

3368

- ① refactoring
- ② Jinja templating

idea  
we have the  
same  
implementation on  
front-end but  
2 diff  
implementations on  
backend.

eliminate  
team-specific  
implementation

A) Replace widget at every call site (3368)

B) Make generic widget more semantic (3367) ← this is already done, we're adding a refactoring / templating change

- store selected values in `<select multiple>` HTML
- remove all references to `multiple_team_selector`

### Call Sites

- 1) admin project
- 2) contributor settings
- 3) messaging compose
- 4) existing tags page

### Backend Form

CharField → ModelMultipleChoiceField

### Build Pipeline

/base.py, drop deleted CSS/JS bundles

aside

Machine Translation calls) `pretranslate.py` / `transvision.py` / `translator.py`

- 2 pane shuttle widget, available (left), chosen (right)  
reusable Jinja macro

### Jinja re-cap

- create smart documents
- templating engine for python, write static (HTML, SQL, config) inject dynamic data
- create single Jinja template & engine fills in blanks

variables { { ... } }

control structures { % ... % } # logic loops, if/then, only show on specific condition

comments { # ... # }

### refactoring

- enhance code for readability (e.g. arranging a kitchen)
- extensibility) making code flexible enough so

### 2 widgets, move objects to sides

- have 2 versions of very similar behavior, so collapse into multiple item selector.
- item / team selector different HTML files, item selector list nested
- team selector either collapsed to make it one OR
  - ↳ also includes locale code

### Team selector

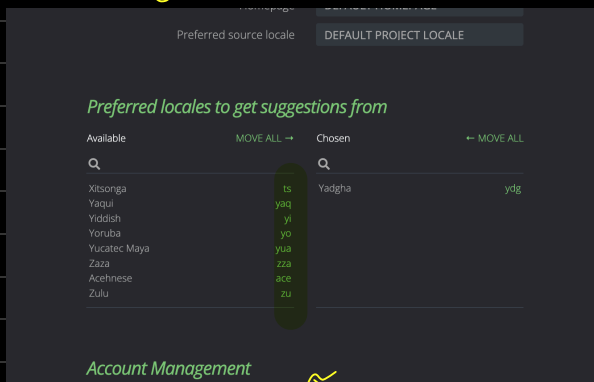
- default locales, } beware of impacting the CSS since direct reference

### Current Situation

- UI pattern in 3 places: pick some subset of items from a long list
- 2 parallel implementations

### changes

① locale not green, fix:

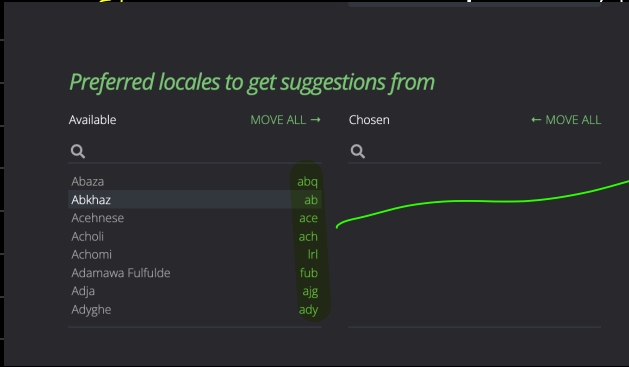


CSS changes propagate and cause this drift against dev for first fix

### Pages where widget is rendered

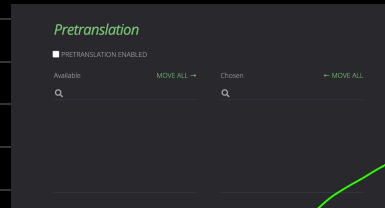
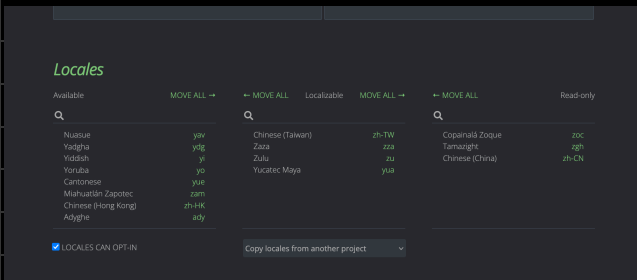
- ① /settings
- ② /admin/projects/<slug>
- ③ /messaging/<id>

