




# Using AngularJS In APEX

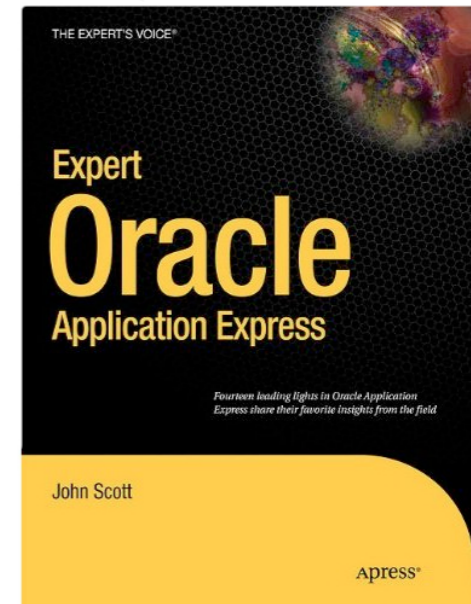
---

Dan McGhan

*Senior Technical Consultant*

# My background

- Dan McGhan 
- Senior consultant with Enkitech
  - Joined in March 2013
  - [dmcghan@enkitech.com](mailto:dmcghan@enkitech.com)
- Co-Author, Expert APEX
- A.K.A “that guy that wrote that plug-in”
- JavaScript fanatic



# About Enkitec

- Oracle Platinum Partner
  - Established in 2004
  - Headquartered in Dallas, TX
  - Locations throughout the US & EMEA
- Specialties include
  - Exadata Implementations
  - Development Services
    - PL/SQL / Java / APEX
  - DBA/Data Warehouse/RAC
  - Business Intelligence



# Hammer syndrom

“I call it the law of the instrument, and it may be formulated as follows: Give a small boy a hammer, and he will find that everything he encounters needs pounding”

~Abraham Kaplan

# New Year's Resolutions

- Get MEAN
  - Mongo, Express, Angular, Node
- Many other supporting technologies
  - Git, Grunt, Bower, Bootstrap, Yeoman, and many more

# What is Angular?

- A JavaScript MVC framework
  - Others include Backbone, Ember, & Knockout
- Created by and maintained by Google
- Typically used to build SPAs
- Requires a “new” way of thinking

# Thinking with MVC

“Developing a UI? think state 1st, actions 2nd. Model the UI's state as observable. Then, make actions that update state.”

“Everything else calls actions to update state, and listens to state changes. Click -> action > state change -> update UI”

~Justin B. Meyer

# Why use Angular in APEX?

- Your boss wants a “Todo” app
- Seems simple enough, APEX to the rescue!
- 10 minutes later you deliver the app



# But your boss wants more

- Dashboard w/Today, Tomorrow, & Future columns
- Inline edit ability w/auto filtering & sorting
- Drag & drop functionality
- Custom validations & notifications
- Who knows what else!

# Stage 1: Mock it up

- Download and include Bootstrap
  - Add to /c/todos/vendor
- Include Bootstrap JS & CSS in page
- Create HTML region w/HTML & CSS

# Stage 1: Mock it up

JavaScript
<p>File URLs</p> <input type="text" value="/c/todos/vendor/bootstrap-3.1.1/js/bootstrap.min.js"/>
<p>Function and Global Variable Declaration</p> <input type="text"/>
<p>Execute when Page Loads</p> <input type="text"/>
CSS
<p>File URLs</p> <input type="text" value="/c/todos/vendor/bootstrap-3.1.1/css/bootstrap.min.css"/>

# Stage 1: Mock it up

### Identification

Page: 3 Todos


\* Title:   exclude title from translation

Type:

### Source



Region Source

```
.todo-due-date{
  font-weight: 500;
  margin-top: 5px;
}
</style>
<table id="todos-container">
  <tr>
    <td class="todos-column-container">
      <div class="todos-column">
        <h3 class="todos-column-header">Today
          <button type="button" class="btn btn-default btn-xs">
            <span class="glyphicon glyphicon-plus"></span>
          </button>
```



