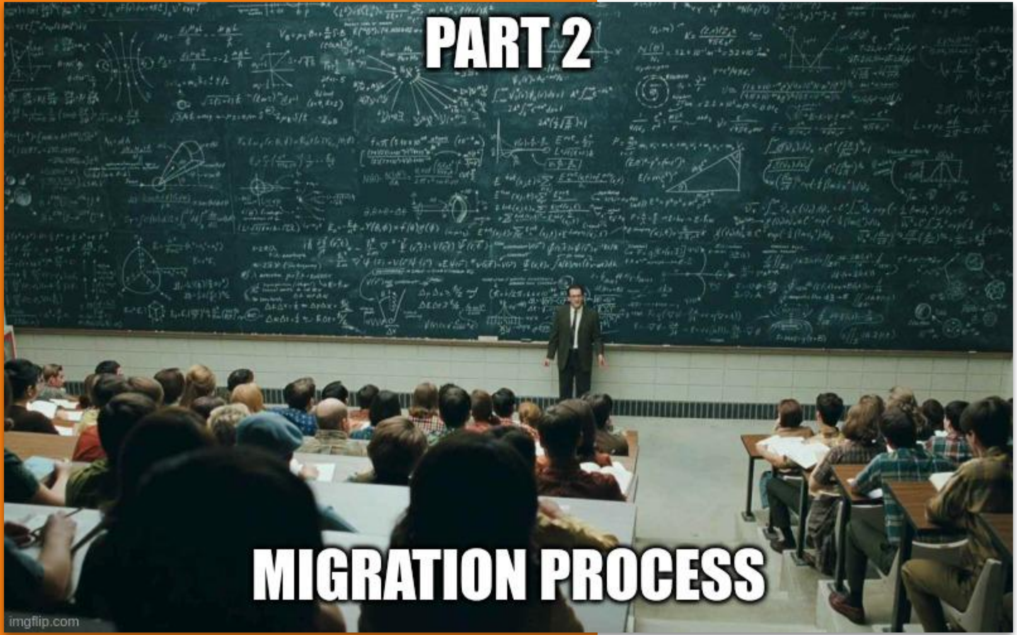
Moodle 4.0 course creation

Moodle Moot It 2021
02 12 2019

- Course navigation
- Activity cards design
- Collapsible sections
- Course index
- Blocks drawer
- Drag&Drop
- Add sections
- Move modal
- Completion icons
- Accessibility

Demo time!

Because nothing can go wrong with an old-fashioned live demo



- Courseformat subsystem
- Output classes
- Format renderer
- Base class

Demo time!

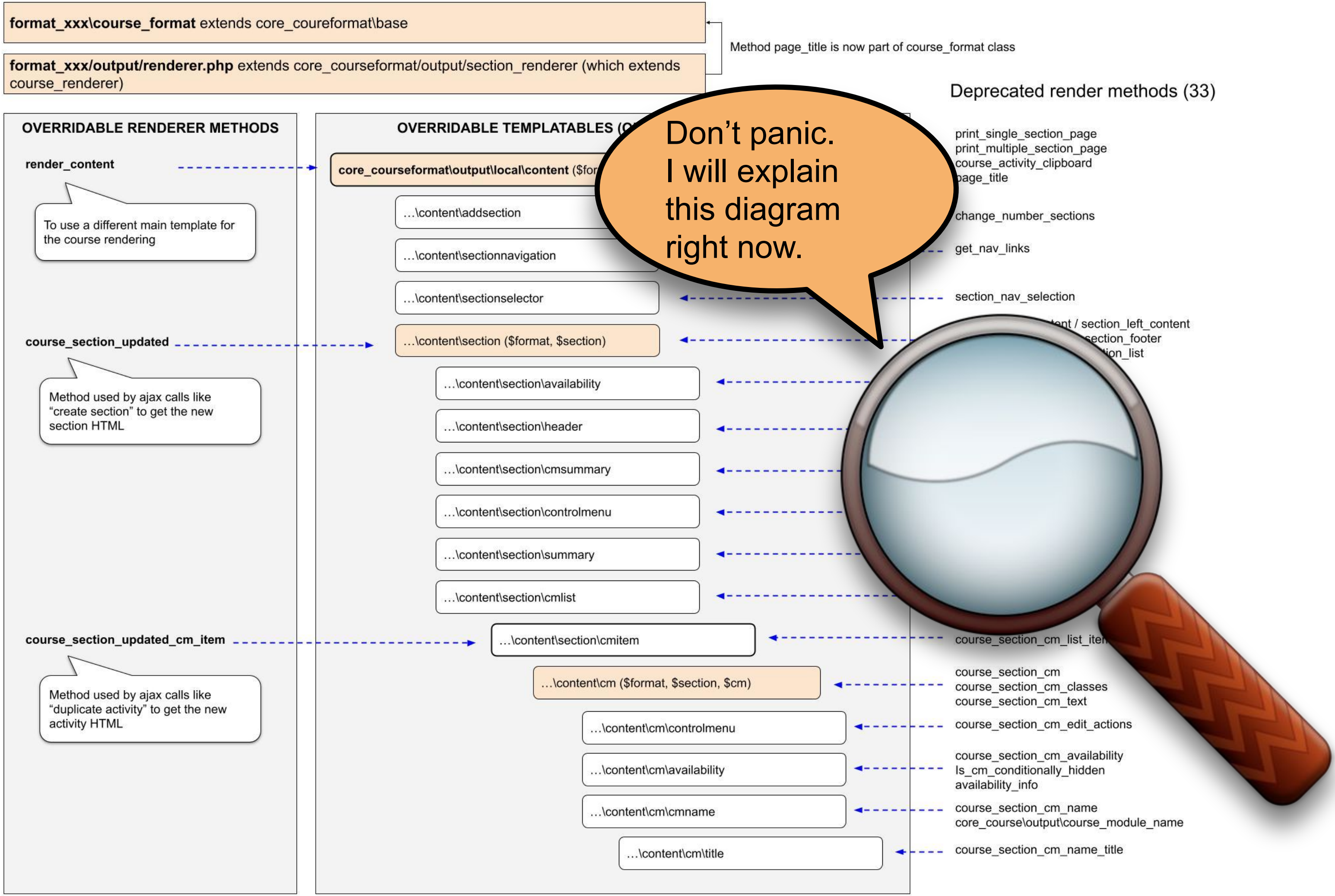
What happens when you use a Moodle 3.x course format in Moodle 4.0

