

```
import request

import logging
import argparse
from request.adapters import HTTPAdapter
from urllib3.util.retry import Retry

logging.basicConfig(level=logging.INFO, format='%(asctime)s -
%(levelname)s - %(message)s')

def create_session():
    session = request.Session()
    retries = Retry(total=5, backoff_factor=1, status_forcelist
                    = [500, 502, 503, 504])
    session.mount("https://", HTTPAdapter(max_retries=retries))
    return session

def fetch_and_save(url, output_file):
    session = create_session()
    try:
        logging.info(f"Requesting URL: {url}")
        response = session.get(url, timeout=10)
        response.raise_for_status()
        with open(output_file, 'w'), encoding="utf-8" as f:
            f.write(response.text)
        logging.info(f"Page saved to {output_file}")
    except requests.exceptions.RequestException as e:
        logging.error(f"Error occurred: {e}")
```


1

```
if __name__ == "__main__":
```

```
    parser = argparse.ArgumentParser(description = 'Can  
download a webpage and save it to a file')
```

```
    parser.add_argument("url", help="The URL of the  
webpage to download")
```

```
    parser.add_argument("-o", "--output", default = "output.htm",  
                        help="output filename (default : output.html)")
```

```
    args = parser.parse_args()
```

```
    fetch_and_save(args.url)
```

```
    fetch_and_save(args.url, args.output)
```