12

# Stage 1: Mock it up

Today <span>+</span>	Tomorrow <span>+</span>	Future <span>+</span>
<p data-bbox="312 666 825 748">All Active Completed</p> <div data-bbox="194 813 943 1030"><p data-bbox="249 846 515 891">This is a title</p><p data-bbox="203 901 532 938"><input type="checkbox"/> This is a description</p><p data-bbox="249 954 592 987">2014-03-20 at 13:00:00</p><div data-bbox="834 833 935 1003"> </div></div>	<p data-bbox="1114 666 1627 748">All Active Completed</p>	<p data-bbox="1915 666 2428 748">All Active Completed</p>

# Stage 2: Get Angular working

- Download & include Angular
  - Add to /c/todos/vendor
- Create “todos” module
- Move HTML to todos.tpl.html
- Move CSS to todos.css
- Include content via ng-app & ng-include

# Stage 2: Get Angular working

## Download AngularJS

**Branch**  1.2.x (legacy)  1.3.x (latest) [?](#)

**Build**  Minified  Uncompressed  Zip [?](#)

**CDN**  [?](#)

**Bower**  [?](#)

**Extras** [Browse additional modules](#)

[Previous Versions](#)

[Download](#)

# Stage 2: Get Angular working

## Source

Region Source

```
<div ng-app="todos" ng-include="'/c/todos/src/todos.tpl.html'"></div>
```



# What is a directive?

- Extend HTML w/custom attributes and elements
  - Adds some web component functionality today
- Angular includes many directives by default
- Developers can create their own directives

# Stage 2: Get Angular working

```
Source  
Region Source  
<div ng-app="todos" ng-include="/c/todos/src/todos.tpl.html"></div>
```

app.js

```
1 angular.module('todos', []);
```

# Stage 2: Get Angular working

## JavaScript

### File URLs

```
/c/todos/vendor/bootstrap-3.1.1/js/bootstrap.min.js  
/c/todos/vendor/angular-1.2.14/angular.min.js  
/c/todos/src/app.js
```

### Function and Global Variable Declaration

### Execute when Page Loads



## CSS

### File URLs

```
/c/todos/vendor/bootstrap-3.1.1/css/bootstrap.min.css  
/c/todos/src/todos.css
```



# Stage 2: Get Angular working

Today +	Tomorrow +	Future +
<p>All Active Completed</p> <p><input type="checkbox"/> This is a title This is a description 2014-03-20 at 13:00:00</p> <p> </p>	<p>All Active Completed</p>	<p>All Active Completed</p>

# Stage 3: Get filter buttons working

- Create todos-controller.js
- Link controller to template via ng-controller
- Use ng-class and ng-click to get buttons working

# What is a controller?

- The business logic behind views
- Typically used for two things
  - Initialize scope
  - Handle interaction with the user

# What is scope?

- The glue between the model and the view

```
function PersonController($scope){  
  $scope.person = {  
    name: "Dan McGhan",  
    mood: "disappointed"  
    hair: "gone"  
  };  
}
```

+

```
<div ng-controller="PersonController">  
  {{person.name}} is {{person.mood}} because his hair is {{person.hair}}  
</div>
```

=

Dan McGhan is disappointed because his hair is gone



# Stage 3: Get filter buttons working

## todos-controller.js

```
1 angular.module('todos')
2   .controller('todosController', function($scope) {
3     $scope.todayStatus = 'active';
4     $scope.tomorrowStatus = 'active';
5     $scope.futureStatus = 'active';
6   });
```

## todos.tpl.html

```
1 <table id="todos-container" ng-controller="todosController">
2   <tr>
3     <td class="todos-column-container">
4     <div class="todos-column">
```



