

# Day 2

# Technical Training

Odoo JavaScript Framework

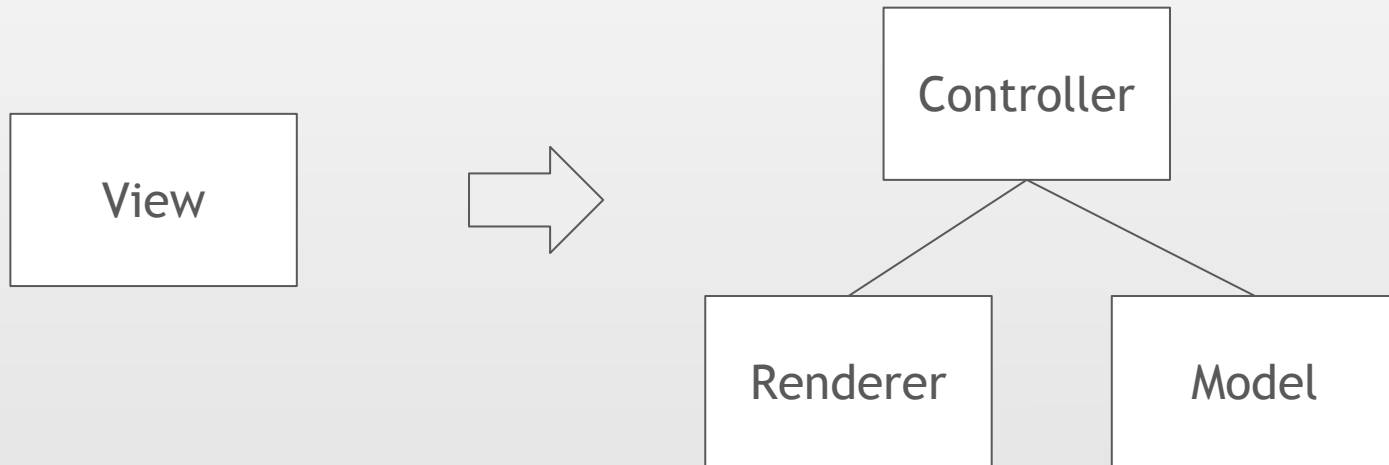
Géry Debongnie (ged)  
RD Framework Team

**Welcome back!**

Topic of the day:

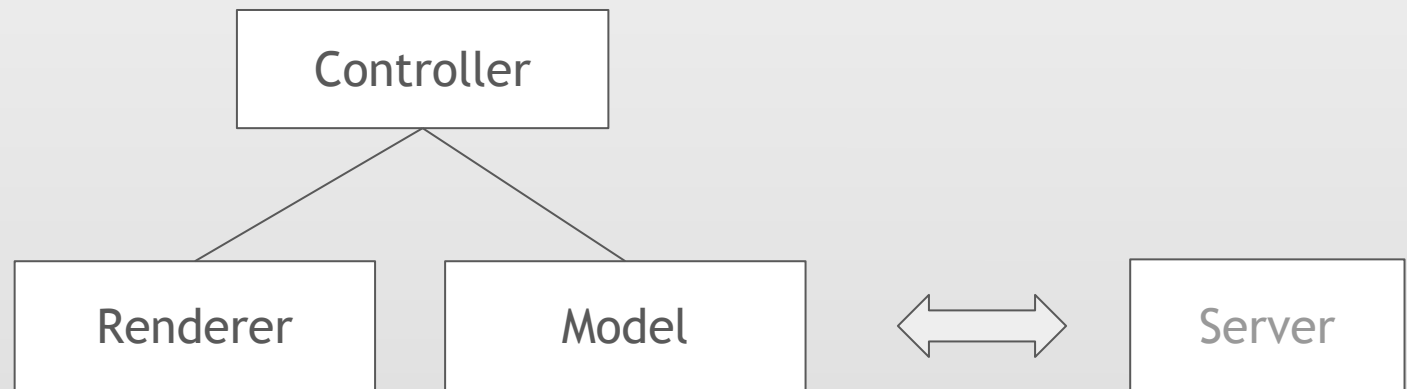
working with JS views

# Views Architecture: MVC



# Views Architecture: main classes and roles

- *AbstractView*: factory, set up the MVC, parse the arch, configure renderer, controller, model
- *AbstractController*: coordinate the renderer/model, coordinate with the rest of the web client (and the search view, indirectly), take care of the control panel
- *AbstractRenderer*: rendering
- *AbstractModel*: load and process data, talk to the server