core_courseformat subsystem

- The course/format folder is now a subsystem containing:
    - course formattings and rendering methods
    - course tests
    - course format API

Course format output classes

- Added **3 main renderer methods** (for full course, section and activity)
- The format **renderer class is now mandatory**
- 17 new output classes + mustache templates
- 33 deprecated methods from course_renderer and section_renderer
  (aka. format_section_renderer_base in Moodle 3.11)
- 13 new methods in core_courseformat\base class
  (aka. format_base in Moodle 3.11)



format_xxx\course_format extends core_coureformat\base

format_xxx/output/renderer.php extends core_courseformat/output/section_renderer (which extends course_renderer)

Method page_title is now part of course_format class

OVERRIDABLE RENDERER METHODS

render_content

To use a different main template for the course rendering

course_section_updated

Method used by ajax calls like “create section” to get the new section HTML

course_section_updated_cm_item

Method used by ajax calls like “duplicate activity” to get the new activity HTML

OVERRIDABLE TEMPLATABLES (O

core_courseformat\output\local\content ($for

…\content\addsection

…\content\sectionnavigation

…\content\sectionselector

…\content\section ($format, $section)

…\content\section\availability

…\content\section\header

…\content\section\cmsummary

…\content\section\controlmenu

…\content\section\summary

…\content\section\cmlist

…\content\section\cmitem

…\content\cm ($format, $section, $cm)

…\content\cm\controlmenu

…\content\cm\availability

…\content\cm\cmname

…\content\cm\title

Deprecated render methods (33)

print_single_section_page
print_multiple_section_page
course_activity_clipboard
page_title

