JavaScript, DOM, AJAX, CORS and SVG

General part

- Explain about the Object Model, and why it's (very) relevant for modern Web-development
- Explain (using an example of your own choice) about JavaScript events, and Event Bubbling
- Elaborate on how JSON or XML supports communication between subsystems, even when the subsystems are implemented on different platforms.
- Explain the topic AJAX and how it has changed the way modern web-applications are created
- Explain the Same Origin Policy (for AJAX), and different ways to work around it

What if I get "stuck"

Note: Skim all the questions before you start. If you get stuck somewhere with the main question you can jump to the alternative question (not included in Exam Preparation Exercises). This could be enough to make you pass.

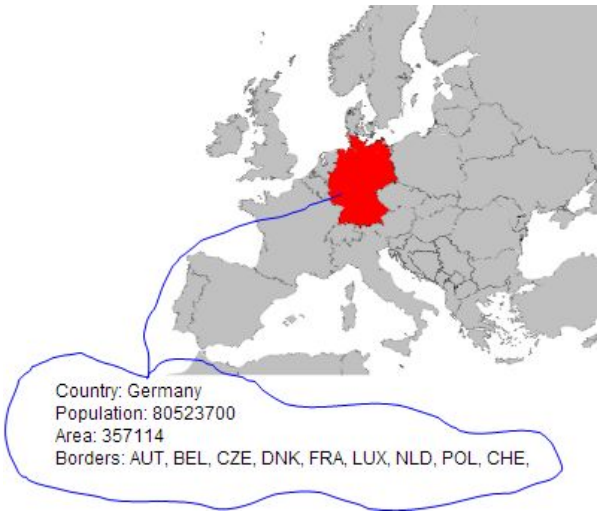
As an extra alternative if you get stuck with the practical part, you chose to demonstrate how you have used the main-topic for this exercise, in the semester project and/or in a CA.

If you select this option, you must first demonstrate how far you came with the practical part and the actual problem that made you select this option.

Practical part

In this exercise, we will combine SVG with several of the topics we have been around this semester such as AJAX and REST-endpoints to obtain data, and Javascript for DOM-manipulation.

The task is to create a web-page with a map of Europe which, when a country is selected with a mouse click, should highlight the country and print details about the country as sketched below.



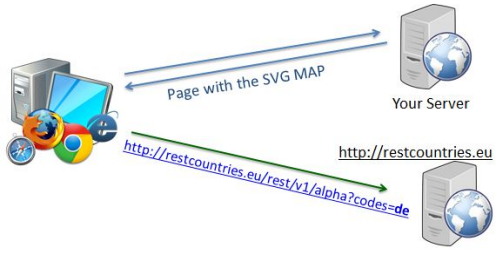
Get the map `Countries_Europe.svg` and copy it into the clipboard. Create a new web project, include an html-file and paste the content into the body.

This is an SVG image where each country is given the ISO-country code as the id. This is very convenient, because using the public REST API given here:
http://restcountries.eu/rest/v1/alpha?codes=**de**

you will obtain a JSON encapsulated data package with all the information needed (+ a lot more) to fill out the details (just test the link above in a Browser).

1. So the exercise boils down to. Hook up an event handler on the map, get the id, perform an AJAX request to fetch the JSON-data from the link given above and update the GUI using the JSON returned as sketched above.

2. For the previous task it was possible to obtain data right from *restcountries.eu* via an AJAX call made from within your Browser (as sketched to the right). Use Chrome Developer tools to explain (with focus on the Same Origin Policy) why this is possible.



3. Let's assume restcountries.eu had not allowed Cross Origin Calls.
   Design a Web Proxy Solution (using a plain Servlet or JAX-RS) where your browser will send the request to your proxy who should forward the request on to the remote server and send back the received response.

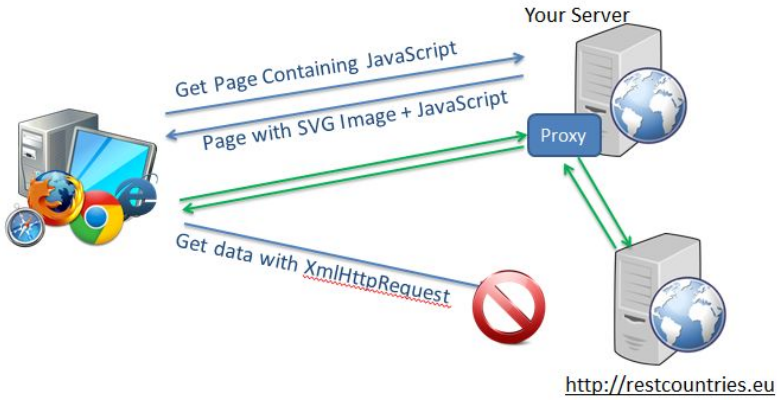
Note: This is not a part of the "real" exercise. It's meant as FYI and this section will not be included with a real question.

Hints:

1) Hook up a click-handler on the overall map (id =svg2), and in that, find the id for the actual element that was clicked (= the country code) via the `target` property of the event handler.
2) Change the colour of the selected country by changing its fill property
3)

***Same Origin Policy**: If your page, and the Service you try to call (via AJAX), are not located on the same Origin (see slides, and links on slides) the request is not permitted by the browser. The two solutions presented in the class (there are others, but focus on these) where either CORS, which is what made it possible for you to fetch data from within your browser, or to fetch the data from a Proxy on the **Origin** Server and provide it as a service from here .*

For part three you should use this second option as sketched in the figure above. Provide a Proxy-service on your



own (origin) server, and make an http request from this to the remote server. Just return the result you get from this call in your own REST service.

The code below shows a simple way to perform a programmatically HTTP-GET request (requesting JSON) in Java.

```
URL url = new URL("http://restcountries.eu/rest/v1/alpha...");
HttpURLConnection con = (HttpURLConnection)url.openConnection();
con.setRequestMethod("GET");
con.setRequestProperty("Accept", "application/json;charset=UTF-8");
Scanner scan = new Scanner(con.getInputStream());
String jsonStr=null;
if (scan.hasNext()) {
   jsonStr = scan.nextLine();
}
scan.close();
System.out.println(jsonStr);
```