



METU EE444 Introduction to Computer Networks

HW1 – HTTP Requests with Python



You are going to submit your homework via **ODTUCLASS** as a Python file named “EE444_HW1_studentid.py” and a report named “EE444_HW1_studentid.pdf”.

Your code should be well-structured and easily understandable. Write comments where needed, otherwise you may not get any credit or receive a penalty. Using code directly taken from any resource, except those provided in ODTUCLASS, is prohibited. You can verbally discuss the homework with your friends, but you should not exchange codes with each other or write your codes together. **Cheating and plagiarism will result in zero grades, whereas disciplinary actions may also be taken.**

In this homework, you will implement a web scraper using Python that can extract the prices in a shopping website and find the cheapest item for purchase. During this homework, you will use HTTP protocol for server communication and analyze HTML documents.

For each part, document the output (print values) you obtained in your report by providing a screenshot. Also, state your reasoning during the writing of your code.

1 Prerequisites

Before starting the homework, please try to get familiar with the basics of the following concepts:

- HTML: Tags, Attributes, Contents, ...
- JSON
- Your browser's developer tools: Inspect Element, Network Analysis, ...
- Python libraries: *BeautifulSoup4*, *Requests*

2 Web-Scraper (100%)

You are going to scrape a book list from ODTÜDEN (<https://www.odtuden.com.tr/>), find the cheapest book from this list, and add it to your cart. Most of the steps will consist of the following logic:

- Request a webpage
- Parse the HTML using *BeautifulSoup4*
- Find the related tag(s)

Since you are going to make multiple requests, please make sure to use `requests.Session()` to obtain a session to preserve the cookies between each request.

2.1 Getting Book Information (50%)

Getting single book info (25%)

If you visit the <https://www.odtuden.com.tr/kitaplik>, you can see the books are displayed on a grid. By inspecting the source, you can find out that each book entry follows the template in Figure 1.

```
<div class="productItem">
    <div class="productImage">...</div>
    <div class="productDetail" ...>
        <div class="productMarka" style="display: none;">ODTÜDEN</div>
        <div class="productName" ...>
            <a title="Köprü Hidroliği" href="/kopru-hidrologi">
                Köprü Hidrologi
            </a>
        </div>
        <div class="productPrice">
            <div class="discountPrice">
                <span class="discountPriceSpan">
                    ₺350,00
                </span>
            </div>
        </div>
    </div>
    ...
</div>
```

Figure 1: Product Item Example

To obtain the book information, follow these steps:

- Request the page `https://www.odtuden.com.tr/kitaplik` using `requests.get()`
- Parse the response payload using `BeautifulSoup4`
- Find the first `div` with `class productItem`
- Obtain the book title, link, and price

Print the book title, link, and price to check if your code runs successfully.

Getting multiple book info on a single page (10%)

After you validate that your previous code runs correctly, instead of finding the first `div` with `class productItem`, find all `divs` with `class productItem`. Extract the required information from these `divs` and put them in a list of tuples. **Print the book list you have obtained.**

Getting multiple pages (15%)

Since all the books do not fit on a single page, you need to make multiple requests for each page and gather all the available book information from there. To access a single page you need to make a request to `https://www.odtuden.com.tr/kitaplik?sayfa=NUMBER` where `NUMBER` is the page number starting from 1. To obtain all the book information,

you need to iterate every page and get all the information on that page. Iterations should continue until the response from the server is not **200 OK**.



If a page number is invalid, ODTÜDEN redirects to the first page. By default, *Requests* follow these redirects. Because of this, you should disable redirects in your requests. Also, put a hard cap on page numbers to prevent a possible infinite loop.

After all these steps are done, **print the number of books in your list**. You can check the correct number by manually visiting <https://www.odtuden.com.tr/kitaplik>. From your list, **find the cheapest book and print its information**.

2.2 Cart Interactions (50%)

Adding a book to your cart (10%)

During this step, make a request to the cheapest book's link and parse the response payload. To add the book to your cart, you need to obtain two values:

- UrunId
- UrunKartId

These values are held in a JSON object wrapped inside a `script`. To obtain these values, use the code snippet in Figure 2. **Print these values**. Adding an item to the cart is done by making a POST request to <https://www.odtuden.com.tr/api/cart/AddToCartV3> with a special JSON object as a payload. The format of this payload is given in Figure 3. Add the cheapest book to your cart using this API and **print the response status code**.

```
productDetail = soup.find("script", type="text/javascript",
                           string=lambda t: t.find("productDetailModel") != -1)
productDetail = productDetail.get_text().splitlines()[2]
productDetail = productDetail.split(";")[0].removeprefix(
    "var productDetailModel = ")
productDetail = json.loads(productDetail)

UrunId = productDetail["product"]["id"]
UrunKartId = productDetail["product"]["urunKartiId"]
```

Figure 2: Item Information Retrieval

```
{  
    "Adet" : 1,  
    "AsortiUrunKartId" : 0,  
    "BagliUrunId" : 0,  
    "FormId" : 0,  
    "KampanyaId" : 0,  
    "SelectProductGroupPostInfo" : [] ,  
    "UrunId" : UrunId,  
    "UrunKartId" : UrunKartId,  
    "UrunNot" : ""  
}
```

Figure 3: AddToCartV3 JSON Payload

Format of the AddToCart action (10%)

Using the developer tools of your browser, find the request format yourself given in Figure 3. Show all your steps clearly in your report.

Checking the cart (10%)

The cart information can be obtained by making a request to <https://www.odtuden.com.tr/api/cart/GetMemberCart>. The response will be a JSON object. By using this JSON object, **find the number of items in your cart and print it to the console**.

Removing the book from your cart (20%)

By using the developer tools, find a method to remove the book you have added from your cart. Show how you have obtained the format clearly in your report. After this, remove the book from your cart by making a post request. **Print the required information about the book for this request**. Finally, check the cart information to validate that your cart is empty. **Print the number of items in your cart**.