**Using sessions in views**[**¶**](https://docs.djangoproject.com/en/2.1/topics/http/sessions/#using-sessions-in-views)

**When SessionMiddleware is activated, each** [**HttpRequest**](https://docs.djangoproject.com/en/2.1/ref/request-response/#django.http.HttpRequest) **object – the first argument to any Django view function – will have a session attribute, which is a dictionary-like object.**

**You can read it and write to request.session at any point in your view. You can edit it multiple times.**

***class* backends.base.SessionBase[¶](https://docs.djangoproject.com/en/2.1/topics/http/sessions/" \l "django.contrib.sessions.backends.base.SessionBase" \o "Permalink to this definition)**

**This is the base class for all session objects. It has the following standard dictionary methods:**

**\_\_getitem\_\_(*key*)**[**¶**](https://docs.djangoproject.com/en/2.1/topics/http/sessions/#django.contrib.sessions.backends.base.SessionBase.__getitem__)

**Example: fav\_color = request.session['fav\_color']**

**\_\_setitem\_\_(*key*, *value*)**[**¶**](https://docs.djangoproject.com/en/2.1/topics/http/sessions/#django.contrib.sessions.backends.base.SessionBase.__setitem__)

**Example: request.session['fav\_color'] = 'blue'**

**\_\_delitem\_\_(*key*)**[**¶**](https://docs.djangoproject.com/en/2.1/topics/http/sessions/#django.contrib.sessions.backends.base.SessionBase.__delitem__)

**Example: del request.session['fav\_color']. This raises KeyError if the given key isn’t already in the session.**

**\_\_contains\_\_(*key*)**[**¶**](https://docs.djangoproject.com/en/2.1/topics/http/sessions/#django.contrib.sessions.backends.base.SessionBase.__contains__)

**Example: 'fav\_color' in request.session**

**get(*key*, *default=None*)**[**¶**](https://docs.djangoproject.com/en/2.1/topics/http/sessions/#django.contrib.sessions.backends.base.SessionBase.get)

**Example: fav\_color = request.session.get('fav\_color', 'red')**

**pop(*key*, *default=\_\_not\_given*)**[**¶**](https://docs.djangoproject.com/en/2.1/topics/http/sessions/#django.contrib.sessions.backends.base.SessionBase.pop)

**Example: fav\_color = request.session.pop('fav\_color', 'blue')**

**keys()**[**¶**](https://docs.djangoproject.com/en/2.1/topics/http/sessions/#django.contrib.sessions.backends.base.SessionBase.keys)

**items()**[**¶**](https://docs.djangoproject.com/en/2.1/topics/http/sessions/#django.contrib.sessions.backends.base.SessionBase.items)

**setdefault()**[**¶**](https://docs.djangoproject.com/en/2.1/topics/http/sessions/#django.contrib.sessions.backends.base.SessionBase.setdefault)

**clear()**[**¶**](https://docs.djangoproject.com/en/2.1/topics/http/sessions/#django.contrib.sessions.backends.base.SessionBase.clear)

**It also has these methods:**

**flush()**[**¶**](https://docs.djangoproject.com/en/2.1/topics/http/sessions/#django.contrib.sessions.backends.base.SessionBase.flush)

**Deletes the current session data from the session and deletes the session cookie. This is used if you want to ensure that the previous session data can’t be accessed again from the user’s browser (for example, the** [**django.contrib.auth.logout()**](https://docs.djangoproject.com/en/2.1/topics/auth/default/#django.contrib.auth.logout) **function calls it).**

**set\_test\_cookie()**[**¶**](https://docs.djangoproject.com/en/2.1/topics/http/sessions/#django.contrib.sessions.backends.base.SessionBase.set_test_cookie)

**Sets a test cookie to determine whether the user’s browser supports cookies. Due to the way cookies work, you won’t be able to test this until the user’s next page request. See** [**Setting test cookies**](https://docs.djangoproject.com/en/2.1/topics/http/sessions/#setting-test-cookies) **below for more information.**

**test\_cookie\_worked()**[**¶**](https://docs.djangoproject.com/en/2.1/topics/http/sessions/#django.contrib.sessions.backends.base.SessionBase.test_cookie_worked)