# Stage 3: Get filter buttons working

## todos.tpl.html

```
<div class="todos-column-button-container">
  <div class="btn-group">
    <button type="button" class="btn btn-default" ng-class="{active: todayStatus === 'all'}" ng-click="todayStatus = 'all'">All</button>
    <button type="button" class="btn btn-default" ng-class="{active: todayStatus === 'active'}" ng-click="todayStatus = 'active'">Active</button>
    <button type="button" class="btn btn-default" ng-class="{active: todayStatus === 'completed'}" ng-click="todayStatus = 'completed'">Completed</button>
  </div>
</div>
```

# Stage 3: Get filter buttons working

The screenshot displays a task management interface with three columns: Today, Tomorrow, and Future. Each column has a header with a plus sign and a filter bar with 'All', 'Active', and 'Completed' buttons. The 'Active' button is selected in all three columns. The 'Today' column contains a task card with the following details:

- Title: This is a title
- Description:  This is a description
- Timestamp: 2014-03-20 at 13:00:00
- Actions: Edit (wrench icon) and Delete (trash icon)

# Stage 4: Get checkbox working

- Create PL/SQL procedure to generate JSON
- Create a service to fetch the data
- Use module config to override Ajax POST
- Expose data from service to template
- Use ng-repeat to iterate over data
- Create the PL/SQL process to save state
- Add the save completed method to the service
- Add ng-change directive

# Stage 4: Get/set real data

## GET\_TODOS\_JSON

```
Source
* Process [Download Source]
DECLARE
    l_sql VARCHAR2(4000);
BEGIN
    l_sql :=
        'SELECT id AS "id", ' ||
        ' title AS "title", ' ||
        ' description AS "description", ' ||
        ' TO_CHAR(due_date, ''YYYY-MM-DD"T"HH24:MI:SS"Z"''') AS "dueDate", ' ||
        ' complete AS "completed" ' ||
        'FROM todos ORDER BY due_date';

    apex_util.json_from_sql (l_sql);
END;
```

# What is a service?

- Reusable business logic independent of views
- Can be created different ways
  - value
  - factory
  - service
  - constant
  - provider

# Stage 4: Get/set real data

## todos-service.js

```
angular.module('todos')
  .factory('todosService', function($http) {
    var factory = {};

    factory.todos = {};

    factory.fetchData = function() {
      var httpOptions = {
        method: 'POST',
        url: 'wv_flow.show',
        data: {
          p_flow_id: angular.element('#pFlowId').val(),
          p_flow_step_id: angular.element('#pFlowStepId').val(),
          p_instance: angular.element('#pInstance').val(),
          p_request: 'APPLICATION_PROCESS=GET_TODOS_JSON'
        }
      };

      $http(httpOptions)
        .success(function(data) {
          factory.todos = data.row;
        });
    };

    factory.fetchData();

    return factory;
  });
```



# Stage 4: Get/set real data

## app.js

```
angular.module('todos', [])
  .config(function($httpProvider) {
    //Configure the $httpProvider so that POSTs work more like jQuery
    //Solution found here:
    http://victorblog.com/2012/12/20/make-angularjs-http-service-behave-like-jquery-ajax/
    // Use x-www-form-urlencoded Content-Type
    $httpProvider.defaults.headers.post['Content-Type']
      = 'application/x-www-form-urlencoded;charset=utf-8';

    /**
     * The workhorse; converts an object to x-www-form-urlencoded serialization.
     * @param {Object} obj
     * @return {String}
     */
    var param = function(obj) {
      ...
    };

    // Override $http service's default transformRequest
    $httpProvider.defaults.transformRequest = [function(data) {
      return angular.isObject(data)
        && String(data) !== '[object File]' ? param(data) : data;
    }];
  });
```

# Stage 4: Get/set real data

## todos-controller.js

```
angular.module('todos')
  .controller('todosController', function($scope, todosService) {
    $scope.todayStatus = 'active';
    $scope.tomorrowStatus = 'active';
    $scope.futureStatus = 'active';

    $scope.todosService = todosService;
  });
```



