Semantic-Score	Original-Comment	Agent-Comment	Label
0.475059	"""This is called at the end of each request, after generating a response, before removing the request context. It is skipped if :meth:`is_null_session` returns `True`. """	""" Saves a session. This method is intended to be overridden by subclasses. It takes in the Flask application, the session object, and the response object as parameters. The implementation of this method should be provided by the subclass. Parameters: app (Flask): The Flask application instance. session (SessionMixin): The session object. response (Response): The response object. Returns: None Raises: NotImplementedError: This method is intended to be overridden and should not be called directly. """	L
0.428235	"""What environment the app is running in. This maps to the :data:`ENV` config key. **Do not enable development when deploying in production.** Default: `'production'` .. deprecated:: 2.2 Will be removed in Flask 2.3. """	""" Deprecation Warning: `env` method is deprecated and will be removed in Flask 2.3. Use `debug` attribute instead. Args: value (str): The environment variable to set. Returns: None """	L
0.512154	"""Reload templates when they are changed. Used by :meth:`create_jinja_environment`. It is enabled by default in debug mode. .. deprecated:: 2.2 Will be removed in Flask 2.3. Use `app.config["TEMPLATES_AUTO_RELOAD"]` instead. .. versionadded:: 1.0	""" Returns the value of `TEMPLATES_AUTO_RELOAD` from the application configuration. If `TEMPLATES_AUTO_RELOAD` is set, its value is returned. Otherwise, the value of `debug` is used as a fallback. Deprecated since Flask 2.3 in favor of using `TEMPLATES_AUTO_RELOAD` in `app.config`. Args:	L

Semantic-Score	Original-Comment	Agent-Comment	Label
	This property was added but the underlying config and behavior already existed. """ """When used in combination with a `with` statement this opens a session transaction. This can be used to modify the session that the test client uses. Once the `with` block is left the session is stored back. :: with client.session_transaction() as session: session['value'] = 42 Internally this is implemented by going through a temporary test request context and since session handling could depend on request variables this function accepts the same arguments as :meth:`~flask.Flask.test_request_context` which are directly passed through. """ """The actual WSGI application. This is not implemented in :meth:`__call__` so that middlewares can be applied without losing a reference to the app object. Instead of doing this:: app = MyMiddleware(app) It's a better idea to do this instead::	None Returns: bool: The value of `TEMPLATES_AUTO_RELOAD` or `debug` if not set. """ """ Yield a session object for the current test request context. This function is used to create and manage sessions for testing purposes. It checks if cookies are enabled, sets up the WSGI context, opens a new session, and saves it after use. If the session backend fails to open a session, a RuntimeError is raised. Args: *args: Variable length argument list containing any arguments passed to the test request context. **kwargs: Keyworded arguments for the test request context. Returns: A generator yielding SessionMixin objects. """ """ WSGI Application Function This function serves as the entry point for the WSGI application. It takes in an environment dictionary and a start response callable, and returns any response object generated by the application. The function first creates a request context using `self.request_context(environ)`. It then attempts to execute the full dispatch of the request,	
0.516763			M
0.568346			L

Semantic-Score	Original-Comment	Agent-Comment	Label
		handling any exceptions that may occur during this process. If an exception is caught, it will be handled and propagated up the call stack.	
	app.wsgi_app = MyMiddleware(app.wsgi_app)	Finally, the response object is returned to the caller.	
	Then you still have the original application object around and	Note: This function should not be called directly by users of the application. Instead, it should be used as part of a larger WSGI server or framework.	
	can continue to call methods on it.	"""	
	.. versionchanged:: 0.7		
	Teardown events for the request and app contexts are called		
	even if an unhandled error occurs. Other events may not be		
	called depending on when an error occurs during dispatch.		
	See :ref:`callbacks-and-errors`.		
	:param environ: A WSGI environment.		
	:param start_response: A callable accepting a status code,		
	a list of headers, and an optional exception context to		
	start the response.		
	"""		
0.555169	"""The URL prefix that the static route will be accessible from.	"""	L
		Returns a static URL path.	
	If it was not configured during init, it is derived from		
	:attr:`static_folder`.	This method takes a string or None as input and returns the URL path after removing any trailing slashes. If the input is None, it sets the internal `_static_url_path` attribute to None.	
	"""		
		Args:	
		value (str	None): The URL path to be processed.
		Returns:	
		None	
		"""	
0.55458	"""Can be overridden in order to modify the response object	"""	M
	before it's sent to the WSGI server. By default this will	Processes the given response by executing any after-request functions and saving the session.	
	call all the :meth:`after_request` decorated functions.		
		Args:	
	.. versionchanged:: 0.5	response (Response): The response to be processed.	
	As of Flask 0.5 the functions registered for after request		

