



## Assignment 6 (HF)

Due: Mon 01.12.2014;16:00h(2 Weeks)

### Task 1: Currency Converter App

This assignment challenges you to write a currency conversion web app relying on jQuery and its AJAX functionalities. The currencies should be queried from a REST web service. The picture below shows a screenshot of a possible solution for the converter.

If you want to see it in action, please visit: <http://youtu.be/QGAOyUfm9Jg>

To retrieve currency conversion rates, please [sign up](#) for an API key at [openexchangerates.org](http://openexchangerates.org). This service is free to use for personal projects. **Update:** The link to the free sign up page is merely hidden below the 3 paid plans: [Sign up for free](#). Read the service's [API documentation](#) before your start coding.

The screenshot shows a web application titled "MMN Currency Converter". It features two input fields: the first contains "1.1" and is set to "USD", the second contains "0,8788802" and is set to "EUR". Below these are radio buttons for selecting currencies: EUR (checked), USD (checked), GBP (checked), CHF (checked), AUD (unchecked), CAD (unchecked), AED (unchecked), and DKK (unchecked). A section titled "Conversion Overview" contains a table of exchange rates:

<b>EUR:</b>	1 EUR = 1.2516 USD	1 EUR = 0.7989 GBP	1 EUR = 1.2014 CHF
<b>USD:</b>	1 USD = 0.7990 EUR	1 USD = 0.6383 GBP	1 USD = 0.9599 CHF
<b>GBP:</b>	1 GBP = 1.2518 EUR	1 GBP = 1.5667 USD	1 GBP = 1.5039 CHF
<b>CHF:</b>	1 CHF = 0.8323 EUR	1 CHF = 1.0417 USD	1 CHF = 0.6649 GBP

Please consider the following requirements for your submission:

- **Namespacing.** Write a function `currencyApp` that includes everything necessary to operate the filters. This function should be your only global function that is called when the DOM is ready.
- **Conversion.** Your app should allow the user to convert in both ways. For that you should have:
  - Input field for currency A & B. This is where the user can type numbers.
  - Select field for currency A & B. This is where the user chooses a currency.

If the user starts typing in one field, the other field should be automatically updated immediately.

- **Overview table.** Create a table that shows all exchange rates between a given set of currencies. Allow the user to generate this table by choosing from USD, EUR, GBP, CHF, JPY, CAD, and AUD.



- **Usability.** Your app should always provide feedback and make things easy for the user.
  - *Thousands Separators.* Make sure to let the user type the number without thousands separators. Have your script insert them automatically. Decide whether you use commas or colons for the decimal part numbers.
  - *Visual Feedback.* Use a loading indicator when you start querying the currency rates. Because the app is only usable once the exchange rates have been loaded, you could block user interaction with a `div` element overlaying your entire content area. Once the data is there, you can fade out this element. You could also use an animated GIF to indicate a loading process: <http://www.ajaxload.info/>
- **Code Quality.** Make sure to have as few code redundancies as possible – stay flexible wherever possible.
- **History (optional).** Allow the user to choose a date for the conversion rates. This could come in handy, e.g. if you purchased an item online in the past or if you were on vacation and want to calculate exactly how much money you have spent. Load the exchange rates for a particular date on demand. Use a jQuery UI or HTML5 date picker for easy access.

Further notes:

- Please make sure to comment your code sufficiently to facilitate correction of your submission.
- Incomplete submissions are welcome. However, please make sure to include a “README” textfile to tell us how far you have come.
- **Don’t forget about “the challenge”:** if more than 75 people submit this assignment (partial solutions over 50% count), there will be a mock exam in January.