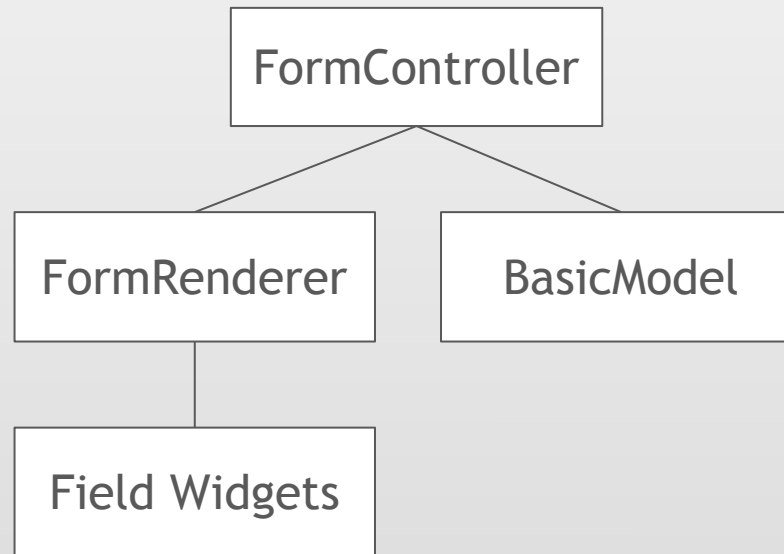


# Basic Views

BasicView: Form, List, Kanban

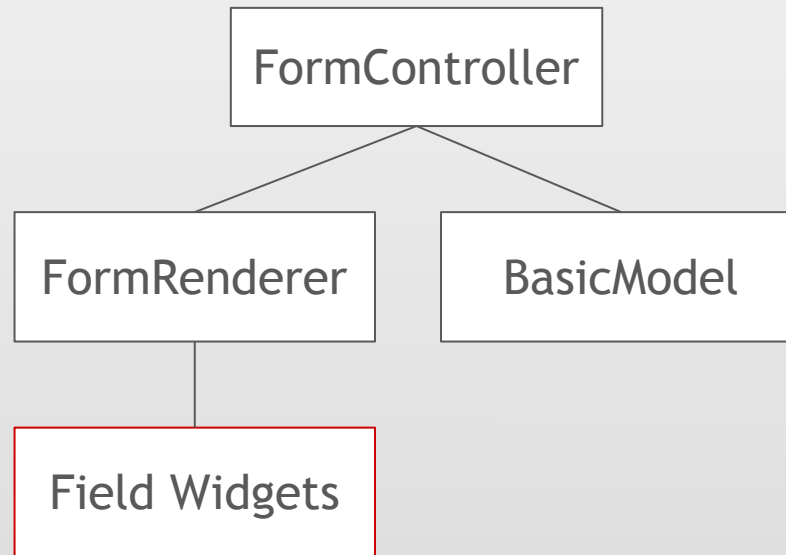
Common base: BaseModel, BasicRenderer, BasicView, BasicController

Support for field widgets, for modifiers (readonly/invisible/...), ...



