

Countries




In this exercise we'll create a simple app which will accept a country name as input from the user and retrieve some data about it from an API.

We will use the <https://restcountries.com> API as the data source, and JQuery to make the AJAX calls simpler to write.


We will use Git with branches in our workflow. Use the provided starter project, and follow the steps described below.



Use git and make meaningful commits.

Places which are marked with  are good places to make a commit, but you are welcome to experiment...

Basic Functionality

- Initialize a repo.
- Add an `<input>`, a `<button>` & an empty `<pre>` tag to the html file.
- Connect a script file.
- Add an event handler to the button. When clicked, it will get data about the country in the input and call a render function to render it into the empty `<pre>` tag.
- Commit your work 

Add a Country Service

- **Create a new branch (f-country-service)**
- Add another script with a simple service which provides a single function - `getCountryByName(name, cb)`
- The event handler will now use the service to fetch the data and pass it a pointer to the render function. ⬇
- Add a caching mechanism to the service which saves all query results to local storage. The cache map should look something like this:

```
{  
  Israel: { ...data from API call },  
  France: { ...data from API call },  
  Ireland: { ...data from API call },  
  ...  
}
```

- Return data about countries which have already been queried from the cache instead of the API ⬇

DOM Layout

- Checkout the the main branch
- **Create another branch (f-dom-layout)**
- Prepare an `<h2>` for the country name, an `` for the country's flag and two `<p>` tags for population and area.
- Refactor the render function to inject the correct data into these placeholders. ⬇
- Add some css to organize the display layout. ⬇

Merge

- Checkout the the main branch
- Merge f-country-service into main
- Merge main into f-dom-layout, resolve any conflicts.
- Merge f-dom-layout into main

Clear Cache

- **Create a new branch (f-clear-cache)**
- Add a button which will clear all cache contents 📌


Loader

- **Create a new branch (f-loader)**
- Add a loader which will be displayed when the data loads... you can test it by throttling the network in the dev tools. 📌

Merge

- Checkout the the main branch
- Merge f-clear-cache into main
- Merge main into f-loader, resolve any conflicts.
- Merge f-loader into main



Form

- **Create a new branch from main (f-form)**
- Refactor the UI to use a `<form>` element and connect the handler to the form `submit` event instead of the button `click` event. You will need to prevent the default form submit behavior with `event.preventDefault()`. 

Map

- **Create a new branch from main (f-map)**
- Add a link to the country's map 

Neighbors

- **Create a new branch from main (f-neighbors)**
- First add another function to the country service - `getCountryByCode(code, cb)`. Use the codes end point for this. 
- Add links to all of the country's neighboring countries. Use the [codes](#) end-point for this. 

Merge

- Checkout the the main branch
- Merge f-form into main
- Merge main into f-map, resolve any conflicts.
- Merge f-map into main
- Merge main into f-neighbors, resolve any conflicts.
- Merge f-neighbors into main

Fave Fun :-)

Lesson Materials

Link to a Step by Step Guide	Link to the presentation
	