change_number_sections

get_nav_links

section_nav_selection

tent / section_left_content
section_footer
tion_list

course_section_cm_list_item

course_section_cm
course_section_cm_classes
course_section_cm_text

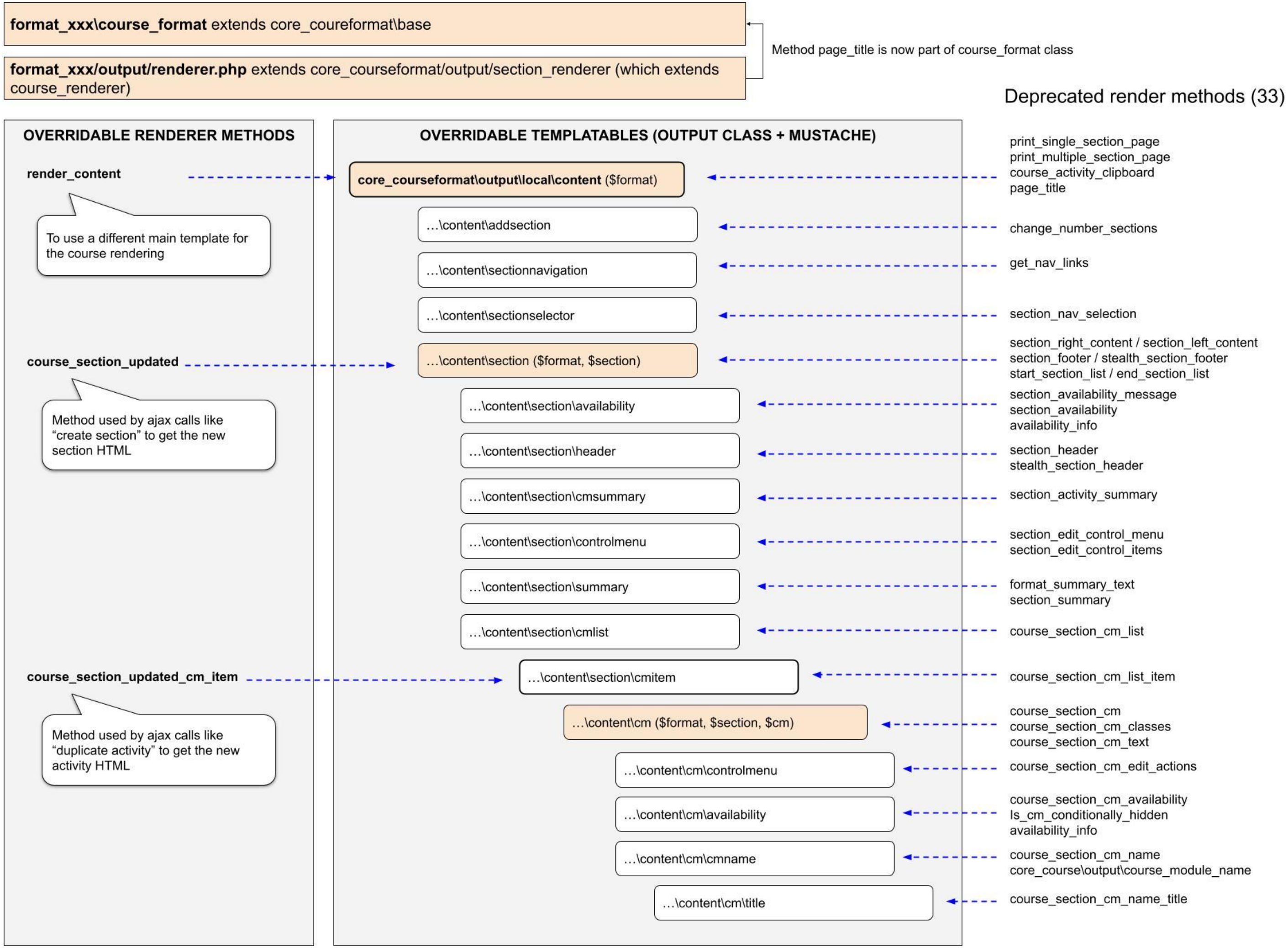
course_section_cm_edit_actions

course_section_cm_availability
Is_cm_conditionally_hidden
availability_info

course_section_cm_name
core_course\output\course_module_name

course_section_cm_name_title

Don’t panic.
I will explain
this diagram
right now.



format_xxx\course_format extends core_coureformat\base

format_xxx/output/renderer.php extends core_courseformat/output/section_renderer (which extends course_renderer)

Method page_title is now part of course_format class

Deprecated render methods (33)

OVERRIDABLE RENDERER METHODS

render_content

To use a different main template for the course rendering

course_section_updated

Method used by ajax calls like “create section” to get the new section HTML

course_section_updated_cm_item

Method used by ajax calls like “duplicate activity” to get the new activity HTML

OVERRIDABLE TEMPLATABLES (OUTPUT CLASS + MUSTACHE)

core_courseformat\output\local\content ($format)

…\content\addsection

…\content\sectionnavigation

…\content\sectionselector

…\content\section ($format, $section)

…\content\section\availability

…\content\section\header

…\content\section\cmsummary

…\content\section\controlmenu

…\content\section\summary

…\content\section\cmlist

…\content\section\cmitem

…\content\cm ($format, $section, $cm)