# AbstractField

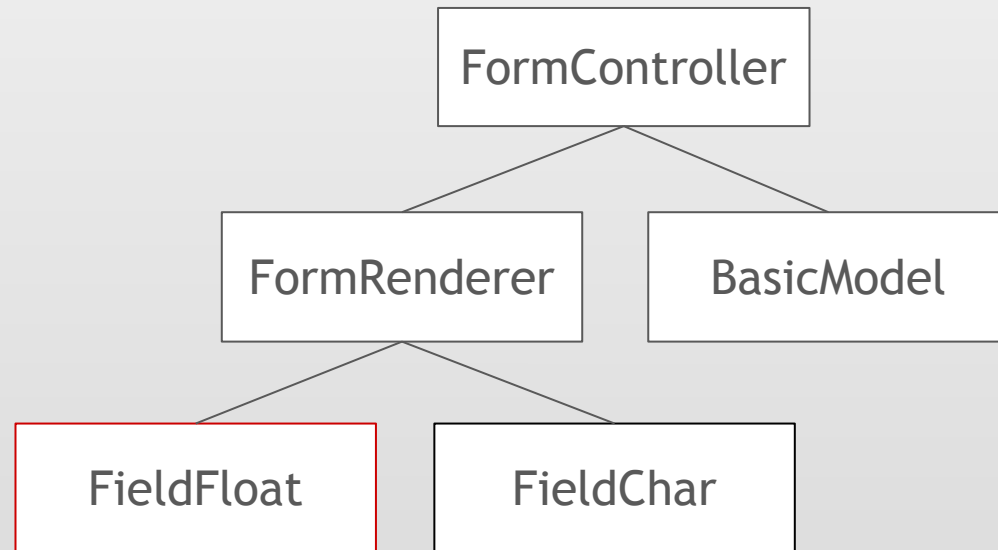
- 2 modes: readonly, edit
- does not switch between mode (is destroyed and recreated)
- represent a value, but is not the owner of the state
- works in all basic views (form/list/kanban)



# AbstractField:

## what happens when a value change

1. the user changes a value (for ex, in a float field, form view, edit mode)

