

STARRYING
SHARP with
Angular.js



Services Level 3

Provider Section 3

Current Gravatar Factory

We used a factory to generate our Gravatar URL.

services/gravatar.js

```
angular.module("NoteWrangler")
.factory("Gravatar", function GravatarFactory() {
  var avatarSize = 80; // Default size
  var avatarUrl = "http://www.gravatar.com/avatar/";
  return function(email){
    return avatarUrl + CryptoJS.MD5(email) + "?size=" + avatarSize.toString();
  };
});
```

The problem with this is our avatarSize is hard coded and not configurable.

The solution is simple: Use the Provider recipe instead!

Provider Service Recipe

The solution is simple: Use the Provider recipe instead!

- Factory

Used for sharing functions and objects across an application.

- Provider

Shares methods and objects (like a Factory), but allows for configuration.

```
.provider("<Service Name>", function <Service Name>Provider() {  
  this.$get <function or object of functions>  
});
```

All providers define a \$get function

Provider Service Recipe

The \$get function is where you return an object and inject services.

```
.provider("example", function exampleProvider() {  
  this.$get = function($http){} ;  
});
```

Providers run before everything else, so the only thing you can inject into them is other providers.

```
.provider("example", function exampleProvider(otherProvider) {  
  this.$get <function or object of functions>  
});
```

Provider Service Recipe

In order to turn our Gravatar service into a provider, we need to define a \$get function.

services/gravatar.js

```
angular.module("NoteWrangler")
.provider("Gravatar", function GravatarProvider() {
  var avatarSize = 80; // Default size
  var avatarUrl = "http://www.gravatar.com/avatar/";
  this.$get = function() {
    return function(email){
      return avatarUrl + CryptoJS.MD5(email) + "?size=" + avatarSize.toString();
    }
  };
}) ;
```

🎮 Calling the Provider Service in a Controller

Calling the provider is just like calling the factory.

📄 users-index-controller.js

```
angular.module("NoteWrangler")
.controller("UserIndexController", function($scope, Gravatar) {
  $scope.gravatarUrl = function(user) {
    return Gravatar(user.email);
  }
});
```



But how do we do the configuration bit?

Setting Configurable Items

We need to define a set method for each item we want to be configurable.

services/gravatar.js

```
angular.module("NoteWrangler")
.provider("Gravatar", function GravatarProvider() {
  var avatarSize = 80; // Default size
  var avatarUrl = "http://www.gravatar.com/avatar/";

  this.setSize = function(size) {
    avatarSize = size;
  }

  this.$get = function(){...}
});
```

Calling the Configuration Method

We set these configurable items in our app.js file that will run before everything else.

app.js

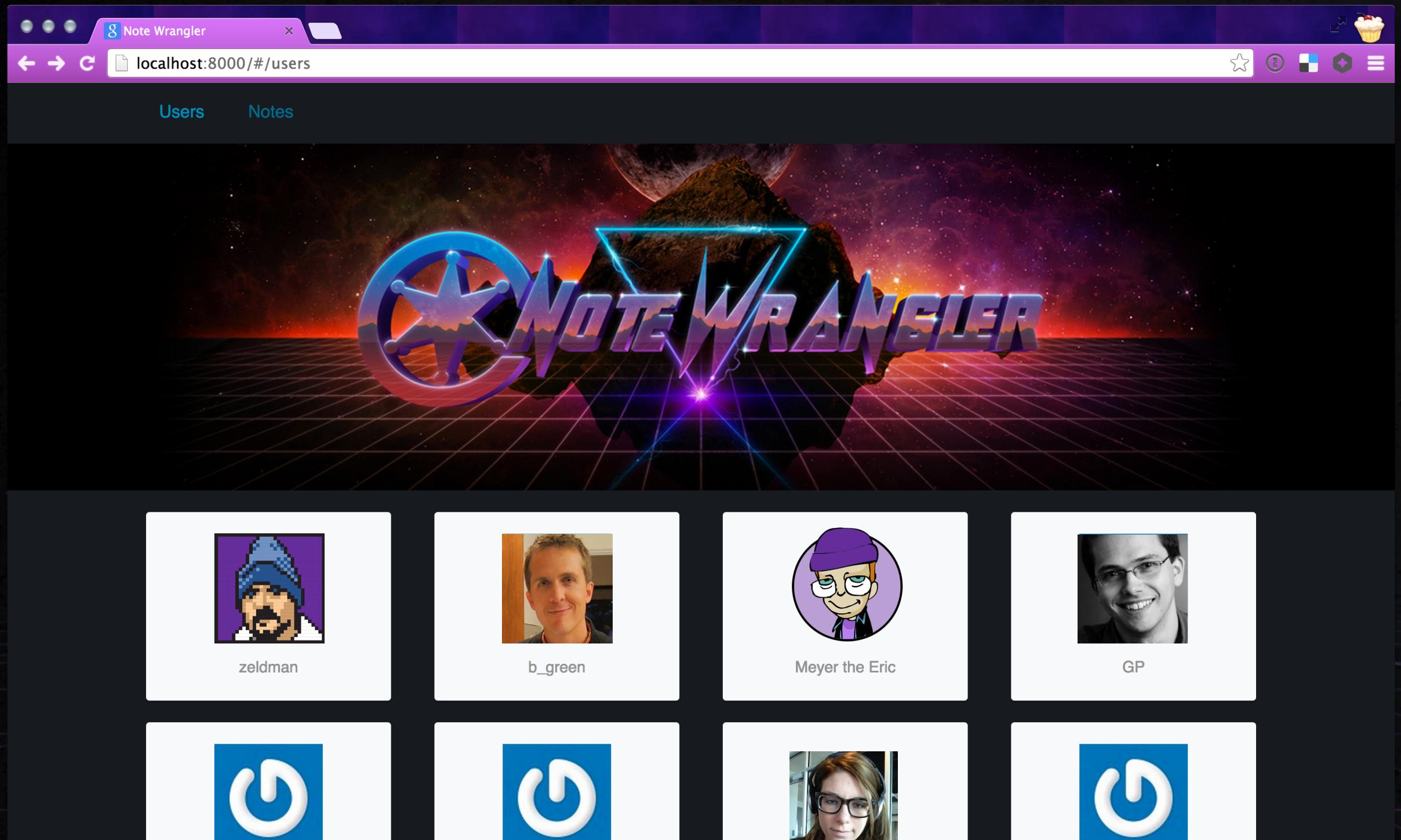
```
angular.module("NoteWrangler", ["ngRoute", "Gravatar"])
.config(function (GravatarProvider) {
  GravatarProvider.setSize(100);
});
```

services/gravatar.js

```
this.setSize = function(size) {
  avatarSize = size;
}
...
```

Notice we pass in
"GravatarProvider"
- not just "Gravatar"

Gravatar Provider Works



STAYING SHARP with Angular.js