…\content\cm\controlmenu

…\content\cm\availability

…\content\cm\cmname

…\content\cm\title

print_single_section_page
print_multiple_section_page
course_activity_clipboard
page_title

change_number_sections

get_nav_links

section_nav_selection

section_right_content / section_left_content
section_footer / stealth_section_footer
start_section_list / end_section_list

section_availability_message
section_availability
availability_info

section_header
stealth_section_header

section_activity_summary

section_edit_control_menu
section_edit_control_items

format_summary_text
section_summary

course_section_cm_list

course_section_cm_list_item

course_section_cm
course_section_cm_classes
course_section_cm_text

course_section_cm_edit_actions

course_section_cm_availability
Is_cm_conditionally_hidden
availability_info

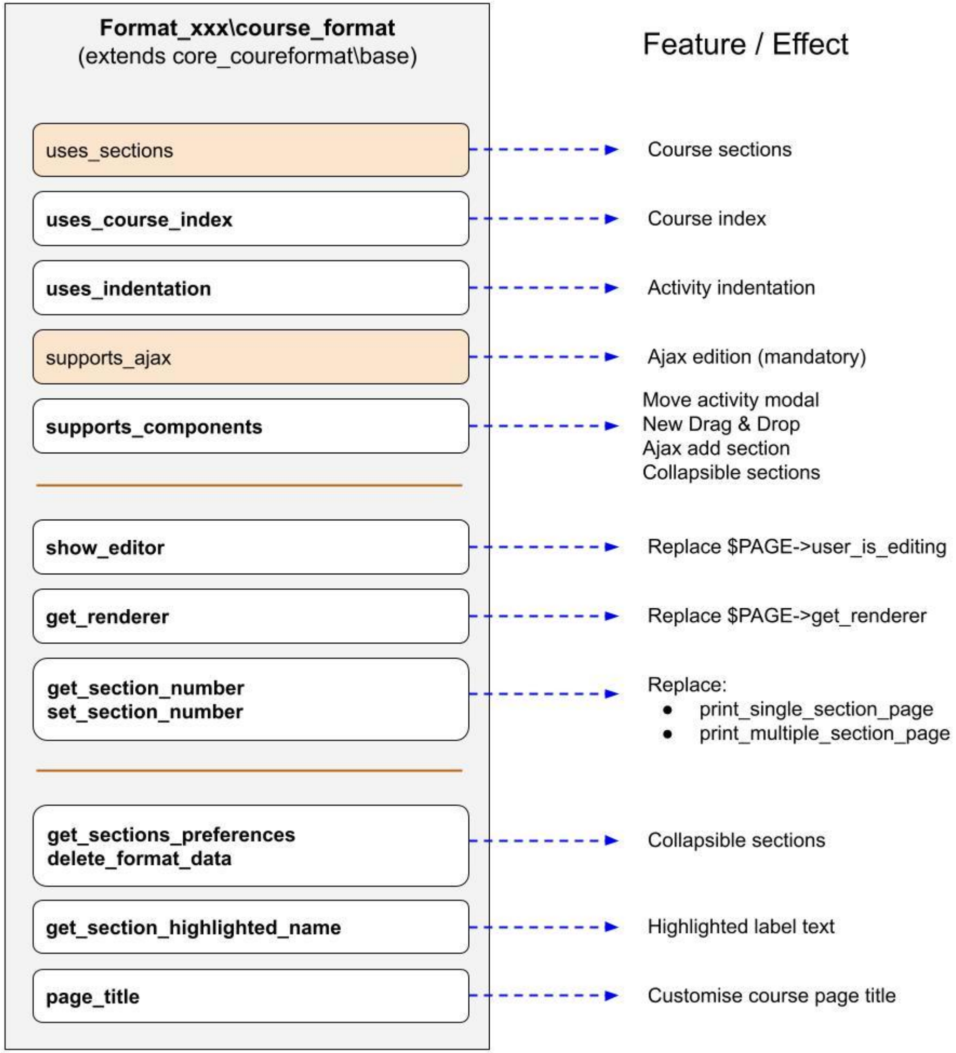
course_section_cm_name
core_course\output\course_module_name

course_section_cm_name_title

4.0 new features

Course index	Opt-in
Indentation (disabled in the standard formats)	Opt-out
New drag&drop	Opt-in
Move section/activity modal	Opt-in
Collapsible sections	Opt-out (overridable)
Activity card display	Opt-out (overridable)

4.0 Format base class methods



Format_xxx\course_format (extends core_coureformat\base)	Feature / Effect
uses_sections	Course sections
uses_course_index	Course index
uses_indentation	Activity indentation
supports_ajax	Ajax edition (mandatory)
supports_components	Move activity modal New Drag & Drop Ajax add section Collapsible sections
show_editor	Replace $PAGE->user_is_editing
get_renderer	Replace $PAGE->get_renderer
get_section_number set_section_number	Replace: ● print_single_section_page ● print_multiple_section_page
get_sections_preferences delete_format_data	Collapsible sections
get_section_highlighted_name	Highlighted label text
page_title	Customise course page title

- Reactive state
- State actions
- Components
- Mutations

Demo time!

