Web — Master 1 IFI
Lab Session #5: AJAX

Andrea G. B. Tettamanzi
Université Côte d'Azur
andrea.tettamanzi@univ-cotedazur.fr

Academic Year 2019/2020

Abstract

The goals of this lab session is to practice the AJAX type of interaction between a Web client script and a server script.

1 Introduction

We will modify the CGI program developed for Lab Session #3 to return JSON content instead of an HTML page. Notice that, according to the RFC 4627, the MIME media type for JSON text is `application/json`. Its default encoding is UTF-8. On the client-side, we will modify the simple HTML page developed in the same lab session to use AJAX, instead of form submission, to request a prime factor decomposition from the server script and display it without reloading the page.

2 Requirement

1. Modify your CGI program, developed in Lab Session #3, to return a JSON object containing a prime factor decomposition according to the following specifications:

   - The natural number whose decomposition is requested will be assigned to the key `n`;
   - The prime factors will be returned in an array/list assigned to the key `prime_factors`;
   - Each element of the `prime_factors` array/list will be an object consisting of two key-value pairs: `prime`, the prime factor, and `power`, its power (> 0) in the decomposition.

   For instance, the JSON object for $n = 1234567890$ should be

   ```
   {
     "n": "1234567890",
     "prime_factors": [
       { "prime": "2", "power": "1" },
       { "prime": "3", "power": "2" },
       { "prime": "5", "power": "1" },
       { "prime": "3607", "power": "1" },
       { "prime": "3803", "power": "1" }
     ]
   }
   ```

2. Modify the HTML page developed in Lab Session #3 to include an HTML element to hold the computed decomposition and assign it a unique `id`, so that your JavaScript code will be able to get a reference to it.

3. Change the opening tag of the `form` element in that page as follows:

   ```
   <form method="get" onsubmit="return sendAjaxRequest(this);">
   ```

   It is important that the `sendAjaxRequest` function that you will define always returns `false`: this will prevent the form from actually being submitted. The form will still be there, but, this time, it will be your function that will take control when the user clicks on the "submit" button.

4. Implement the `sendAjaxRequest` function to create an AJAX request and send it to the CGI script.

5. Write also the callback method that will be in charge of receiving the answer and use it to insert the properly formatted decomposition in the HTML element previously identified for that purpose. You can call this function `getDecomposition`.

Notice that, this time, it will be the client the one that formats the decomposition in HTML. Indeed, that looks like a reasonable division of labor: the server does the Maths (the business logic), while the client takes care of the presentation.