion de **props** dans VUEJS, cela nous permettait de transmettre des informations (des données) depuis un composant parent vers son composant enfant, nous avons également constaté que pour établir cette communication de données entre les composants, VUE se base sur un flux de données UNIDIRECTIONNEL (**unidirectional dataflow**), c'est-à-dire que pour des raisons d'optimisation / sécurité / etc..., on ne pouvait pas directement faire des modifications d'une **props** (une données injectée) dans le composant enfant (pour résoudre ce problème nous avons assigné une **data** à la **props** que l'on voudrait modifier).

Rappel : dans UnAmi .vue, pour pouvoir modifier la **props** premium dans ce composant on était passé par une **data** à laquelle on assigne la props.

```
export default {
  props: {
    premium: {
      type: Boolean,
      required: false,
      default: false
    }
  },
  data() {
    return {
      premiumData: this.premium
    }
  }
}
```

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<b><u>Mathieu Mithridate</u></b>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

Dans VUE c'est une approche au cœur du système de composant (et leur imbrication), sur des applications SPA (Single Page Application).

🤔 Quand on y pense :

Généralement c'est le root composant (racine) soit App.vue, qui va héberger / faire appel à des composants enfant (UnAmi.vue).

Mais quand on commence à complexifier les applications (gérant beaucoup de données, de fonctions, etc...), nous allons vouloir communiquer dans le sens inverse, que le composant enfant communique des informations au composant parent. (C'est du **unidirectional dataflow** mais dans l'autre sens).

Pour faire en sorte qu'un composant enfant envoie une information vers le composant parent, le composant enfant va devoir émettre un événement (le changement d'une variable dans par exemple), on peut faire le parallèle où en JS on capte un événement ('click', 'keyup', 'mouseout', ...) qui contient des informations (les coordonnées du click, la lettre tapée au clavier, etc..), ici c'est le même principe mais on crée des événements « custom » qui correspondent aux besoins précis des données de l'application.

Dans le composant enfant cette communication se fait via une fonction de VUE :

**\$emit('leNomDuEvent')**

Et dans le composant **parent** on va réagir à cet événement :

**@leNomdeDuEvent** = " du code JS / une variable / une fonction / etc...".

Dans l'exercice suivant nous allons faire en sorte que le composant enfant émette un événement très simple (qui ne contient pas d'informations (pour le moment)) et dans le composant parent on va réagir à cet événement en exécutant un console.log.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<b><u>Mathieu Mithridate</u></b>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

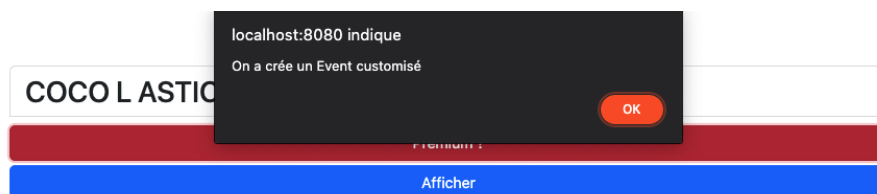
Exercice : component communication (\$emit()) test :réagir à un évènement custom

### Dans unAmi.vue :

° Au niveau des **methods**, modifier la méthode afficherPremium, mettre en commentaire la manipulation de la variable premiumData, à la place on va utiliser this.\$emit() avec une chaîne de caractère en paramètre « mon-event-premium » (c'est le nom de l'évènement que le composant enfant va ... émettre).

### Dans App.vue :

- ° Au niveau des **data**, pour chaque objet du tableau lesAmis, rajouter une propriété premium (à true ou false).
- ° Au niveau des **methods**, une méthode afficherStatusPremium dans laquelle on fait juste un console log (message au choix).
- ° Au niveau du template sur le composant **<un-ami>** :
  - ° modifier le **v-bind** de premium pour le lier à la propriété premium de unAmi.
  - ° mettre en place un **v-on** qui réagit à notre évènement « mon-event-premium » et appelle la méthode afficherStatusPremium



[GitHub](#)

On vient d'utiliser la fonction \$emit(), en passant 1 paramètre : le nom de l'évènement ...

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<b><u>Mathieu Mithridate</u></b>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

Exercice : component communication (\$emit()) test : réagir à un évènement custom + données

### Dans UnAmi.vue :

- ° ajouter une props id (de type String, et required)
- ° techniquement on utilise plus premiumData (on peut mettre en commentaire dans les **data**), et dans la condition du <h2> (dans le template) utiliser premium (la **props** directement)
- ° Dans les **methods**, ajouter cette props id en second paramètre de \$emit()

Le composant enfant émet un event qui contient un id, on va le capter dans le composant parent

### Dans App.vue :

- ° Dans le template sur le composant <un-ami>, on peut ajouter un v-bind de id pour le lier à unAmi.id
- ° modifier la methods afficherStatusPremium pour qu'elle prenne une variable en paramètre (lIdDansUnAmi).
- ° Dans afficherStatusPremium :
  - ° créer une variable unAmiIdentified, lui assigner le tableau lesAmis (**data**), sur lesAmis utiliser la fonction. find(unTruc => unTruc.id === lIdDansUnAmi)
  - ° à la propriété premium de unAmiIdentified, lui assigner son inverse

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<b><u>Mathieu Mithridate</u></b>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		



## VUE

### Les Amis

Captain Hadock (Ami nul)

Premium ?

