



## Portal Pet Adoption and Thoughts on Modals

I hosted a webinar back in July where we built a not-so-simple Service Portal widget from scratch. If you haven't seen it, [check it out here](#). For those of you looking for the code, your wish has been granted. I've posted below the HTML, Client Controller, Server Script, and CSS for the main widget and the embedded widget.

Since July I've also had a few thoughts about the widget we built, mainly around whether to use spModal or \$uibModal for the modal dialogue. I'd like to share this first before spilling all the code.

### Thoughts on modals in Service Portal

In the webinar, we relied on a modal dialogue for presenting the pet selection widget. I was torn between whether to use the Service Portal modal wrapper (spModal), or to use the bootstrap component itself (\$uibModal). Though they both result in a similar experience for the user, it's the experience for the developer that's quite different.

spModal is very easy to use. There used to be some nice documentation for it on gitHub, but alas, it was removed in August, due to SN product documentation politics I'm guessing. ServiceNow, if you're reading this, PLEASE BRING BACK THE SP GITHUB REPO!!! Anyway, spModal allowed us to easily open a modal dialogue and define it's messages, buttons, etc. The drawback is that the more advanced options were limited

[+ Ready to learn more?](#)

## CATEGORIES

- BLOG
- CAPIO BLOG & NEWSFEED
- CASE STUDIES
- PODCAST
- PORTAL GURU BLOG
- PRESS RELEASES & PRODUCT UPDATES
- SERVICE PORTAL
- WEBINAR

## ARCHIVES

- FEBRUARY 2019
- JANUARY 2019
- DECEMBER 2018
- NOVEMBER 2018
- OCTOBER 2018
- SEPTEMBER 2018
- AUGUST 2018
- JULY 2018
- JUNE 2018
- MAY 2018
- APRIL 2018
- MARCH 2018
- FEBRUARY 2018

For example, in the widget we built, we wanted a 'submit' button for each pet in our selection. However, the 'buttonClicked()' function we needed to trigger was not available to us in the spModal API. So, we had to use angular.element to do some DOM manipulation in the client controller to get the user experience we wanted; and DOM manipulation from the client controller is frowned upon in front-end dev. But I'm a rebel, Dottie. I did it anyway.

Our other option was to use the bootstrap UI modal component to define our modal window. The drawback here is slightly more complex code, and having to put my CSS in a global style sheet rather than in the widget's CSS field. Furthermore, I'd have to write my HTML template in a script tag in the main widget, but I really wanted to demonstrate embedding a widget for the purpose of logically separating views and functions.

I could have gone either way, and I could go on for another few paragraphs about the differences, pros, and cons. But you didn't come here for a lecture on modals. You want the code for the widgets...

Anyhow, I realized since July that I didn't have to DOM manipulation via angular.element to call the modal's 'clickButton()' function. From the embedded widget I have access to parent scopes, which means that 'clickButton()' as well as the buttons object live just two steps above in the hierarchy.

Therefore, I can call

`'$scope.$parent.$parent.buttonClicked($scope.$parent.$parent.options.buttons[0])'` from anywhere I want in my embedded widget HTML.

Voila! All the \$uibModal functions and objects are available to me when using spModal. I've since done this in new widgets and it works like a charm!

## Don't miss our September webinar

Be sure to reserve your place for our **September 7th webinar**.