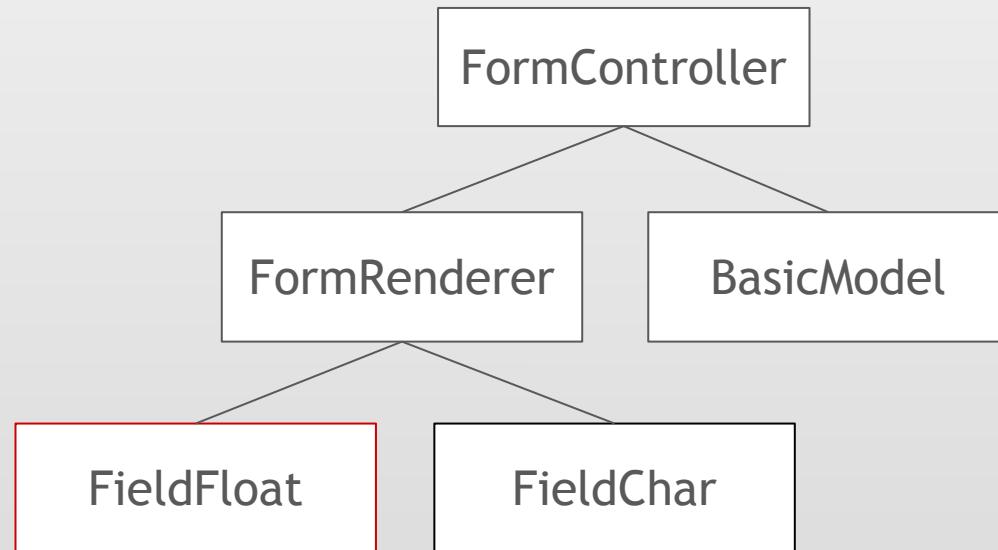




# AbstractField:

## what happens when a value change

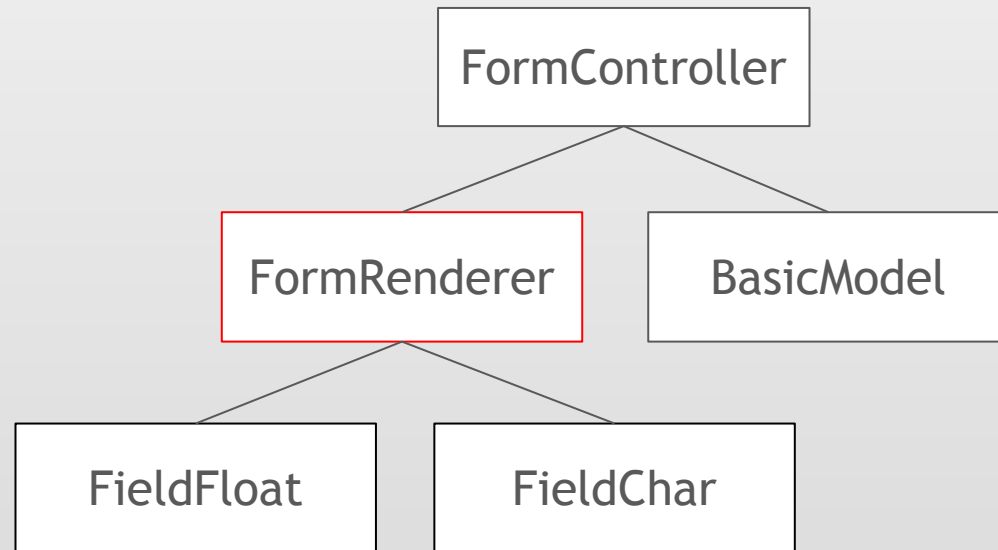
1. the user changes a value (for ex, in a float field, form view, edit mode)
2. the widget call the `_setValue` method, it triggers up an event



# AbstractField:

## what happens when a value change

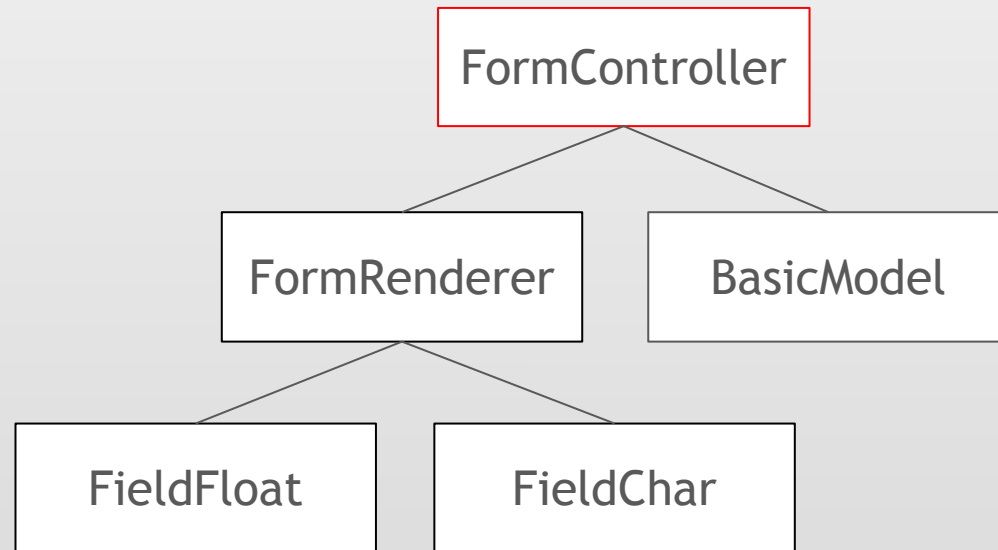
1. the user changes a value (for ex, in a float field, form view, edit mode)
2. the widget call the `_setValue` method, it triggers up an event
3. the event bubble up to renderer, nothing happens