# Stage 4: Get/set real data

## todos.tpl.html

```
<li ng-repeat="todo in todosService.todos">
  <div class="todo-container">
    <table>
      <tr>
        <td>
          <input type="checkbox" ng-model="todo.completed" ng-true-value="Y" ng-false-value="N" />
        </td>
        <td>
          <div class="todo-title">{{todo.title}}</div>
          <div>{{todo.description}}</div>
          <div class="todo-due-date">{{todo.dueDate | date:'yyyy-MM-dd \'at\' hh:mm:ss'}}</div>
        </td>
        <td>
          <div class="btn-group-vertical">
            <button type="button" class="btn btn-default">
              <span class="glyphicon glyphicon-wrench"></span>
            </button>
            <button type="button" class="btn btn-default">
              <span class="glyphicon glyphicon-trash"></span>
            </button>
          </div>
        </td>
      </tr>
    </table>
  </div>
</li>
```

# Stage 4: Get/set real data

## SAVE\_COMPLETED

```
Source
* Process [Download Source]
DECLARE
    l_id      TODOS.ID%TYPE;
    l_complete TODOS.COMPLETE%TYPE;
BEGIN
    l_id := apex_application.g_x01;
    l_complete := apex_application.g_x02;

    UPDATE todos
    SET complete = l_complete
    WHERE id = l_id;

    htp.prn('{"success": true}');
END;
```

# Stage 4: Get/set real data

## todos-service.js

```
factory.fetchData();

factory.saveCompleted = function(todo) {
  var httpOptions = {
    method: 'POST',
    url: 'wwv_flow.show',
    data: {
      p_flow_id: angular.element('#pFlowId').val(),
      p_flow_step_id: angular.element('#pFlowStepId').val(),
      p_instance: angular.element('#pInstance').val(),
      p_request: 'APPLICATION_PROCESS=SAVE_COMPLETED',
      x01: todo.id,
      x02: todo.completed
    }
  };

  $http(httpOptions)
    .success(function(data) {
      if (!data.success) {
        todo.completed = (todo.completed === 'Y' ? 'N' : 'Y');
      }
    });
}

return factory;
```

# Stage 4: Get/set real data

## todos.tpl.html

```
<li ng-repeat="todo in todosService.todos">
  <div class="todo-container">
    <table>
      <tr>
        <td>
          <input type="checkbox" ng-model="todo.completed" ng-true-value="Y" ng-false-value="N"
            ng-change="todosService.saveCompleted(todo) " />
        </td>
        <td>
          <div class="todo-title">{{todo.title}}</div>
          <div>{{todo.description}}</div>
          <div class="todo-due-date">{{todo.dueDate | date:'yyyy-MM-dd \'at\' hh:mm:ss'}}</div>
        </td>
        <td>
          <div class="btn-group-vertical">
            <button type="button" class="btn btn-default">
              <span class="glyphicon glyphicon-wrench"></span>
            </button>
            <button type="button" class="btn btn-default">
              <span class="glyphicon glyphicon-trash"></span>
            </button>
          </div>
        </td>
      </tr>
    </table>
  </div>
</li>
```



# Stage 5: Add filters

- Add orderby filter
- Create custom filters for
  - Today
  - Tomorrow
  - Future

# What is a filter?

- Ummmm, a filter
  - But wait, there's more!
- Filters also sort and format
  - date
  - currency
  - uppercase
  - lowercase

# Stage 5: Add filters

## todos.tpl.html

```
<li ng-repeat="todo in todosService.todos | orderBy:'dueDate'">
  <div class="todo-container">
    <table>
      <tr>
```

# Stage 5: Add filters

## today-filter.js

```
angular.module('todos')
  .filter('todayFilter', function() {
    return function(todos, todayStatus) {
      var filteredTodos = [],
          today = new Date();

      today.setHours(0,0,0,0);

      if (todayStatus === 'all') {
        for (var x = 0; x < todos.length; x++) {
          if (new Date(todos[x].dueDate).setHours(0,0,0,0) <= today) {
            filteredTodos.push(todos[x]);
          }
        }
      } else if (todayStatus === 'active') {
        for (var x = 0; x < todos.length; x++) {
          if (todos[x].completed === 'N'
              && new Date(todos[x].dueDate).setHours(0,0,0,0) <= today
          ) {
            filteredTodos.push(todos[x]);
          }
        }
      } else if (todayStatus === 'completed') {
        ...
      }

      return filteredTodos;
    };
  });
```



