

# 305CDE Challenge 4

Colin Stephen

October 2014

---

## Edit 26.10.14

You may find it a challenge to make calls directly to the Wikipedia API without some advanced CORS skills, or installing Chrome extensions. To make life easier you can do the following:

- Use the following base URL for queries:
  - `https://community-wikipedia.p.mashape.com/api.php`
- Add the following lines *after* `req.open('get', url);` in your `get(url)` function block:

```
var key = "oCdV3MjLj1mshtyIXwBVzBqRktY9p1XJNiajsn1vsCETYVLwK3";  
req.setRequestHeader("X-Mashape-Key", key);
```

These lines will authenticate your AJAX request, and the JSON response should come back as required. Note that the documentation referred to in the constraints below, regarding the valid API query parameters, is still valid for this endpoint.

---

## 305CDE Challenge 4

This challenge should be attempted after you have attended the fourth lab and worked through Worksheet 4 (*The DOM and AJAX*). You may need to attend other labs to fulfil some of the requirements.

## Guidelines

- Commit your solution to a new private project/repository in GitLab, and give it a sensible name: for example **305CDE Challenge 4**.
- Add the following members with *reporter* permissions under **Settings > Members** for your project:
  - Mark Tyers `marktyers`
  - Jianhua Yang `jianhua`
  - Colin Stephen `c0lin`
- You are free to commit changes to your code to improve it further at any point up to the assessment submission deadline.
- You may not manage to satisfy all of the requirements, but you should aim to do so.
- Extended requirements may require knowledge from future lab sessions to complete.
- Marking will be based on how closely you meet the requirements, whether you attempt any of the extended requirements, and how maintainable your JS code is.

## Specification

Implement a Wikipedia (auto)-suggestion form using the Wikimedia JSON API.

## Requirements

### Minimal

This challenge requires you to implement a simple Wikipedia suggestion form, in which the user can type a word or words, and which then lists a range of associated Wikipedia articles that the user may be interested in.

- Produce an HTML page with (at least) a form `<input>` element for a “Query”, a “Suggest” `<button>`, and a “Suggestions” `<div>`.
- When the user clicks “Suggest” the div should be populated with an unordered list of items.
- Each item in the displayed list should be the title of a Wikipedia article related to the word(s) currently in the input field.
- Up to 20 relevant suggestions should be displayed.
- The user should be able to click through on the item links to the relevant article on the Wikipedia site.

### Extended

Try to achieve at least *two* of the following.

- Remove the “Suggest” button, and instead dynamically update the list of query suggestions presented while the user is typing

- In addition, replace any callbacks with a *promises-based approach* (covered in Week 5) to retrieving your JSON data and handling any errors
  - Note that promises will actually make the previous requirement *easier* to achieve.
- Refactor your code in to a re-usable JS module that could be easily integrated in to other HTML pages.

## Constraints

As always:

- Use only HTML, JavaScript, and optionally CSS styling. Do not use any additional JS libraries.

In addition:

- The implementation should be entirely client-side, using AJAX and JS DOM manipulation
  - In particular, any HTML form must NOT be submitted
- Use appropriate callback(s) to populate the suggestion list with the data you retrieve
- You must use the [Wikimedia JSON/XML API](#)
- The AJAX query you send must include the “[Opensearch](#)” action to retrieve the data.
  - See the [Opensearch documentation](#) for an example AJAX query
- Since the Opensearch action simply returns a list of Wikipedia article titles, use the fact that HTML links to articles use the article title with spaces being replaced by underscores, to generate your links. For example:
  - “JavaScript syntax” is found at  
`http://en.wikipedia.org/wiki/JavaScript\_syntax`
  - “Software design pattern” is found at  
`http://en.wikipedia.org/wiki/Software\_design\_pattern`