Speaker Notes

Angular Application Structure Demo

1. Start with DemoStep0, complete will be like DemoStep1
2. Add index.html
3. Add ref to Angular
4. Add ng-app to html element
5. Add evaluated expression and test
6. Add app.js
7. Create app module
8. Add ref and ng-app=”wex1View” to index.html
9. Verify everything still runs
10. Demonstrate using different modules for directives and controllers
    1. Explain that these would likely be in different files
11. Mention
    1. Organization by feature vs. by type
    2. Modules are independent of file structure/order
    3. Explain the folder structure

Controllers Demo

1. Start with DemoStep1, complete will be like DemoStep2
2. Add ExpensesSearchCtrl, injecting $scope.
3. Add controller context to html.
4. Set it up to alert something.
5. Set up a div that we can click and make the alert fire

Databinding Demo

1. **Start with DemoStep2**, complete will be like DemoStep3
2. Note that there is some extra stuff for prettiness - css, bootstrap, etc
3. Get rid of alert
4. Use input, div, and ng-model to show two-way binding.
5. Set up method in controller to change data
6. Remove all that sample stuff
7. Add model and model.expenses to scope in controller
8. In index.html, add h1 with 1View Dashboard just inside body
9. Add class=”dashboard” to controller div
10. Inside that, add div class=”expenses” ng-repeat=”expense in model.expenses”
11. Inside that, add div class=”expense”
12. Inside that, add divs for four expense props – category, date, description, amount (class for amount)

Routing Demo

1. Start with DemoStep3, complete will be like DemoStep4
2. Add ngRoute as dependency to app module
3. Pull data into var to be reused in two controllers
4. Add config section incl new expense detail route - may want to just copy in
5. Add new controller for expense detail - may want to just copy in
6. Add partials to index.html in script tags - may want to just copy in
7. Add div with ng-view
8. In index.html, add script ref for routing and jquery ref for grep

Services Demo

1. Start with DemoStep4, complete will be like DemoStep5
2. Note added nodejs/express stuff
3. Show data coming from http://localhost:3001/expenses
4. Get rid of data in app.js
5. Change controllers to use $http.get
6. Pull partials out of index.html, into their own files in partials folder
7. ?? Consider creating a factory that wraps data access ??

Directives Demo

1. Start with DemoStep5, complete will be like DemoStep6
2. Modify data so it includes provider - may want to just copy in
3. Create to directive-templates folder and add the two templates
4. Add directives to app.js
5. Add provider markup, including new directives, to expense-detail.html

Filters Demo

1. **Start with DemoStep6,** complete will be like DemoStep7
2. Add date and currency built-in filters to expense-detail.html
3. Add searchtext input w/ toolbar to expenses.html - may want to just copy in
4. Add filter to repeater - | filter:searchText – can search both fields
5. Add category to model
6. Use orderBy (expression, use quotes)

Interceptors Demo

1. Start with DemoStep7, complete will be like DemoStep8
2. Introduce delay to server (expense.js) - setTimeout 1000ms
3. Add interceptor to app.js - may want to just copy in
4. Explain why the DOM manip should be in a directive
5. In expense-detail.html, add a new div before expenses div, id of loadingDiv, style="dispay: none;"
6. Give content div id of contentDiv

Testing Demo

1. Start with DemoStep8, complete will be like DemoCompleted

Wizard Demo (Transactions.aspx)

1. Show codebehind
2. Show some standard asp.net webforms stuff
3. Show controller wired up, mention ng-cloak
4. Explain ng-switch
5. Show loadingWidget
6. Go to SpinnerInterceptor.js
7. AccountsActivityTransactions
   1. Explain some dependencies – logic, data, ajax services
   2. Init, etc.
   3. WatchCollection of history, back, next
   4. Brief overview of logic