# Stage 5: Add filters

## todos.tpl.html

```
<li ng-repeat="todo in todosService.todos | todayFilter:todayStatus | orderBy:'dueDate'">
  <div class="todo-container">
    <table>
      <tr>
```

# Stage 6: Move todos to a directive

- Move todo HTML to todo.tpl.html
- Create todo-directive.js

# Stage 6: Move todos to a directive

## todo.tpl.html

```
<div class="todo-container">
  <table>
    <tr>
      <td>
        <input type="checkbox" ng-model="todo.completed" ng-true-value="Y" ng-false-value="N"
          ng-change="todosService.saveCompleted(todo)" />
      </td>
      <td>
        <div class="todo-title">{{todo.title}}</div>
        <div>{{todo.description}}</div>
        <div class="todo-due-date">{{todo.dueDate | date:'yyyy-MM-dd \'at\' HH:mm:ss'}}</div>
      </td>
      <td>
        <div class="btn-group-vertical">
          <button type="button" class="btn btn-default">
            <span class="glyphicon glyphicon-wrench"></span>
          </button>
          <button type="button" class="btn btn-default">
            <span class="glyphicon glyphicon-trash"></span>
          </button>
        </div>
      </td>
    </tr>
  </table>
</div>
```

# Stage 6: Move todos to a directive

## todo-directive.js

```
angular.module('todos')
  .directive('todo', function() {
    return {
      restrict: "EA",
      templateUrl: '/c/todos/src/todo.tpl.html'
    };
  });
```

# Stage 6: Move todos to a directive

## todos.tpl.html

```
<li ng-repeat="todo in todosService.todos | todayFilter:todayStatus | orderBy:'dueDate'">  
  <div todo></div>  
</li>
```

or

```
<li ng-repeat="todo in todosService.todos | todayFilter:todayStatus | orderBy:'dueDate'">  
  <todo></todo>  
</li>
```

# Stage 7: Start edit functionality

- Create DELETE\_TODO process
- Add edit and remove methods to the service
- Update todo.tpl.html to handle edit “state”
- Create todo.css
- Edit todo directive to use datepicker/timepicker



# Stage 7: Start edit functionality

## DELETE\_TODO

```
Source
* Process [Download Source]
DECLARE
    l_id TODOS.ID%TYPE;
BEGIN
    l_id := apex_application.g_x01;

    DELETE FROM todos
    WHERE id = l_id;
END;
```



# Stage 7: Start edit functionality

## todos-service.js

```
factory.edit = function(todo) {  
  todo.tempTitle = todo.title;  
  todo.tempDescription = todo.description;  
  todo.tempDueDate = $filter('date')(todo.dueDate, 'yyyy-MM-dd');  
  todo.tempDueTime = $filter('date')(todo.dueDate, 'HH:mm');  
  todo.editMode = true;  
};
```

# Stage 7: Start edit functionality

## todos-service.js

```
factory.remove = function(todo) {  
    var httpOptions = {  
        method: 'POST',  
        url: 'wwv_flow.show',  
        data: {  
            p_flow_id: angular.element('#pFlowId').val(),  
            p_flow_step_id: angular.element('#pFlowStepId').val(),  
            p_instance: angular.element('#pInstance').val(),  
            p_request: 'APPLICATION_PROCESS=DELETE_TODO',  
            x01: todo.id  
        }  
    };  
  
    $http(httpOptions)  
        .success(function(data) {  
            for (var x = 0; x < factory.todos.length; x++) {  
                if (factory.todos[x].id === todo.id) {  
                    factory.todos.splice(x, 1);  
                    break;  
                }  
            }  
        });  
};
```