Semantic-Score	Original-Comment	Agent-Comment	Label
	execution are called in reverse order of registration. :param response: a :attr:`response_class` object. :return: a new response object or the same, has to be an instance of :attr:`response_class`. """	Returns: Response: The processed response. """	
0.536651	"""What environment the app is running in. This maps to the :data:`ENV` config key. **Do not enable development when deploying in production.** Default: `'production'` .. deprecated:: 2.2 Will be removed in Flask 2.3. """	""" Returns the environment variable as a string. Deprecation Warning: This method is deprecated and will be removed in Flask 2.3. Use `app.debug` instead. Args: None Returns: str: The environment variable value. Raises: DeprecationWarning: If the 'app.env' method is called. """	M
0.61878	"""Given the return value from a view function this finalizes the request by converting it into a response and invoking the postprocessing functions. This is invoked for both normal request dispatching as well as error handlers. Because this means that it might be called as a result of a failure a special safe mode is available which can be enabled with the `from_error_handler` flag. If enabled, failures in response processing will be logged and otherwise ignored. :internal: """	""" Finalizes a request by processing the response and sending a signal to indicate that the request has finished. Args: rv (ft.ResponseReturnValue from_error_handler (bool, optional): Whether this is being called from an error handler. Defaults to False. Returns: Response: The finalized response object. Raises: Exception: If the request finalizing fails and `from_error_handler` is False. """	M HTTPException): The response value or exception to be finalized.

Semantic-Score	Original-Comment	Agent-Comment	Label
0.604298	"""Create a :class:`~flask.ctx.RequestContext` representing a WSGI environment. Use a `with` block to push the context, which will make :data:`request` point at this request. See :doc:`/reqcontext`. Typically you should not call this from your own code. A request context is automatically pushed by the :meth:`wsgi_app` when handling a request. Use :meth:`test_request_context` to create an environment and context instead of this method. :param environ: a WSGI environment """	""" Returns a new instance of RequestContext with the given environment. Args: environ (dict): The current HTTP environment. Returns: RequestContext: A new instance of RequestContext. """	M
0.596418	"""Called before the request is dispatched. Calls :attr:`url_value_preprocessors` registered with the app and the current blueprint (if any). Then calls :attr:`before_request_funcs` registered with the app and the blueprint. If any :meth:`before_request` handler returns a non-None value, the value is handled as if it was the return value from the view, and further request handling is stopped. """	""" Preprocesses the request by applying URL value preprocessors and before request functions. This method iterates over the blueprint names in reverse order, applying any URL value preprocessors to each one. It then iterates over the same list again, applying any before request functions to each one. If a function returns a non-None value, it is returned immediately. Otherwise, None is returned at the end. Args: self: The instance of the class this method belongs to. Returns: ft.ResponseReturnValue """	H None: The result of the preprocess request, or None if no functions return a value.
0.618487	"""Create the Jinja environment based on :attr:`jinja_options` and the various Jinja-related methods of the app. Changing :attr:`jinja_options` after this will have no effect. Also adds Flask-related globals and filters to the environment.	""" Creates a Jinja environment with custom options and updates its globals. This method creates a new Jinja environment based on the provided options.	H

Semantic-Score	Original-Comment	Agent-Comment	Label
		It also updates the environment's globals dictionary to include necessary functions	
	.. versionchanged:: 0.11 `Environment.auto_reload` set in accordance with `TEMPLATES_AUTO_RELOAD` configuration option. .. versionadded:: 0.5 """	and variables for use in templates. Args: self: The object instance that owns this method. Returns: Environment: A newly created Jinja environment with custom options and updated globals. """	
0.601796	"""The `--key` option must be specified when `--cert` is a file. Modifies the `cert` param to be a `(cert, key)` pair if needed. """	""" Validate the key for a given certificate. This function checks if the provided key is valid based on the type of certificate used. It raises an error if the key is not required or if it's used with an invalid certificate type. Args: ctx (click.Context): The context object containing the command-line arguments. param (click.Parameter): The parameter being validated. value: The value to be validated. Returns: value: The validated key value. Raises: click.BadParameter: If the key is not required or if it's used with an invalid certificate type. """	L
0.614353	"""Loads the Flask app (if not yet loaded) and returns it. Calling this multiple times will just result in the already loaded app to be returned. """	""" Loads a Flask application instance. This method attempts to load an existing Flask application from the `self._loaded_app` attribute, or creates a new one if none exists. It also sets the debug flag of the loaded application if `self.set_debug_flag` is True. If no application can be found, it raises a `NoAppException`. Returns: Flask: The loaded or created Flask application instance.	H