# AbstractField:

## what happens when a value change

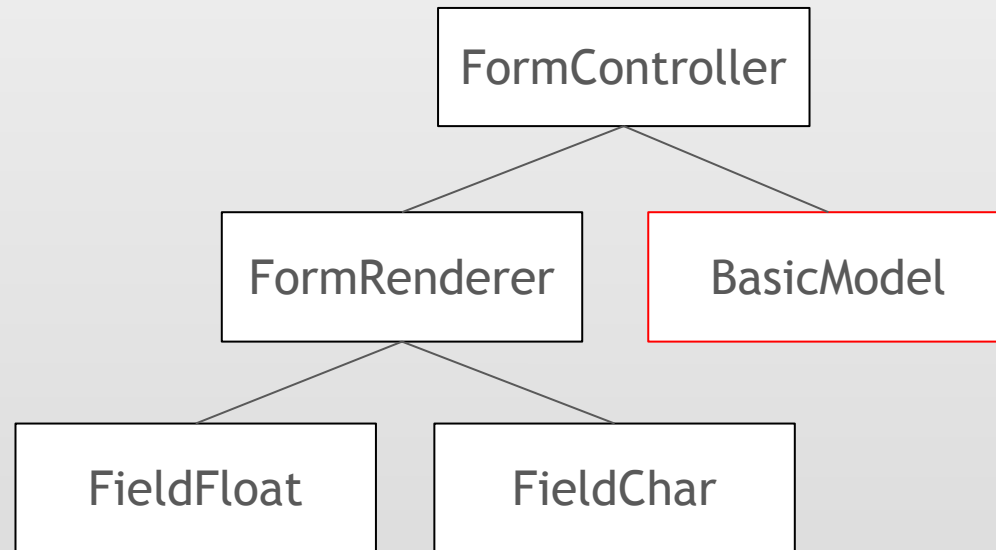
1. the user changes a value (for ex, in a float field, form view, edit mode)
2. the widget call the `_setValue` method, it triggers up an event
3. the event bubble up to renderer, nothing happens
4. it arrives in controller



# AbstractField:

## what happens when a value change

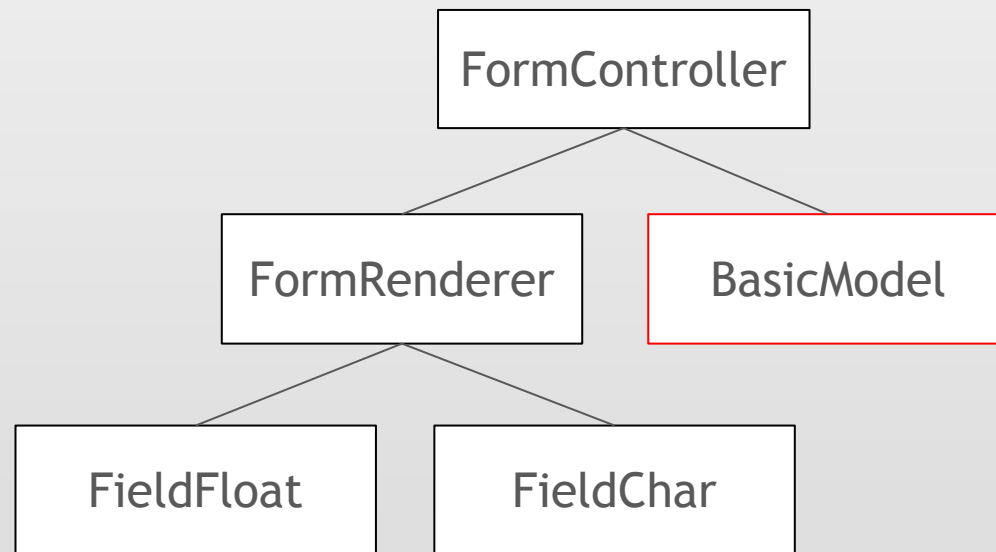
1. the user changes a value (for ex, in a float field, form view, edit mode)
2. the widget call the `_setValue` method, it triggers up an event
3. the event bubble up to renderer, nothing happens
4. it arrives in controller
5. controller notifies model