“Pay no attention to that man behind the curtain”
- L. Frank Baum (The Wizard of Oz)

Reactive state principles

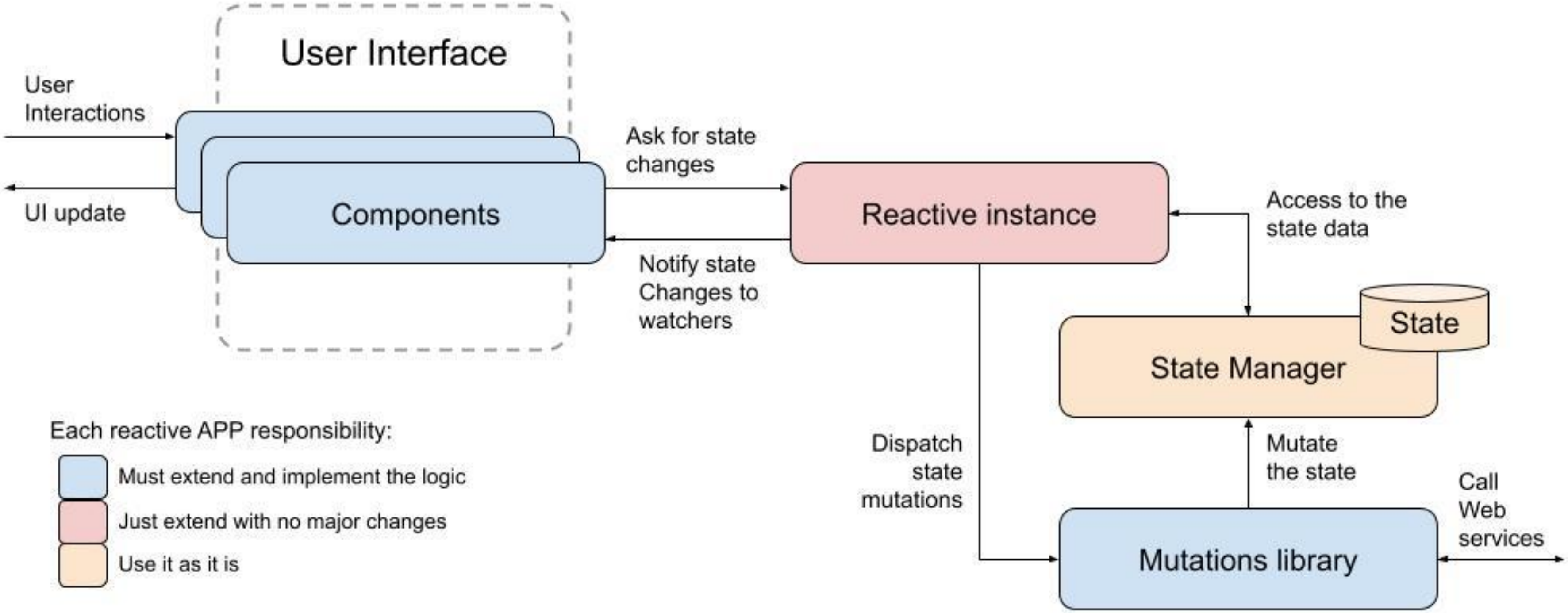
1. Each reactive element is controlled by its own **component** class
2. The UI is and **updated representation of a state data**
3. All user interactions can alter the state data **but never the HTML**
4. The new services are designed to alter the current state data, not the interface.

Module: core/reactive

Provides several classes to create small reactive states in Moodle.

- **Reactive**: each reactive application is represented by an instance of this class.
- **BaseComponent**: all reactive components extends this class
- **DragDrop**: a class to convert components into draggable and/or dropzones.
- **M.cfg.reactive**: object to debug your reactive application from the javascript console

Reactive workflow



User Interface

User Interactions

UI update

Components

Ask for state changes

Notify state Changes to watchers

Reactive instance

Access to the state data

State

State Manager

Dispatch state mutations

Mutate the state

Call Web services

Mutations library

Each reactive APP responsibility:

- 🟦 Must extend and implement the logic
- 🟥 Just extend with no major changes
- 🟧 Use it as it is

Thanks for your attention!

moodle

Empowering educators to improve our world