UI-Router's Resolve

Consider the following …

A DEPENDENCY FAILS TO LOAD AND THE USER GET A BROKEN PAGE.

A dependency loads slowly and the page fills in.

These cause a poor user experience.

Resolve

Like Promise.all for UI-Router.

You can use resolve to provide your controller with content or data that is custom to the state.

RESOLVE IS AN OPTIONAL MAP OF DEPENDENCIES WHICH SHOULD BE INJECTED INTO THE CONTROLLER.

If any of these dependencies are promises, they will be resolved and converted to a value before the controller is instantiated and the $stateChangeSuccess event is fired.

```
$stateProvider
  .state("students", {
  url : "/students",
  templateUrl: 'students.html',
  controller : 'studentsController',
  resolve: {

  }
});
```

```
resolve {
  // Not a promise, resolves immediately
  simpleObj: function() {
    return {value: 'simple!'};
  },
  // You need to inject any services that you are using.
  // Returns a promise, will resolve when promise is resolved.
  students1: function($http) {
    return $http({method: 'GET', url: '/api/students'});
  }
}
```

```
resolve{
  students2:  function($http){
    return $http({method: 'GET', url: '/someUrl'})
      .then (function (data) {
      return doSomeStuffFirst(data);
    });
  },
}
```

```
resolve {
  studentsService: 'studentsService'
}
```

```
studentsController(simpleObj,students,students2,studentInfo) {
  vm = this;
  vm.message = simpleObj;
  vm.students = students.data;
  vm.students = students;

  vm.studentInfo = studentsService.getInfo();
}
```

```
greeting: function($q, $timeout){
    var deferred = $q.defer();
    $timeout(function() {
        deferred.resolve('Hello!');
    }, 1000);
    return deferred.promise;
}
```

Questions?