

Topics:

- Frontend libraries and frameworks
- jQuery / Bootstrap
- JSON responses

This is the second component of your individual coursework mark. It is worth 10% of your overall mark. As with the first task, ask a demonstrator or myself to check your solution by the given deadline (11am on Thu 14 November). It is okay to work on this task during the next three labs.

Task. Develop a basic Ajax frontend and Django backend for a web application that deals with a list of products. You will only need one model **Product** for this task. This model should have three fields: **name** (CharField), **description** (TextField) and **price** (DecimalField).

[5%] On the Django backend your application should respond to the four request types (GET, POST, PUT and DELETE). You can use the “method” field of the request objects to find out which type of request is being made. For each request type you should respond appropriately. Whenever data is required to be returned to the client this should be done using JSON (for instance, returning the list of products, or the details of a given product). Make use of Django’s JsonResponse object.

(<https://docs.djangoproject.com/en/stable/ref/request-response/#jsonresponse-objects>)

[5%] On the frontend one should be able to

- see the list of all products or the details of a single product (GET)
- add a new product to the list (POST)
- update the price of a product (PUT)
- delete a product (DELETE)

It is up to you to decide how these will be laid out on the browser. Any request from the client to the server should be done via **Ajax requests** using the jQuery library. So there should not be any page refreshes.

Sample use cases:

- When accessing the application for the first time an Ajax request is made to the server requesting the current list of all products. The backend responds with this list in JSON format. Upon receiving this list, the frontend includes the list of products on the HTML using jQuery. This would be an example of an Ajax request using the GET method.
- A user clicks on the “delete” button next to a product. A DELETE Ajax request is made to the server so this product is removed from the database. At the same time, that list item containing the product is removed from the DOM.
- At the bottom of the page a user has the ability to include new products through a simple form. Upon inputting the details of a new product and clicking the “Add” button a POST Ajax request is triggered, so that the product is added on the application database, and at the same time the new product is added to the list of products shown on the page.

Note: This task is based on a similar task given to a former student as part of the selection process for a web development job at a well-known London-based IT company. The student was given 24 hours to complete the task.