Http

- Using the `Http` class, you can interact with API endpoints
- Http is available as an injectable class
- `Http` has a request method that returns an Observable which will emit a single Response when a response is received.
- You can inject `http` in the constructor of a class: `constructor(http: Http) {...}`

Getting Data from Server

In this section we are going to use the `http` class to get a list of students from a server by hitting `/api/students`

Project Files

The project files for this section are in angular2-intro/project-files/angular-examples/http/get-students

Getting Started

Before anything, let's add the `http.js` file from Angular's bundle. In your `index.html` file add the following to the head tag:

```
<script src="/node_modules/angular2/bundles/http.js"></script>
```

After that, we are going to make a service that handles getting data from the endpoint. We are going to call this `StudentSvc`:

```
@Injectable()
class StudentSvc {
  constructor(private http: Http) {} /* Inject Http */
  getStudents(): Observable<Response> {
    return this.http.get('/api/students');
  }
7 }
```

- On line 1, we are using the `Injectable` decorator to make our class injectable
- In the constructor we are injecting the `Http` service and making a reference to it in a private variable `http`
- The `getStudents` method makes a `GET` call to our local endpoint an returns an `Observable`

Now that we have the `StudentSvc` service, we can create a component and inject the `StudentSvc` to it:

```
@Component({
  selector: 'app',
  templateUrl :'templates/app.tpl.html',
  providers: [StudentSvc] // <- adding to the list of providers
})
```

In addition to the `StudentSvc`, we also need to add `HTTP_PROVIDERS` in the providers array:

```
@Component({
  selector: 'app',
  templateUrl :'templates/app.tpl.html',
  providers: [HTTP_PROVIDERS, StudentSvc] // <- adding `HTTP_PROVIDERS`
})
```

After adding the providers, we can define the component class:

```
@Component({...})
class HttpGetExample  {
  private name: string;
  private students: Observable<Response>;
  constructor (studentSvc: StudentSvc) {
    this.name = 'HTTP Get';
    studentSvc.getStudents().subscribe(resp => this.students = resp.json());
  }
}
```

If you notice, we are injecting the `StudentSvc` in the constructor and we are calling the `getStudents` method in the constructor. The `getStudents` returns an observable that we can subscribe to get the data out as they arrive. We also call the `json` method on each response to get the JSON data.

After getting the data, we can print the result in the view:

`app.tpl.html`

```
<h1>{{ name }}</h1>
<ul>
  <li *ngFor="#student of students">
    {{ student.name }}, {{ student.lastname }}
  </li>
</ul>
```

Here we are using the built-in `ngFor` directive to loop through the array of students and print their name and last name to the page.