# Stage 7: Start edit functionality

## todo.tpl.html

```
<div class="todo-container">
  <table ng-show="!todo.editMode">
    ...
  </table>
  <table ng-show="todo.editMode">
    <tr>
      <td>&nbsp;</td>
      <td>
        <input class="temp-title" type="text" ng-model="todo.tempTitle" /><br />
        <textarea class="temp-description" ng-model="todo.tempDescription"></textarea><br />
        <input class="temp-due-date" type="text" ng-model="todo.tempDueDate" />
        <input class="temp-due-time" type="text" ng-model="todo.tempDueTime" />
      </td>
      <td>
        <div class="btn-group-vertical">
          <button type="button" class="btn btn-default">
            <span class="glyphicon glyphicon-floppy-disk"></span>
          </button>
          <button type="button" class="btn btn-default">
            <span class="glyphicon glyphicon-repeat"></span>
          </button>
        </div>
      </td>
    </tr>
  </table>
</div>
```

# Stage 7: Start edit functionality

## todo.css

```
.todo-container {  
  border: 1px solid #aaa;  
  border-radius: 3px;  
  margin-top: 3px;  
  padding: 0 3px 3px 3px;  
  background-color: #FFF  
}  
  
.todo-container > table {  
  width: 100%;  
  height: 80px;  
}  
  
.todo-container > table > tbody > tr > td:first-of-type {  
  width: 20px;  
  vertical-align: middle;  
}  
  
.todo-container > table > tbody > tr > td:nth-of-type(2) {  
  padding: 10px 5px 0 0;  
}  
  
...
```



# Stage 7: Start edit functionality

## todo-directive.js

```
angular.module('todos')
  .directive('todo', function() {
    var linker;

    linker = function(scope, element, attrs) {
      element.find('.temp-due-date').datepicker({'dateFormat': 'yy-mm-dd'});
      element.find('.temp-due-time').timepicker({'timeFormat': 'H:i'});
    };

    return {
      restrict: "EA",
      link: linker,
      templateUrl: '/c/todos/src/todo.tpl.html'
    };
  });
```

# Stage 8: Finish edit functionality

- Add the UPDATE\_TODO process
- Add save and discardChanges methods to service
- Update the template to call the methods

# Stage 8: Finish edit functionality

## UPDATE\_TODO

```
Source
* Process [Download Source]
DECLARE
    l_id          TODOS.ID%TYPE;
    l_title       TODOS.TITLE%TYPE;
    l_description TODOS.DESCRPTION%TYPE;
    l_due_date    TODOS.DUE_DATE%TYPE;

BEGIN
    l_id := apex_application.g_x01;
    l_title := apex_application.g_x02;
    l_description := apex_application.g_x03;
    l_due_date := TO_DATE(apex_application.g_x04, 'YYYY-MM-DD HH24:MI');

    UPDATE todos
    SET title = l_title,
        description = l_description,
        due_date = l_due_date
    WHERE id = l_id;

END;
```



# Stage 8: Finish edit functionality

## todo-service.js

```
factory.discardChanges = function(todo) {  
    .....  
    todo.editMode = false;  
};
```

# Stage 8: Finish edit functionality

## todo-service.js

```
factory.saveChanges = function(todo) {
    var httpOptions = {
        method: 'POST',
        url: 'wv_flow.show',
        data: {
            p_flow_id: angular.element('#pFlowId').val(),
            p_flow_step_id: angular.element('#pFlowStepId').val(),
            p_instance: angular.element('#pInstance').val(),
            p_request: 'APPLICATION_PROCESS=UPDATE_TODO',
            x01: todo.id,
            x02: todo.tempTitle,
            x03: todo.tempDescription,
            x04: todo.tempDueDate + ' ' + todo.tempDueTime
        }
    };

    $http(httpOptions)
        .success(function(data) {
            todo.title = todo.tempTitle;
            todo.description = todo.tempDescription;
            todo.dueDate = todo.tempDueDate + 'T' + todo.tempDueTime + ':00';
            todo.editMode = false;
        });
};
```