**Returns either True or False, depending on whether the user’s browser accepted the test cookie. Due to the way cookies work, you’ll have to call set\_test\_cookie() on a previous, separate page request. See** [**Setting test cookies**](https://docs.djangoproject.com/en/2.1/topics/http/sessions/#setting-test-cookies) **below for more information.**

**delete\_test\_cookie()**[**¶**](https://docs.djangoproject.com/en/2.1/topics/http/sessions/#django.contrib.sessions.backends.base.SessionBase.delete_test_cookie)

**Deletes the test cookie. Use this to clean up after yourself.**

**set\_expiry(*value*)**[**¶**](https://docs.djangoproject.com/en/2.1/topics/http/sessions/#django.contrib.sessions.backends.base.SessionBase.set_expiry)

**Sets the expiration time for the session. You can pass a number of different values:**

* **If value is an integer, the session will expire after that many seconds of inactivity. For example, calling request.session.set\_expiry(300) would make the session expire in 5 minutes.**
* **If value is a datetime or timedelta object, the session will expire at that specific date/time. Note that datetime and timedelta values are only serializable if you are using the** [**PickleSerializer**](https://docs.djangoproject.com/en/2.1/topics/http/sessions/#django.contrib.sessions.serializers.PickleSerializer)**.**
* **If value is 0, the user’s session cookie will expire when the user’s Web browser is closed.**
* **If value is None, the session reverts to using the global session expiry policy.**

**Reading a session is not considered activity for expiration purposes. Session expiration is computed from the last time the session was *modified*.**

**get\_expiry\_age()**[**¶**](https://docs.djangoproject.com/en/2.1/topics/http/sessions/#django.contrib.sessions.backends.base.SessionBase.get_expiry_age)

**Returns the number of seconds until this session expires. For sessions with no custom expiration (or those set to expire at browser close), this will equal** [**SESSION\_COOKIE\_AGE**](https://docs.djangoproject.com/en/2.1/ref/settings/#std:setting-SESSION_COOKIE_AGE)**.**

**This function accepts two optional keyword arguments:**

* **modification: last modification of the session, as a** [**datetime**](https://docs.python.org/3/library/datetime.html#datetime.datetime) **object. Defaults to the current time.**
* **expiry: expiry information for the session, as a** [**datetime**](https://docs.python.org/3/library/datetime.html#datetime.datetime) **object, an** [**int**](https://docs.python.org/3/library/functions.html#int) **(in seconds), or None. Defaults to the value stored in the session by** [**set\_expiry()**](https://docs.djangoproject.com/en/2.1/topics/http/sessions/#django.contrib.sessions.backends.base.SessionBase.set_expiry)**, if there is one, or None.**

**get\_expiry\_date()**[**¶**](https://docs.djangoproject.com/en/2.1/topics/http/sessions/#django.contrib.sessions.backends.base.SessionBase.get_expiry_date)

**Returns the date this session will expire. For sessions with no custom expiration (or those set to expire at browser close), this will equal the date** [**SESSION\_COOKIE\_AGE**](https://docs.djangoproject.com/en/2.1/ref/settings/#std:setting-SESSION_COOKIE_AGE) **seconds from now.**

**This function accepts the same keyword arguments as** [**get\_expiry\_age()**](https://docs.djangoproject.com/en/2.1/topics/http/sessions/#django.contrib.sessions.backends.base.SessionBase.get_expiry_age)**.**

**get\_expire\_at\_browser\_close()**[**¶**](https://docs.djangoproject.com/en/2.1/topics/http/sessions/#django.contrib.sessions.backends.base.SessionBase.get_expire_at_browser_close)

**Returns either True or False, depending on whether the user’s session cookie will expire when the user’s Web browser is closed.**

**clear\_expired()**[**¶**](https://docs.djangoproject.com/en/2.1/topics/http/sessions/#django.contrib.sessions.backends.base.SessionBase.clear_expired)

**Removes expired sessions from the session store. This class method is called by** [**clearsessions**](https://docs.djangoproject.com/en/2.1/ref/django-admin/#django-admin-clearsessions)**.**

**cycle\_key()**[**¶**](https://docs.djangoproject.com/en/2.1/topics/http/sessions/#django.contrib.sessions.backends.base.SessionBase.cycle_key)

**Creates a new session key while retaining the current session data.** [**django.contrib.auth.login()**](https://docs.djangoproject.com/en/2.1/topics/auth/default/#django.contrib.auth.login) **calls this method to mitigate against session fixation.**