Flask Response Open Source Report

General Information and Licensing

Code Repository	**Flask**: https://github.com/pallets/flask/ **werkzeug**: https://github.com/pallets/werkzeug
License Type	**Flask**: BSD-3-Clause Source License **werkzeug**: BSD-3-Clause License
License Description	Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. 3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
License Restrictions	THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Who worked with this?	Sungho, Tariq, Ahsan

Class flask.Response

Purpose

What does this tech do for you in your project?	This class handles generating HTTP responses from requests.
Where specifically is this tech used in your project?	Throughout /flask_server.py as part of class Flask instantiation (https://github.com/rdstrach/CSE312_Group_Project/blob/main/flask_server.py)

Magic ★★ ° · ☽ ° ⤵ ° ★彡★ ৺

Documentation	• https://flask.palletsprojects.com/en/2.1.x/api/#flask.Response • https://werkzeug.palletsprojects.com/en/2.1.x/wrappers/

How does this technology do what it does for you in the Purpose section of this report?	• The class flask.Response works similarly as a subclass of the class werkzeug.wrappers.Response that represents an outgoing WSGI HTTP response with headers, status, and body. An exception to flask.Response is that it is set to have a "text/HTML" mimetype by default. https://github.com/pallets/flask/blob/a03719b01076a5bfdc2c8f4024eda7b874614bc1/src/flask/wrappers.py#L136
	• The WSGI environment containing the HTTP headers and information/body from the WSGI is contained in "environ". https://github.com/pallets/werkzeug/blob/c7ae2fea4fb229ffd71187c2b665874c91b96277/src/werkzeug/wrappers/response.py#L67
	• The data contained in "environ" is passed to werkzeug.wrappers.Response, where it is accessible as class variables. It then initializes that information in werkzeug.Response.__init__(). https://github.com/pallets/werkzeug/blob/c7ae2fea4fb229ffd71187c2b665874c91b96277/src/werkzeug/wrappers/response.py#L170
	• The main arguments to build the response in werkzeug.Response.__init__() include:
	  1. response
	  2. status
	  3. headers
	  4. mimetype
	  5. content_type
	  6. direct_passthrough

	- The status codes, mimetype, headers, and content_type are found in "environ". The processing for them can be found in http.py as referenced below https://github.com/pallets/werkzeug/blob/c7ae2fea4fb229ffd71187c2b665874c91b96277/src/werkzeug/http.py#L112 https://github.com/pallets/werkzeug/blob/c7ae2fea4fb229ffd71187c2b665874c91b96277/src/werkzeug/http.py#L100 https://github.com/pallets/werkzeug/blob/c7ae2fea4fb229ffd71187c2b665874c91b96277/src/werkzeug/http.py#L100 https://github.com/pallets/werkzeug/blob/c7ae2fea4fb229ffd71187c2b665874c91b96277/src/werkzeug/http.py#L85 - The data/string for the response body is passed as an argument to the Response class, and generally set by the server using routing as shown in lines 86-101 https://github.com/pallets/werkzeug/blob/c7ae2fea4fb229ffd71187c2b665874c91b96277/src/werkzeug/wrappers/response.py#L86 - The response body is then passed to direct_passthrough, and then set using werkzeug.Response.set_data(), which encodes and process the response where necessary https://github.com/pallets/werkzeug/blob/c7ae2fea4fb229ffd71187c2b665874c91b96277/src/werkzeug/wrappers/response.py#L322 - All this HTTP response information is stored in the werkzeug.wrappers.Response object, and accessible to the flask.Request object in the same manner.
Where is the specific code that does what you use the tech for?	- flask.Response: https://github.com/pallets/flask/blob/a03719b01076a5bfdc2c8f4024eda7b874614bc1/src/flask/wrappers.py#L136 - werkzeug.wrappers.Response: https://github.com/pallets/werkzeug/blob/c7ae2fea4fb229ffd71187c2b665874c91b96277/src/werkzeug/wrappers/response.py#L67 - werkzeug.wrappers.Response.__init__(): https://github.com/pallets/werkzeug/blob/c7ae2fea4fb229ffd71187c2b665874c91b96277/src/werkzeug/wrappers/response.py#L170 - werkzeug/http.py: https://github.com/pallets/werkzeug/blob/c7ae2fea4fb229ffd71187c2b665874c91b96277/src/werkzeug/http.py - werkzeug/http.py Status Codes: https://github.com/pallets/werkzeug/blob/c7ae2fea4fb229ffd71187c2b665874c91b96277/src/werkzeug/http.py#L112 - werkzeug/http.py Headers: https://github.com/pallets/werkzeug/blob/c7ae2fea4fb229ffd71187c2b665874c91b96277/src/werkzeug/http.py#L100

	- werkzeug/http.py MIME types: https://github.com/pallets/werkzeug/blob/c7ae2fea4fb229ffd71187c2b665874c91b96277/src/werkzeug/http.py#L85 - werkzeug.wrappers.Response body data: https://github.com/pallets/werkzeug/blob/c7ae2fea4fb229ffd71187c2b665874c91b96277/src/werkzeug/wrappers/response.py#L86 - werkzeug.wrappers.response.set_data(): https://github.com/pallets/werkzeug/blob/c7ae2fea4fb229ffd71187c2b665874c91b96277/src/werkzeug/wrappers/response.py#L322