# Stage 8: Finish edit functionality

## todo.tpl.html

```
<div class="btn-group-vertical">  
  <button ng-click="todosService.saveChanges(todo)" type="button" class="btn btn-default">  
    <span class="glyphicon glyphicon-floppy-disk"></span>  
  </button>  
  <button ng-click="todosService.discardChanges(todo)" type="button" class="btn btn-default">  
    <span class="glyphicon glyphicon-repeat"></span>  
  </button>  
</div>
```

# Stage 9: Add “new” ability

- Add “add” method to service
- Add the CREATE\_TODO process
- Update “change” method to accommodate
- Update template to call the method

# Stage 9: Add “new” ability

## todo-service.js

```
factory.addTodo = function(column) {  
  var todo = {};  
  
  todo.id = 0;  
  todo.title = '';  
  todo.description = '';  
  todo.completed = 'N';  
  
  if (column === 'today') {  
    todo.dueDate = new Date().toISOString();  
  } else if (column === 'tomorrow') {  
    todo.dueDate = new Date(new Date().getTime() + 24 * 60 * 60 * 1000).toISOString();  
  } else if (column === 'future') {  
    todo.dueDate = new Date(new Date().getTime() + 48 * 60 * 60 * 1000).toISOString();  
  }  
  
  factory.todos.unshift(todo);  
  
  factory.edit(todo);  
};
```



# Stage 9: Add “new” ability

## CREATE\_TODO

### Source

\* Process [Download Source]

```
DECLARE

  l_id          TODOS.ID%TYPE;
  l_title       TODOS.TITLE%TYPE;
  l_description TODOS.DESCRPTION%TYPE;
  l_due_date    TODOS.DUE_DATE%TYPE;

BEGIN

  l_title := apex_application.g_x02;
  l_description := apex_application.g_x03;
  l_due_date := TO_DATE(apex_application.g_x04, 'YYYY-MM-DD HH24:MI');

  INSERT INTO todos (
    title, description, due_date
  ) VALUES (
    l_title, l_description, l_due_date
  )
  RETURNING id INTO l_id;

  http.prn('{"id": ' || l_id || '}');

END;
```

# Stage 9: Add “new” ability

## todo-service.js

```
factory.saveChanges = function(todo) {  
  var httpOptions = {  
    method: 'POST',  
    url: 'wv_flow.show',  
    data: {  
      p_flow_id: angular.element('#pFlowId').val(),  
      p_flow_step_id: angular.element('#pFlowStepId').val(),  
      p_instance: angular.element('#pInstance').val(),  
      p_request: 'APPLICATION_PROCESS=' + ((todo.id === 0) ? 'CREATE_TODO' : 'UPDATE_TODO'),  
      x01: todo.id,  
      x02: todo.tempTitle,  
      x03: todo.tempDescription,  
      x04: todo.tempDueDate + ' ' + todo.tempDueTime  
    }  
  };  
  
  $http(httpOptions)  
    .success(function(data) {  
      if (todo.id === 0) {  
        todo.id = data.id;  
      }  
  
      todo.title = todo.tempTitle;  
      todo.description = todo.tempDescription;  
      todo.dueDate = todo.tempDueDate + 'T' + todo.tempDueTime + ':00';  
      todo.editMode = false;  
    });  
};
```



# Stage 9: Add “new” ability

## todo-service.js

```
factory.discardChanges = function(todo) {  
  if (todo.id === 0) {  
    factory.remove(todo);  
  } else {  
    todo.editMode = false;  
  }  
};
```

# Summary

- Angular is complex
  - Lots of new terms
  - Steep learning curve
- But it's also very powerful
  - Feature rich
  - Built for testing (yes, you should be testing)
- Want to learn more?
  - Google: angularjs in 60ish minutes



<http://www.enkitec.com>