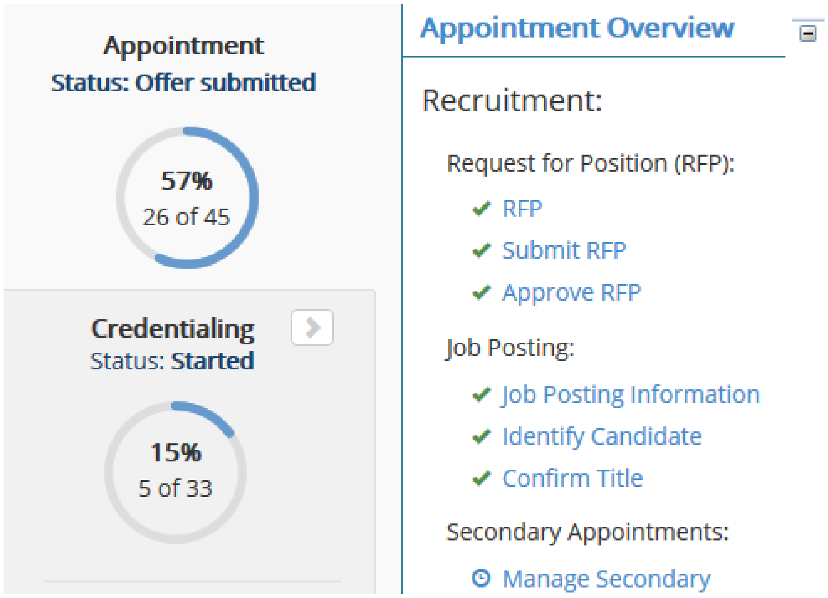
MPS UX Notes

Some of this will be pretty generic & is shared info Front & Back End.

MPS Appt > Workflow & related JSON:



Appointment
Status: Offer submitted

57%
26 of 45

Credentialing
Status: Started

15%
5 of 33

Appointment Overview

Recruitment:

Request for Position (RFP):

- ✓ RFP
- ✓ Submit RFP
- ✓ Approve RFP

Job Posting:

- ✓ Job Posting Information
- ✓ Identify Candidate
- ✓ Confirm Title

Secondary Appointments:

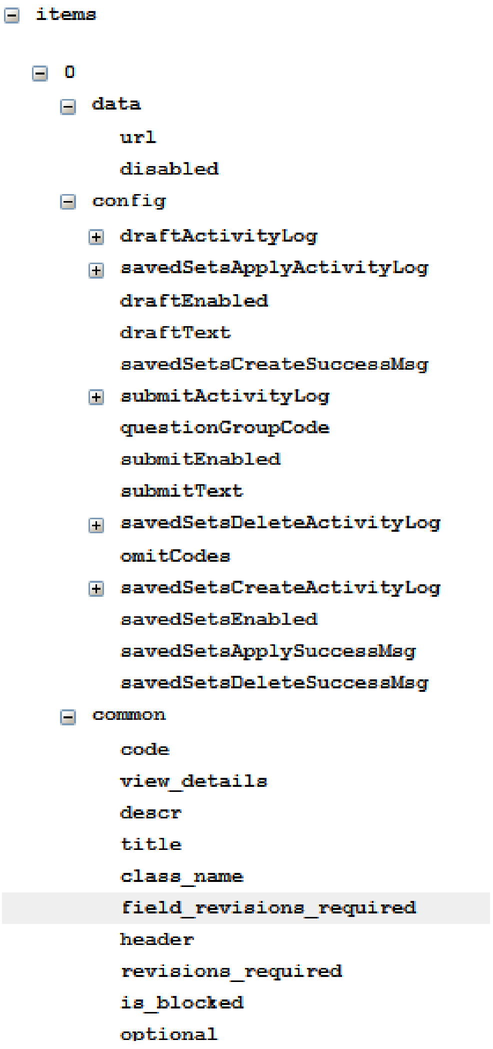
- Manage Secondary

The back end sends a hefty all-included Workflow definition as JSON:

Each container is turned into a UX Column of item links, each Section is a Heading, each Item is a Link in a Column/Heading.

The Front End (FE) additionally pre-processes the Workflow JSON to add some UX state properties (e.g. 'TBD-ness', 'Blocked-ness', etc), then iterates through building containers & items for the page DOM.

