# Liferay AngularJS - simple applications and forms in Web content without using SDK

Liferay provides many ways to develop and extend your portal. We can write portlets, hooks, scripts, themes, layouts etc. But Liferay gives us one more powerful tool which is Web content and JSONWS API.

Very often customers ask us (developers, project managers) if we can give them some tool to display data that they want.

We have 2 options:

* Develop portlet with rich configuration
* Use templates, web content, widgets that can be modified during runtime

Both approaches have own cons:

* Portlets require more resources: SDK, development, deployment, knowledge and time of developer
* Templates, web contents and widgets are more difficult to maintain, and have limited capabilities (can use only existing libraries)

There are much more downsides of both approaches.

I would like show that with existing infrastructure Liferay6.2, JSONWS Api, AngularJS we can meet many small goals just using Web content (just basic web content without templates and structures).

Case 1

I would like to display user name for currently logged-in user

We just need web content:

<script src= "http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>

<div ng-app="myApp" ng-controller="personCtrl">

<button ng-click="userData()">Get user data</button>

<p>User Name: <input ng-model="userName" type="text" /></p>

</div>

<script>

var app = angular.module('myApp', []);

app.controller('personCtrl', ['$scope', '$window', function($scope, $window) {

$scope.userName = "",

$scope.userData = function() {

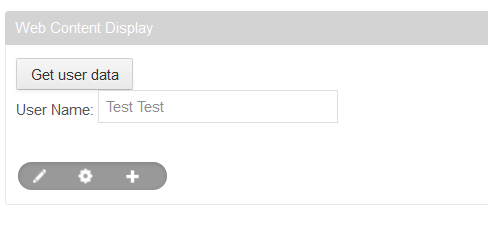
$scope.userName = $window.Liferay.ThemeDisplay.getUserName();

};

}]);

</script>

This web content can be displayed in asset publisher or web content display:



This is very simple and quick example. Angular library and some tags can be moved to the theme, but in this case we can just copy paste the content to vanilla Liferay 6.2 and see same result.

Case 2

I would like to display current user email via JSONWS API and Liferay JS Service.

All JSONWS API methods are available under http://localhost:8080**/api/jsonws** of your installation

We just need following content in our journal article:

<script src= "http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>

<div ng-app="myApp" ng-controller="personCtrl">

<button class="btn" ng-click="userData()">Get user email</button>

<p>User Email: <input ng-model="email" type="text" /></p>

</div>

<script>

var app = angular.module('myApp', []);

app.controller('personCtrl', ['$scope', '$window', function($scope, $window) {

$scope.userData = function() {

$window.Liferay.Service(

'/user/get-user-by-id',

{

userId: $window.Liferay.ThemeDisplay.getUserId()

},

function(obj) {

$scope.email = obj.emailAddress;

}

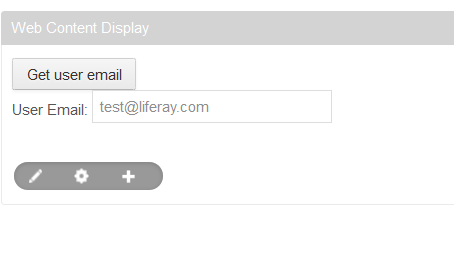
);

};

}]);

</script>

Above code gives following result:



This is another simple example that show how quickly we can expose data to the public in a programmatic way. But web content doesn’t have to be limited only to display things.

Case 3

I would like to modify data of existing user (for example email address)

Again, we will use JSONWS API and Liferay JS service:

<script src= "http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js"></script>

<div ng-app="myApp" ng-controller="personCtrl">

<button class="btn" ng-click="userData()">Get user email</button>

<p>User Email: <input ng-model="email1" type="text" /></p>

<p>Repeat Email: <input ng-model="email2" type="text" /></p>

<button class="btn btn-primary" ng-click="submitData()">Save</button></div>

<script>

var app = angular.module('myApp', []);

app.controller('personCtrl', ['$scope', '$window', function($scope, $window) {

$scope.userData = function() {

$window.Liferay.Service(

'/user/get-user-by-id',

{

userId: $window.Liferay.ThemeDisplay.getUserId()

},

function(obj) {

$scope.email1 = obj.emailAddress;

}

);

};

$scope.submitData = function() {

$window.Liferay.Service(

'/user/update-email-address',

{

userId: $window.Liferay.ThemeDisplay.getUserId(),

password: "",

emailAddress1: $scope.email1,

emailAddress2: $scope.email2

},

function(obj) {

}

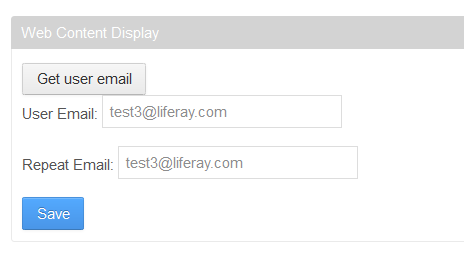
);

};

}]);

</script>

And we have following output:



Of course there are many issues that we didn’t cover in this examples like validation, security etc. There shouldn’t be a problem to add it according to own needs.

Web content base solution can give us many benefits, like:

* Staging
* Metadata (categories, custom fields)
* Versioning
* Power of asset publisher
* Easy to move, update, delete like other articles
* Scheduling
* We can even use workflow :)
* Knowledge of Liferay JSONWS API and Angular should be enough to build many applications and forms
* Easy to build single page application
* Alternative to WordPress Posting Source Code

JSONWS API allows us to expose our own services that can be consumed by others and also invoked in web content. Web content approach is not perfect, but may bring quick solution without big effort. It may encourage more people without deep knowledge of Liferay to build light solutions in the portal.