Angular-UI Router Changes

Draft

Problems

Angular-UI is a wonderful tool, allowing us to express nested states. However, we consistently feel constrained, in each project, by the following limitations:

1. All route parameters must be specified in a route
    a. Say you have the route ‘`/`’, which lists all articles in a blog for the latest month with articles. Clicking a “older entries” button might take you to “`/:month/:year`”. It would be nice if the same controller and route state could be used to fulfill both conditions.
2. All route parameter changes must trigger a reinstantiation of the controller, which consequently regenerates the view DOM

Proposed Solutions

Default / Optional Parameters

It’s proposed that we allow for specification of multiple matching routes, such as:

```
ui-router
  .state "home.movies",
    url: [
      "/",
      "/:postalCode",
      "/:postalCode/",
      "/:postalCode/:dayTime"
    ]
    # Defaults only pulled, by dependency, for missing each variable
    # in the union of all variables in route list.
    # Expects a promise to be returned, or {value:...} for
  # synchronous resolution
  defaults:
   locationResult: ->
    $http.get("/api/get-defaults") # remote call which looks up location by geo-ip and returns local time information.
   postalCode: (locationResult) ->
    { value: locationResult.postalCode }
   dayTime: (locationResult) ->
    { value: if (locationResult.localHour > 16) "evening" else "matinee" }
```

Dynamic Parameters

Signals Angular-ui to pass in parameters as searchParams, or a similar object. When the values change, expect controller bindings to respond via mutation observers; do not instantiate the controller.

```
ui-router
 .state "home.movies",
  url: [
   "/",
   "/:location",
   "/:location/",
   "/:location/:dayTime"]
  dynamic: ["dayTime"]
```