# AbstractField:

## what happens when a value change

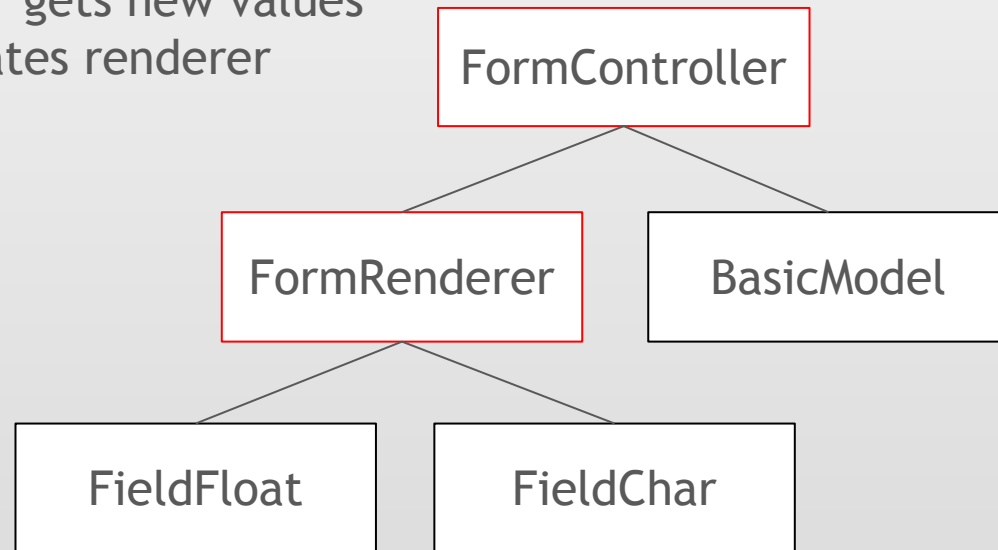
1. the user changes a value (for ex, in a float field, form view, edit mode)
2. the widget call the `_setValue` method, it triggers up an event
3. the event bubble up to renderer, nothing happens
4. it arrives in controller
5. controller notifies model
6. model applies change, applies onchange if necessary (async: talk to server)



# AbstractField:

## what happens when a value change

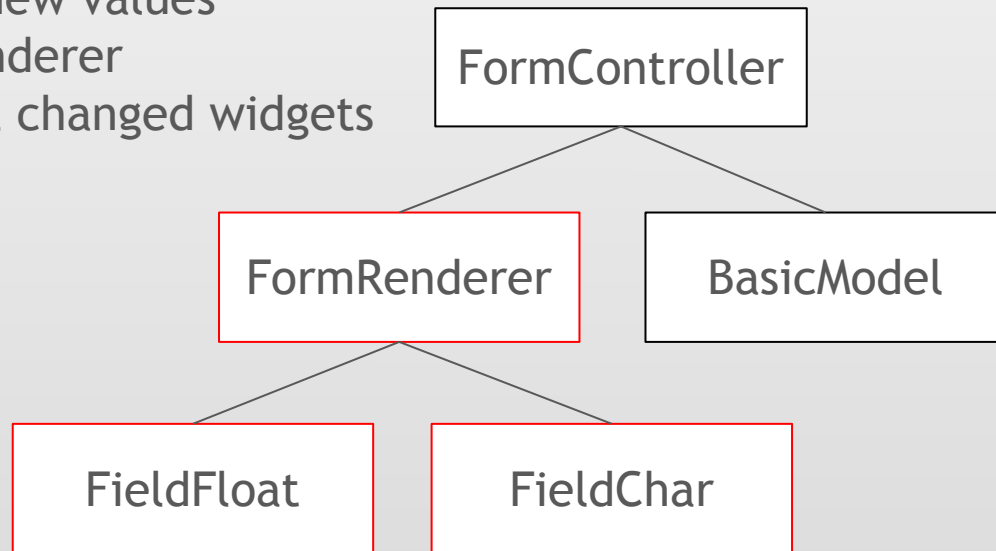
1. the user changes a value (for ex, in a float field, form view, edit mode)
2. the widget call the `_setValue` method, it triggers up an event
3. the event bubble up to renderer, nothing happens
4. it arrives in controller
5. controller notifies model
6. model applies change, applies onchange if necessary (async: talk to server)
7. the controller gets new values and updates renderer



# AbstractField:

## what happens when a value change

1. the user changes a value (for ex, in a float field, form view, edit mode)
2. the widget call the `_setValue` method, it triggers up an event
3. the event bubble up to renderer, nothing happens
4. it arrives in controller
5. controller notifies model
6. model applies change, applies onchange if necessary (async: talk to server)
7. the controller gets new values and updates renderer
8. renderer updates all changed widgets



Let's get to work.