It's far less technical than the last portal webinar, but just [+ Ready to learn more?](#)

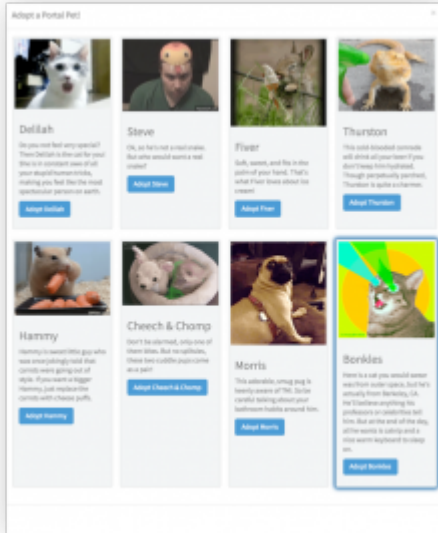
DECEMBER 2017  
NOVEMBER 2017  
OCTOBER 2017  
SEPTEMBER 2017  
AUGUST 2017  
JULY 2017  
JUNE 2017  
MAY 2017  
APRIL 2017  
MARCH 2017  
DECEMBER 2016  
OCTOBER 2016  
SEPTEMBER 2016  
AUGUST 2016  
JULY 2016

useful. Hope to see you there!

## 5 Essentials All Great Service Portals Have in Common

# Portal Pet Adoption Widget

Here's all the code from the widget webinar. Enjoy!



## Main Widget HTML

```
<div ng-if="!data.currentPet">
  <button ng-click="c.onBrowsePets('lg')" class="adopt btn btn-primary">
    {{Need emotional support?}}
  </button>
</div>

<!--House for the adopted pet to live in-->
<div class="pet-house-outer-container" ng-if="data.currentPet">
  <div class="pet-house-container">
    <div class="pet" style="background: url('{{data.currentPet.photo}}') no-repeat center center; background-size: cover;">
    
    <div class="pet-name">
      {{data.currentPet.name}}
    </div>
  </div>
  <!--Link to browse pets-->
  <a class="settings" href="javascript:void(0)" ng-click="c.onBrowsePets('lg')">
    <i class="fa fa-cog"></i>
  </a>
</div>
</div>

<div ng-if="!data.currentPet">
  <button ng-click="c.onBrowsePets('lg')" class="adopt btn btn-primary">
    {{Need emotional support?}}
  </button>
```

[+ Ready to learn more?](#)

```

</div>

<!--House for the adopted pet to live in-->
<div class="pet-house-outer-container" ng-if="data.currentPet">
  <div class="pet-house-container">
    <div class="pet" style="background: url('{{data.currentPet.photo}}') no-repeat center center; background-size: cover;">
    
    <div class="pet-name">
      {{data.currentPet.name}}
    </div>
  </div>
  <!--Link to browse pets-->
  <a class="settings" href="javascript:void(0)" ng-click="c.onBrowsePets()">
    <i class="fa fa-cog"></i>
  </a>
</div>
</div>

```

## Main Widget Client Controller

```

function (spModal) {
  //Including spModal service
  var c = this;
  var shared = {}; //This is an empty array we can add to and share
  c.onBrowsePets = function(size){
    //This function is called by a button in the HTML

    shared.currentPet = c.data.currentPet; //We store the currentPet d

    spModal.open({
      //spModal.open will open a modal, and we'll pass it the following
      size: size,
      title: 'Adopt a Portal Pet!',
      widget: '*****', //Insert sys_id of the embedded widget - "Pet S
      buttons: [
        {label:'${Adopt}', primary: true}
      ],
      shared: shared //We make the 'shared' array available to modal's e
    }).then(function() {
      //This function is triggered by submitting/closing the modal.
      c.data.selectedPet = shared.selectedPet; //The selected pet is sha
      c.server.update(); //Send the updated data object to the server sc
    });
  }
}

```

## Main Widget Server Script

```

(function() {

  //Check the Pet Adoption table to populate the currentPet

```

[+ Ready to learn more?](#)

```
var userPet = new GlideRecord('u_pet_adoptions');
userPet.addQuery('u_user',gs.getUserID());
userPet.query();
if(userPet.next()) {
  data.currentPet = {
    //build the currentPet object
    name: userPet.u_pet.u_name.toString(),
    photo: userPet.u_pet.u_photo.getDisplayValue().toString(),
    id: userPet.sys_id.toString()
  }
}
```