Each Item is widget-ized as a 'superItem', and some are further widget-ized as e.g. Upload Controls or Server Side forms.



```
⊟ items
  ⊟ 0
    ⊟ data
        url
        disabled
    ⊟ config
      ⊞ draftActivityLog
      ⊞ savedSetsApplyActivityLog
        draftEnabled
        draftText
        savedSetsCreateSuccessMsg
      ⊞ submitActivityLog
        questionGroupCode
        submitEnabled
        submitText
      ⊞ savedSetsDeleteActivityLog
        omitCodes
      ⊞ savedSetsCreateActivityLog
        savedSetsEnabled
        savedSetsApplySuccessMsg
        savedSetsDeleteSuccessMsg
    ⊟ common
        code
        view_details
        descr
        title
        class_name
        field_revisions_required
        header
        revisions_required
        is_blocked
        optional
```

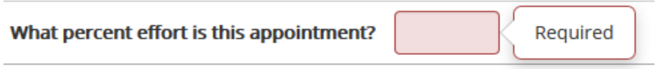
All items are widget-ized & ready in the DOM at load, the Workflow Column & item Links simply hide/show the requested item and handle some light Breadcrumb Bar UX.

The notable exception is the now-ubiquitous Server Side Form (below), which is made a widget at page load, but waits until user click → load to request the server template/form for its content.

'Server Side Forms' Widget:

Originally we had a number of different item widgets, but later features such as Uber Forms plus a huge amount of overlap on feature requirements & shared code in all of them resulted in most items being 'Server Side Forms' in the end, with shared features such as:

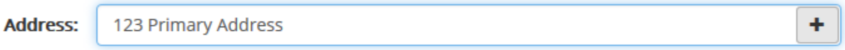
- Validation/Error handling after submit:



What percent effort is this appointment?

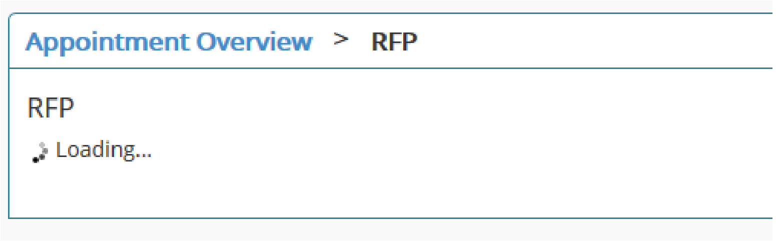
Required

- Repeating Inputs/Selects



Address: 123 Primary Address

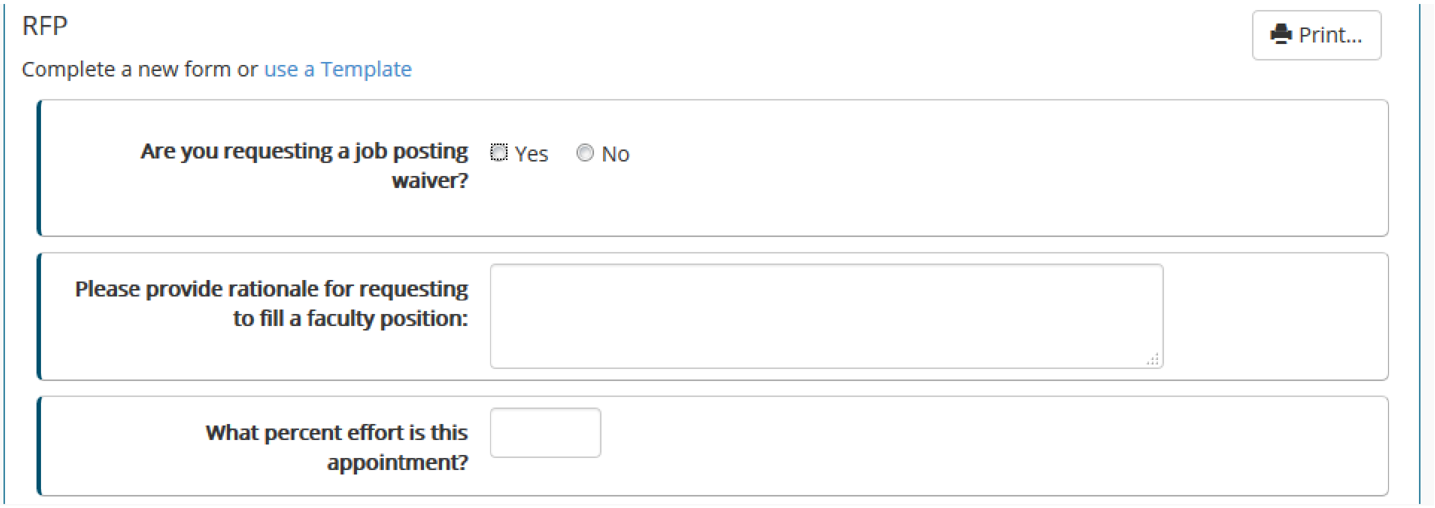
- Repeating Forms (with or without a summary table UX above them)
- Date-Pickers
- Etc.