① /settings → CSS issue part change

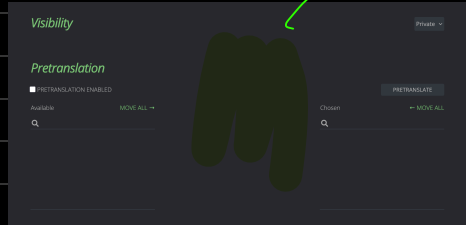


difference in color, part does not match properly

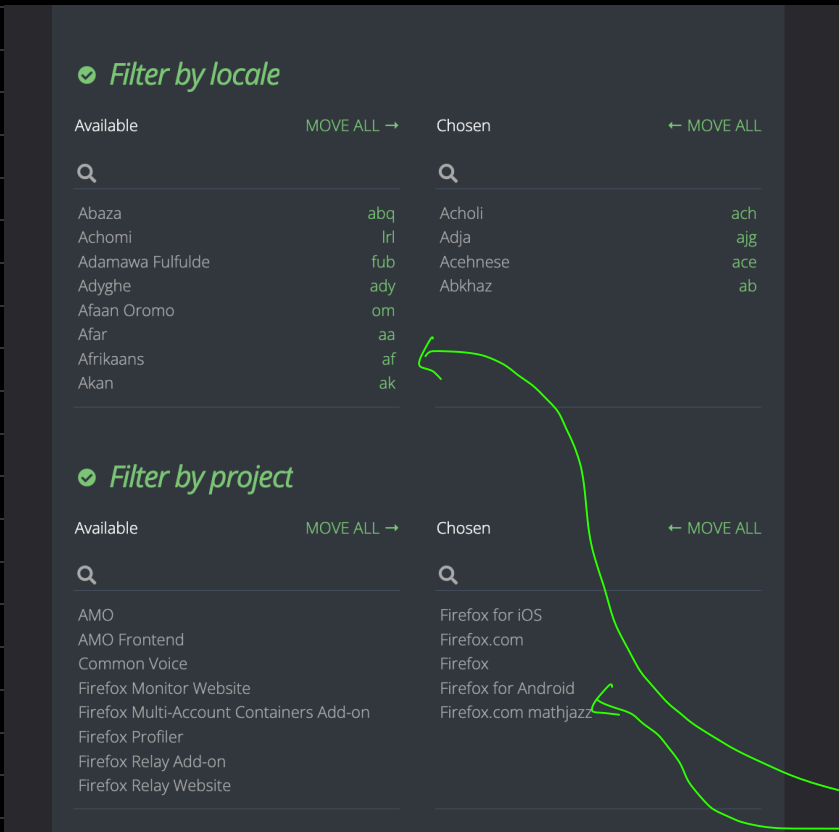
② /admin/projects → 2.1 Locales, 2.2 Pretranslation



have extra distance here in the /admin/projects



③ /messaging



standardize

ensure similarity

nested multiple\_item\_selector.css per pipeline impact audit

→ admin\_project

```
# admin_project.css 1 x
pontoon > administration > static > css > # ac
46
47 form > div {
48   margin: 20px 0;
49   text-align: right;
50 }
51
52 form > div.controls:last-child {
```

```
# admin_project.css 1 x
pontoon > administration > static > css > #
54
55
56 form > section > div {
57   margin-top: 20px;
58 }
59
```

/ form a rules are gone, recheck labels & links inside form

→ project

// latent dead weight, no direct problem as given basis for that

→ settings

```
6 results in 6 files - Open in editor
# admin_project.css pontoon/a... 1 1
padding-bottom: 3px;
# multiple_item_selector.css pontoo... 1
padding-bottom: 3px;
# request.css pontoon/teams/static/css 1
padding-bottom: 3px;
# Entity.css translate/src/modules/ent... 1
padding-bottom: 3px;
# MachinerySkeletonLoader.css tra... 1
padding-bottom: 3px;
# SkeletonLoader.css translate/src/m... 1
padding-bottom: 3px;
```

→ loss of padding pixels  
probability on)

① <section> elements

② <form> "

③ <label> "

→ messaging (scoped, complete)

smoke test, build verification test

→ 4 leaked styling aspects

① /settings/

② /admin/projects/<slug>/

③ /messaging/

④ /projects/<slug>/

Things team widget did that generic widget didn't

- ① colored language badge + code on the right
- ② wrote selected IDs into hidden input (non-Django, non-`<select>` form can submit list)

locals has code attribute, generic items like props don't

parameter added to macro signature so can receive name

```
@@ -1,4 +1,4 @@
1 - {% macro list(id, title, items, left_shortcut=None, right_shortcut=None, sortable=False) %}
1 + {% macro list(id, title, items, left_shortcut=None, right_shortcut=None, sortable=False, form_field=None) %}
2   <div class="item select {{ id }}">
3     <label for="{{ id }}">
4       {% if left_shortcut %}
@@ -19,11 +19,24 @@
19     {% for item in items %}
20       <li class="clearfix" data-id="{{ item.id }}">
21         <span class="arrow fas"></span>
22 -       <span class="item">{{ item.name }}</span>
22 +       {% if item.code %}
23 +       <span class="language {{ item.code }}">{{ item.name }}</span>
24 +       <span class="code">{{ item.code }}</span>
25 +       {% else %}
26 +       <span class="item">{{ item.name }}</span>
27 +       {% endif %}
28     </li>
29     {% endfor %}
30     <li class="no-match">No results</li>
31   </ul>
32 </div>
33 +
34 + {
35 +   {% if form_field %}
36 +   <input
37 +     type="hidden"
38 +     name="{{ form_field }}"
39 +     value="{{ items|map(attribute='pk')|join(',') }}"
40 +   />
41 +   {% endif %}
42 </div>
{% endmacro %}
```

team widget submitted list as single hidden input w/ comma-joined IDs

if code exists, render as language row, otherwise render it as a plain item

- generic widget does a plain HTML form (used by user settings)  
or Django frontend (used by admin)

depending on what caller pass

form\_field) can be either pass string (name for hidden input) or Django form field object (renders itself)

```

@@ -18,9 +18,10 @@
18 18     items=items_selected,
19 19     right_shortcut='- Move all',
20 20     sortable=sortable,
21 21 +     form_field=form_field if form_field is string else None,
21 22 )
22 23 }}
23 24 - {% if form_field %}
24 25 + {% if form_field and form_field is not string %}
24 25     {{ form_field }}
25 26     {% endif %}
26 27 </div>

```

if string, forward to inner list to render input\_type hidden,  
if field object, render field directly

JS array) assures every instance had Django <select> element, so unconditionally updated options

```

@@ -5,16 +5,23 @@ $(function () {
5 5     /*
6 6     function updateSelectedItems(element) {
7 7         const widget = $(element).parents('.multiple-item-selector');
8 8 -         const selectElement = widget.find('select');
9 9         const selectedItems = widget
10 10             .find('.item.selected li[data-id]')
11 11             .map(function () {
12 12                 return $(this).data('id');
13 13             });
14 14         .get();
15 15
16 16 -         for (const option of selectElement[0].options) {
17 17 -             option.selected = selectedItems.includes(parseInt(option.value));
18 18 +         const selectElement = widget.find('select');
19 19 +         if (selectElement.length) {
20 20 +             for (const option of selectElement[0].options) {
21 21 +                 option.selected = selectedItems.includes(parseInt(option.value));
22 22 +             }
23 23 +         }
24 24 +         const hiddenInput = widget.find('.item.selected input[type=hidden]');
25 25 +         if (hiddenInput.length) {
26 26 +             hiddenInput.val(selectedItems.join(','));
27 27         }

```

<select> admin  
hidden <input> settings, messaging

- missing is no-op rather than crash since both branches guarded with length

no operation) command / instruction that does nothing when executed

styles that team widget had that generic widget didn't  
search input not being styled to fill container

```
9 pontoon/base/static/css/multiple_item_selector.css
@@ -22,19 +22,28 @@
22 22     background: transparent;
23 23     border-bottom: 1px solid var(--main-border-1);
24 24     bottom: auto;
25 25 +     box-sizing: border-box;
26 26     display: inline-block;
27 27     margin: 2px 0 -4px -1px;
28 28     overflow: auto;
29 29     padding: 10px 0;
30 30     position: relative;
31 31     width: 295px;
32 32
33 33 +     input[type='search'] {
34 34 +         border: none;
35 35 +         box-sizing: border-box;
36 36 +         width: 100%;
37 37 +     }
38 38 +
39 39     ul {
40 40         height: 170px;
41 41         margin-bottom: 0;
42 42
43 43         li span.code {
44 44 +             color: var(--status-translated-alt);
45 45             float: right;
46 46 +             text-align: left;
47 47             width: auto;
48 48         }
49 49     }
@@@
```

border-box sizing  
added so that  
wid this line up  
and avoids  
padding + border  
push lower

→ needs  
color/alignment

```
11 pontoon/administration/templates/admin_project.html
@@ -1,6 +1,6 @@
1 1     {% extends "base.html" %}
2 2     {% import 'admin_team_selector.html' as admin_team_selector %}
3 3 -     {% import 'teams/widgets/multiple_team_selector.html' as multiple_team_selector %}
4 4 +     {% import 'widgets/multiple_item_selector.html' as multiple_item_selector %}
5 5     {% import 'admin_inline_formset.html' as inline_formset %}
6 6     {% import 'tag_inline_formset.html' as tag_inline_formset %}
@@@
@@ -320,8 +320,13 @@ <h3>
320 320         </div>
321 321     </div>
322 322     <div class="locales-pretranslate clearfix">
323 323 -         {{ multiple_team_selector.render(locales_pretranslate_available, locales_pretranslate) }}
324 324 -         {{ form.locales_pretranslate }}
325 325 +         {{
326 326 +             multiple_item_selector.render(
327 327 +                 locales_pretranslate_available,
328 328 +                 locales_pretranslate,
329 329 +                 form_field=form.locales_pretranslate,
330 330             }}
331 331         {{ form.locales_pretranslate.errors }}
332 332     </div>
@@@
```

switches to  
generic widget,  
Django form  
label panel  
into rows

→ renders internally  
as not string  
hence explicit  
reference not needed

```

    @@ -13,10 +13,6 @@ $(function () {
13 13         selector: '.admin-team-selector .locale.readonly',
14 14         input: $('#id_locales_readonly'),
15 15     },
16 -     {
17 -         selector: '.multiple-team-selector .locale.selected',
18 -         input: $('#id_locales_pretranslate'),
19 -     },
20 16     ];
21 17
22 18     locales.forEach(function (type) {

```

used to maintain  
#id\_locales\_pretranslate  
(Django field's <select>)

JS now does this  
synchronization so duplicate  
handler is removed

Showing 17 changed files with 76 additions and 102 deletions.

```

    @@ -185,7 +185,10 @@ form a:visited {
185 185     }
186 186
187 187     .double-list-selector .locale.select .menu {
188 +     background: transparent;
189 +     border-bottom: 1px solid var(--main-border-1);
190     margin: 2px 0 -4px -1px;
191 +     overflow: auto;
189 192     padding: 10px 0;
190 193     width: 295px;
191 194     }
    @@ -230,6 +233,11 @@ form .locales-pretranslate .locale.select.selected {
230 233     margin-left: 49px;
231 234     }
232 235
236 + form .locales-pretranslate .item.select.selected {
237 +     float: none;
238 +     margin: 0 49px;
239 + }
240 +
233 241     .locale.select.readonly {
234 242         float: right;
235 243     }

```

inlined into  
admin\_project.css  
from  
multiple\_team\_selector.css

layout tweak because new generic widget uses  
.item.select.selected & old uses .locale.select.selected

```

    @@ -3,7 +3,7 @@
3 3     {% import "widgets/toggle.html" as Toggle %}
4 4     {% import "widgets/field.html" as Field %}
5 5     {% import "widgets/theme_toggle.html" as ThemeToggle %}
6 - {% import "teams/widgets/multiple_team_selector.html" as multiple_team_selector %}
6 + {% import "widgets/multiple_item_selector.html" as multiple_item_selector %}
7 7     {% import "teams/widgets/team_selector.html" as team_selector %}
8 8
9 9     {% block title %}Settings{% endblock %}
    @@ -426,7 +426,7 @@ <h3>Default locales</h3>
426 426     <h3>Preferred locales to get suggestions from</h3>
427 427     <div>
428 428         {{
429 -         multiple_team_selector.render(available_locales, selected_locales, form_field='locales_order',
429 +         multiple_item_selector.render(available_locales, selected_locales, form_field='locales_order',
430 430         sortable=True)
431 431     }}
432 432     </div>

```

call-site swap

form\_field string  
'locales\_order'

routes through  
new render hidden  
input branch