5.1 Dependencies

5.2 Refactoring into a Module

200 PTS

After that we will need a way to encapsulate our directives in order to give our `app` access to them. We can use a module to do this! It is time for Module inception! There is a new js file provided for you — `products.js`; extract all store directives(descriptions, specs, reviews, title, gallery, and tabs) and paste them inside this new file. Then create a new module that we will make our original `gemStore` module require as a dependency.

Help Me

Task 1/4

Create a new Module named `store-directives` to encapsulate our store Directives.

```
(function() {
  var app = angular.module("store-directives",[]);
})();

```

Task 2/4

Move the Directive definitions from `app.js` into `products.js`.

```
(function() {
  var app = angular.module("store-directives",[]);
  app.directive("productDescription", function() {
    return {
      restrict: 'E',
      templateUrl: "product-description.html"
    };
  });

  app.directive("productReviews", function() {
    return {
      restrict: 'E',
      templateUrl: "product-reviews.html"
    };
  });
```

Task 3/4

Give `gemStore` Module access to the directives by adding a dependency to `gemStore`'s definition.

```
(function() {
  var app = angular.module('gemStore', ["store-directives"]);

```

Task 4/4

Link in the new `products.js` file.

```
    <script type="text/javascript" src="angu
    <script type="text/javascript" src="app.j
    <script src="products.js"></script>
  </head>
```

5.3. Services //slide

5.4 Built-in AngularJS Services

200 PTS

We can use the built-in `$http` Service to make requests to a server (or in our case a json file). Give our StoreController access to the products using a service.

Help Me

Task 1/5

Inject the `$http` service into our `StoreController`.

```
app.controller('StoreController', ["$http",function($http){
  var store = this;
  store.products = [];

}]);
```

Task 2/5

`get` the `store-products.json` using the `$http` Service.

```
var app = angular.module('gemStore', ['store-directives']);

  app.controller('StoreController', ["$http",function($http){
    var store = this;
    store.products = [];

    $http.get("/store-products.json");
  }]);

  app.controller('ReviewController', function() {…});
})();
```

Task 3/5

Attach a success to our get call.

```
  app.controller('StoreController', ["$http"
    var store = this;
    store.products = [];

    $http.get("/store-products.json")
    .success(function(){

    });
  }]);
```

Task 4/5

Name the first parameter of the success function `data`.

```
  $http.get("/store-products.json")
  .success(function(data){
```

Task 5/5

Give our StoreController access to the products by setting products equal to the data given to us with the http service's success promise.

```
(function() {
  var app = angular.module('gemStore', ['store-directives']);

  app.controller('StoreController', ["$http",function($http){
    var store = this;
    store.products = [];

    $http.get("/store-products.json")
    .success(function(data){
      store.products = data;
    });
  }]);

  app.controller('ReviewController', function() {…});
})();
```