Appointment Overview > RFP

RFP

Loading...



RFP

Complete a new form or use a Template

Are you requesting a job posting waiver? ○ Yes ○ No

Please provide rationale for requesting to fill a faculty position:

What percent effort is this appointment?

Candidate View:

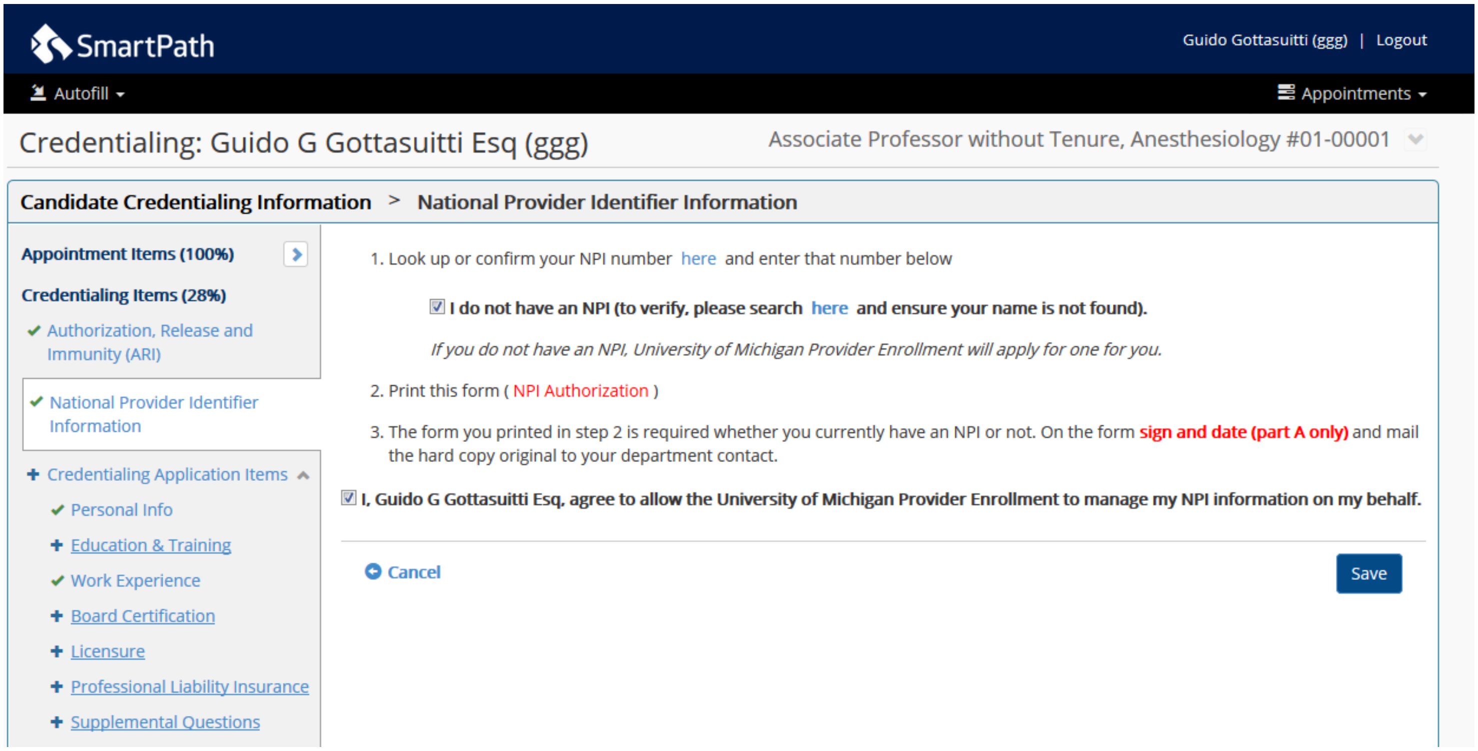
Is a hybrid presentation of the Workflow > Containers for those items the Candidate needs to view, review, edit & complete.

