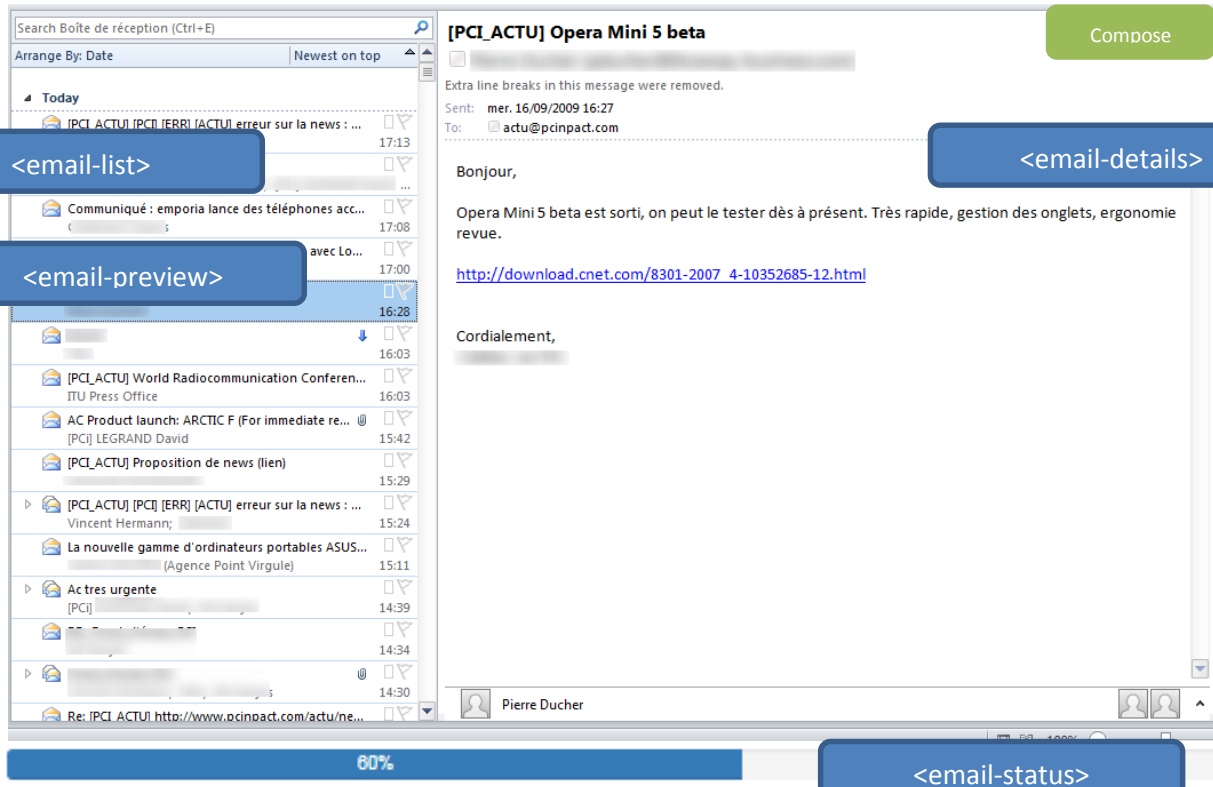




Appsus Emails

Build an email app.

<email-app>



<email-list>

<email-preview>

<email-details>

<email-status>

- Data: emails: {subject: "", body: "", isRead: false}
- <email-app>
 - Has an emails array (hardcoded)
 - Has a selectedEmail property
 - Pass emails down to <email-list> as a prop
 - Pass selectedEmail down to <email-details> as a prop
 - When email is selected on the list, the <email-details> should display that email
 - At the beginning first email should be automatically selected
- <email-list> renders a list of <email-preview> pass down an email prop
 - <email-preview>
 - Has an email prop
 - Renders the subject
 - Gives visual indication for read/unread (i.e.: bold/unbold ; closed or open envelop)
- <email-details>
 - Has an email prop



- show the entire email
- allow deleting an email by \$emit an event to parent
- <email-status>
 - Renders how many read from the emails, render as a <progress-bar> which is a general component you build
- <email-filter>
 - Show a textbox connected with v-model="filter.txt"
 - Show radio buttons: All, Read, Unread with v-model="filter.emailStatus"
 - When user change the radio button or enter text and click enter we emit:
 - \$emit('filterEmails', filter)
 - This event should be emitted up all the way to email-app
 - In the email-app make the emails a computed property
- <email-compose>
 - Has a form with subject and body
 - When send – emit up to parent
 - Email is added to list
 - Yes, we are only supporting selfi-emails for now (-: