* Fetching Data
  + The Angular frontend is going to request information via HTTP requests to the .NET Controller backends
  + Requests and Responses
    - JSON Conventions and Defaults
      * Note that by default .NET Core MVC middleware will make several conversions to the output
      * E.g., properties will be converted from PascalCase to camalCase, and there will be no CRLF (single line)
  + A Very Long List
    - We are going to create a new component that fetches all the cities from the database
    - Note that in the component template (the HTML) attribute we use a new **[hidden] attribute** on the table of cities
    - The [hidden] Attribute
      * The **hidden** attribute is a valid HTML-5 sttribute that can be used to hide content from the user (similar to CSS **display: none**)
      * The square brackets around the attribute are Angular syntax for a property binding
      * This defines an HTML property or attribute that gets its value from the Component
      * Note this binding **only flows one way** (from the Component to the Template)
      * But, couldn’t we have instead used the \*ngIf directive to achieve the same thing?
      * Sort of, the difference is that \*ngIf will add or remove the element from the DOM; this means that the element and its children will be initialized and disposed of every time the conditional changes
      * The hidden attribute will only hide an element from the user; it will still be available in the DOM for manipulation (e.g. by JS)
      * We use the hidden attribute here because the table we have created will soon be a much more complex object