Containers in Candidate View are turned into a LeftNav instead of Workflow Column(s) in the right side page content.

Workflow Dials in Candidate View are turned into ‘just links’ Headers (the ones with 100%, 28%, etc below) in the leftNav.

Because there is no Overview of Columns & Links to return to, there is special & nav logic to load the ‘next TBD item’.

Also there is a ‘Thank You’ Page state when all items have been completed – but it is not an item, it is just a DIV in DOM.


SmartPath
Guido Gottasuitti (ggg) | Logout
Autofill
Appointments
Appointment Items (100%)
Credentialing Items (28%)
✓ Authorization, Release and Immunity (ARI)
✓ National Provider Identifier Information
+ Credentialing Application Items
✓ Personal Info
+ Education & Training
✓ Work Experience
+ Board Certification
+ Licensure
+ Professional Liability Insurance
+ Supplemental Questions
Cancel
Save

Credentialing: Guido G Gottasuitti Esq (ggg)

Associate Professor without Tenure, Anesthesiology #01-00001

Candidate Credentialing Information > National Provider Identifier Information

1. Look up or confirm your NPI number here and enter that number below

☑ **I do not have an NPI (to verify, please search here and ensure your name is not found).**

If you do not have an NPI, University of Michigan Provider Enrollment will apply for one for you.

2. Print this form (NPI Authorization)

3. The form you printed in step 2 is required whether you currently have an NPI or not. On the form **sign and date (part A only)** and mail the hard copy original to your department contact.

☑ **I, Guido G Gottasuitti Esq, agree to allow the University of Michigan Provider Enrollment to manage my NPI information on my behalf.**

Date-Pickers:

<input class="form-control mps-date-picker" type="text" placeholder="{{ context.get('date_format','MM/DD/YYYY') }}" value="" name="start_date" id="start_date"/>

Required:

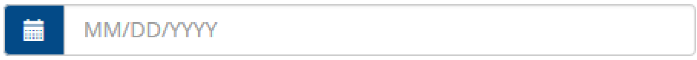
Server-side, all that is required to have a date picker appear in any Form is the addition of the className 'mps-date-picker' to a text input. After that, we do FE JS & Bootstrap to approximate an HTML 5 type='date' UX.

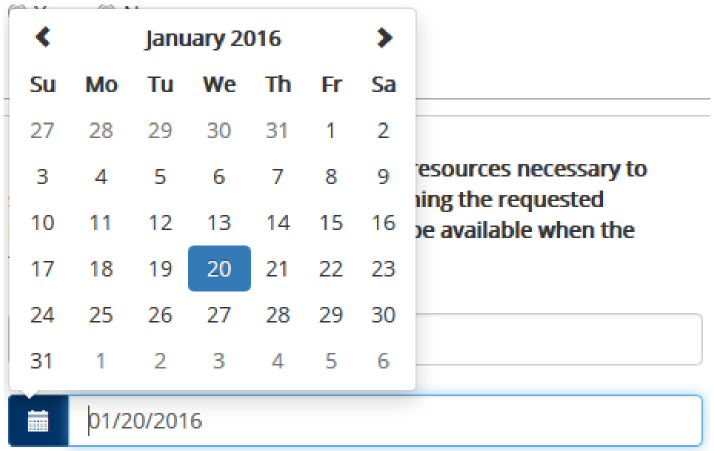
The FE scrapes loaded form content for .mps-date-picker class and date-picker-izes it at load, including insertion of Bootstrap grouped button HTML & then the call to the Datepicker widget.

Optional:

data-now-date-format="{{context.get('date_format','') }}"

e.g. Record Results items often have these fields: the data-now-format is for date-picker inputs that wish to display 'Now' in the Placholder text & allow empty submits. The extra attribute is looked for so that the date-picker widget can use the correct mm/dd/yyyy format.





Su	Mo	Tu	We	Th	Fr	Sa
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

01/20/2016

Dashboard:

Roster: