**Developer home task**

**Website task**

1. Successfully connect the Website to the mock API to receive a green thumbs up, this should happen once both servers are running. See tips for mock API help.
2. Using the mock data from the json-server, list all the countries in the source with the existing models. Paginate if you think it is necessary.
3. Create a detailed view for each country using the existing model and the flags.

Some things we are looking for; effective component break down, good use of observables, examples of unit tests in the spec files.  
Please try not to use ready made components.

**Web Api Task**

1. Make a REST API endpoint which calls out to “<https://restcountries.com/v2/all>” and returns the result with the existing model.
2. Caching the data we return with a time to live of 1 minute.
3. Make a country detail REST API endpoint that can get a specific country's details using the existing model.  
   You can use any endpoints from “<https://restcountries.com/>” (V2 is the example above).

We’re looking for simple code, no need to over engineer, but use SOLID principles.  
Implement unit tests on the solution where you feel they are necessary.

**Get the code**

Unzip the file, there are two folders, web and api.

* Web API solution with swagger and some models
* Website code base has:
  + An empty “countries” component, not wired up to routing.
  + An empty “countries” api service
  + A working health api service plugged into the app component
  + Models based on requirements

**Tips**

* When installing npm packages for website(ensure @angular/cli and json-server are installed globally)
* Website - Angular13
  + “npm install” the packages
  + Website - In a terminal run “ng serve”
  + Mock API - in a terminal run “npm run api” - data and config in the “mock-api” folder - see the readme.
* Web API - Run the Paymentsense.Coding.Challenge.Api project.