## Main Widget CSS

```
img.pet-house,
div.pet,
div.pet-name,
a.settings {
  position: absolute;
}
button.adopt {
  position: relative;
  padding: 10px 20px;
  margin-bottom: 20px;
  font-size: 1.5em;
  width: 100%;
}
div.pet-house-outer-container {
  position: relative;
  height: 220px;
}
div.pet-house-container {
  position: relative;
  width: 200px;
  text-align: center;
  z-index: 9;
  margin: 0 auto;
}
div.pet {
  height: 100px;
  width: 100px;
  top: 99px;
  left: 50px;
  background-size: cover;
}
img.pet-house {
  height: 200px;
  right: 0;
  z-index: 999;
}
div.pet-name {
  width: 100%;
  top: 65px;
```

[+ Ready to learn more?](#)

```

color: #fff;
text-shadow: #414141 1px 1px;
-webkit-transform: rotate(355deg);
-moz-transform: rotate(355deg);
-o-transform: rotate(355deg);
-ms-transform: rotate(355deg);
transform: rotate(355deg);
z-index: 9999;
}
a.settings {
top: 10px;
right: 30px;
color: #7B542B;
font-size: 1.5em;
z-index: 99999;
}

```

## Embedded Widget HTML

```

<div class="row pets">
  <div class="col-sm-6 col-md-3 pet" ng-repeat="pet in data.pets">
    <div class="thumbnail" ng-class="{ 'selected' : c.widget.options.sh
    
    <div class="caption">
      <h3>{{pet.name}}</h3>
      <p>{{pet.bio}}</p>
      <p>
        <button ng-click="c.selectPet(pet)" class="btn btn-primary" role="
      </p>
    </div>
  </div>
</div>
</div>
</div>

```

## Embedded Widget Client Controller

```

function($timeout) {
  var c = this;
  var shared = c.widget.options.shared; //Get the shared object from
  //Function that selects the pet when you click the Adopt button
  c.selectPet = function(pet) {
    shared.selectedPet = pet;

    $timeout(function(){
      //The native 'close' binding from Bootstrap UI Modals is not acces
      //We use Angular Element within a $timeout function to simulate cl
      angular.element('[ng-click*="buttonClicked"]').triggerHandler('cli
      //The modal's native submit button has the ng-click attribute and
      //Therefore, we can target that attribute and click it
    });
  }
  //Disable the visibility of the default modal button

```

[+ Ready to learn more?](#)

```
angular.element('[ng-click*="buttonClicked"]').css({visibility:'hi  
}
```

## Embedded Widget Server Script

```
(function() {  
  //Use GlideRecord to build the list of adoptable pets, store them  
  data.pets=[];  
  var petsGR = new GlideRecord('u_pets');  
  petsGR.query();  
  while(petsGR.next()) {  
    data.pets.push({  
      name: petsGR.u_name.toString(),  
      photo: petsGR.u_photo.getDisplayValue().toString(),  
      bio: petsGR.u_bio.toString(),  
      id: petsGR.sys_id.toString()  
    });  
  }  
})();
```

## Embedded Widget CSS

```
.thumbnail.selected {  
  box-shadow: 0 0 10px 5px #3071a9;  
}  
.pets {  
  display: -ms-flex;  
  display: -webkit-flex;  
  display: flex;  
  flex-wrap: wrap;  
}  
.pet {  
  display: flex;  
}
```

Oh, and here's the pet house png:



Aloha!

[+ Ready to learn more?](#)

Jeff

---



2056 Palomar Airport Road

Carlsbad, CA, 92011

Telephone: +1 844 804 6111 (US)

Telephone: +44 (20) 33254077 (UK)

[Products](#)

[Services](#)

[Resources](#)

[Contact Us](#)



© 2019 Cerna Solutions. All rights reserved. [Privacy Policy](#)

[+ Ready to learn more?](#)