Afficher

Steven Seagal (Ami premium)

Premium ?

Afficher

[GitHub](#)

[\\$.emit\(\)](#) (plus en détails)

Nous venons de voir une première manière d'utiliser la fonction emit(), mais pareillement à props, on peut aller plus loin dans la gestion des événements avec \$.emit() en passant par un tableau et même un objet, nous allons voir la version objet pour rajouter dans notre événement, une fonction qui va vérifier si l'id est bien transmit.

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<b><u>Mathieu Mithridate</u></b>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

Exercice : component communication (\$emit()) en mode objet

Dans UnAmi.vue :

- ° Dans l'export default, après les props, déclarer emits , un objet { }
- ° Dans cet objet on a une chaîne de caractère ('mon-event-premium') à laquelle on assigne une fonction (anonyme ou fléchée) qui prend id en paramètre.
- ° Dans cette fonction on va faire ceci :  
SI id est définie ALORS on return true **SINON** on fait un console.warn('MSG au choix') et on return false
- ° Pour tester (voir si le warning se déclenche), dans les methods retirer le paramètre this.id dans la fonction \$emit
- ° Pensez à le remettre par la suite.

```

  ▲ ▶ ON A PAS DE ID UnAmi.vue?47e4:53
  ▲ ▶ [Vue warn]: Invalid event arguments: event validation failed for event "mon- runtime-core.esm-bundler.js?5c40:38
    event-premium".
  ▲ ▶ [Vue warn]: Unhandled error during execution of component event handler runtime-core.esm-bundler.js?5c40:38
    at <UnAmi key="kimonoSurUnFrigo" id="kimonoSurUnFrigo" leNom="Steven Seagal" ... >
    at <App>
  ▲ ▶ [Vue warn]: Unhandled error during execution of native event handler runtime-core.esm-bundler.js?5c40:38
    at <UnAmi key="kimonoSurUnFrigo" id="kimonoSurUnFrigo" leNom="Steven Seagal" ... >
    at <App>

```

Cette approche permet de renforcer la sécurité des données quand elles communiquent entre composant en passant par des événements.

[GitHub](#)

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<b><u>Mathieu Mithridate</u></b>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

### Schéma communication composants



Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<b>Mathieu Mithridate</b>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		

## VUE

### Exercice : component communication (\$emit()) + nouveau composant

Nous allons rajouter un composant permettant d'ajouter un ami

#### Dans le dossier components :

° Créer un nouveau fichier « NewAmi.vue »

#### Dans main.js :

° importer le composant NewAmi, puis déclarer ce composant sous le nom « new-ami » dans l'app

#### Dans NewAmi.vue :

° Dans le **template** : 1 formulaire avec 3 inputs (pour ajouter un nom, un tél, un mail), et un bouton

° Pour la partie **script** juste la balise <script>: avec l'export default pour le moment

° Dans le **script**, dans export default, rajouter un tableau **emits** qui contient une chaîne de caractère « add-ami » (c'est le nom de l'événement)

° Rajouter des **data** (des string vide) enteredName, enteredPhone, enteredMail

° Rajouter une **methods** submitData dans laquelle :

° on utilise la fonction **\$emit** en lui passant en paramètre « add-ami » (le nom de l'événement) puis les data enteredName/Phone/Mail

° Dans le **template** :

placer un v-model sur chaque input pour les relier aux **data**

sur la balise <form> placer un v-on qui écoute l'événement « submit » (ajouter le **modifiers** prevent)

et qui déclenche notre **methods** submitData

#### Dans App.vue :

° sur un composant <new-ami> placer un v-on qui écoute notre événement add-ami et qui déclenche une méthode ajouterAmi

° Rajouter une méthode ajouterAmi qui prend en param name, phone et email.

° Dans cette fonction ajouterAmi, créer une variable newAmiContact, lui assigner un objet.

° Dans cet **objet** on a :

- une propriété id, lui assigner new Date().toISOString(),
- des propriétés name, phone et email, leur assigner name, phone et email,
- une propriété premium, lui assigner false

° Dans la data lesAmis, faire un push() de newAmiContact

[GitHub](#)

Auteur(s)	Relu, validé et visé par :	Date de création :	Date dernière MAJ :
<b>Mathieu Mithridate</b>	Jérôme CHRETIENNE : Resp. Secteur Tertiaire & Numérique	25/01/2021	
	Florence CALMETTES : Coordinatrice Filière Syst. & Réseaux	Toute reproduction, représentation, diffusion ou rediffusion, totale ou partielle, de ce document ou de son contenu par quelque procédé que ce soit est interdite sans l'autorisation expresse, écrite et préalable de l'ADRAR.	
	Marc CECCALDI Coordinateur Filière Développement		