Semantic-Score	Original-Comment	Agent-Comment	Label
		"""	
0.593989	"""Handles an HTTP exception. By default this will invoke the	"""	H
	registered error handlers and fall back to returning the	Handles HTTP exceptions by checking their type and returning them accordingly.	
	exception as response.		
		If the exception does not have an error code (i.e., it's a ProxyException),	
	.. versionchanged:: 1.0.3	it will be returned unchanged as an error. If it's a RoutingException,	
	`RoutingException`, used internally for actions such as	it will also be returned without modification. Otherwise, it will be	
	slash redirects during routing, is not passed to error	passed to the error handler function to determine its response.	
	handlers.		
		Args:	
	.. versionchanged:: 1.0	e (HTTPException): The HTTP exception to handle.	
	Exceptions are looked up by code *and* by MRO, so	Returns:	
	`HTTPException` subclasses can be handled with a catch-all	HTTPException	ft.ResponseReturnValue: The handled exception or its response.
	handler for the base `HTTPException`.	"""	
	.. versionadded:: 0.3		
	"""		
0.655802	"""Opens a resource from the application's instance folder	"""	H
	(:attr:`instance_path`). Otherwise works like	Opens an instance resource file.	
	:meth:`open_resource`. Instance resources can also be opened for		
	writing.	Args:	
		- `resource` (str): The path to the resource file.	
	:param resource: the name of the resource. To access resources within	- `mode` (str, optional): The mode in which to open the file. Defaults to "rb".	
	subfolders use forward slashes as separator.		
	:param mode: resource file opening mode, default is 'rb'.	Returns:	
	"""	A file object opened at the specified location with the given mode.	
		Raises:	
		FileNotFoundError: If the instance_path does not exist or the resource is not found.	
		"""	
0.627122	"""Wraps a callback so that it's guaranteed to be executed with the	"""	M
	script's application context.	Decorates a function to run with the application context.	
	Custom commands (and their options) registered under `app.cli` or	This decorator is used to ensure that functions decorated with it are executed within	

Semantic-Score	Original-Comment	Agent-Comment	Label
	`blueprint.cli` will always have an app context available, this decorator is not required in that case. .. versionchanged:: 2.2 The app context is active for subcommands as well as the decorated callback. The app context is always available to `app.cli` command and parameter callbacks. """	the application's context. This can be useful for tasks such as database operations, file I/O, or any other operation that requires access to the current application state. The `with_appcontext` function takes a function `f` as an argument and returns the result of invoking `f` with the application context. If the application context is not already set, it will load the application from the `ScriptInfo` object associated with the click context. Args: f (function): The function to be decorated. Returns: function: The original function, wrapped in a decorator that runs it with the application context. """	
0.652406	"""Iterates over all blueprints by the order they were registered. .. versionadded:: 0.11 """	""" Returns an iterator over the blueprint values. This method provides a view of all blueprints in the system, allowing for efficient iteration and access to their attributes. The returned iterator is a `ValuesView` object, which supports various methods for filtering and manipulating the results. Args: None Returns: t.ValuesView[Blueprint]: An iterator over the blueprint values. """	H
0.625945	"""Blueprint-local JSON encoder class to use. Set to `None` to use the app's. .. deprecated:: 2.2 Will be removed in Flask 2.3. Customize :attr:`json_provider_class` instead. .. versionadded:: 0.10 """	""" Returns the JSON encoder class, deprecation warning if applicable. This function is deprecated and will be removed in Flask 2.3. It's recommended to customize 'app.json_provider_class' or 'app.json' instead. Args: None Returns:	H

Semantic-Score	Original-Comment	Agent-Comment	Label
		t.Union[t.Type[json.JSONEncoder], None]: The JSON encoder class or None. """	
0.693479	"*Test jsonify with uuid.UUID types"*	""" Tests the JSONification of UUID types. This function tests that a UUID object can be successfully serialized to JSON and deserialized back into a UUID object. Parameters: app (Flask application): The Flask application instance. client (Flask client): The Flask client instance. Returns: None """	H
0.69815	"""Returns True if the session cookie should be httponly. This currently just returns the value of the `SESSION_COOKIE_HTTPONLY` config var. """	""" Returns whether the session cookie is set to be HTTP-only. Args: app (Flask): The application instance. Returns: bool: True if the session cookie is HTTP-only, False otherwise. """	M
0.745833	"""Decorate a view function to register it for the given endpoint. Used if a rule is added without a `view_func` with :meth:`add_url_rule`. .. code-block:: python app.add_url_rule("/ex", endpoint="example") @app.endpoint("example") def example(): ... :param endpoint: The endpoint name to associate with the view function.	""" Endpoint Decorator Function This function is a decorator that registers an endpoint with the provided endpoint string. It takes in a function `f` and returns a new function that wraps the original function, registering it as a view function for the specified endpoint. Args: endpoint (str): The endpoint to register the function under. f (Callable[[F], F]): The function to be registered as a view function. Returns: Callable[[F], F]: A new function that wraps the original function and registers it as a view function. """	H

Semantic-Score	Original-Comment	Agent-Comment	Label
0.743081	""" """Convert the value to a valid JSON type and add the tag structure around it.""""	""" Converts a given value to JSON format and returns it as a dictionary. Args: value (t.Any): The value to be converted to JSON format. Returns: dict[str, t.Any]: A dictionary containing the key-value pair where the key is 'tag' and the value is the JSON representation of the input value. """	H