## 1requests模块函数与类

### 7.1.1 request.get

函数原型：

def **get**(url, params=None, \*\*kwargs):

*r"""Sends a GET request.*

**:param** *url: URL for the new :class:`Request` object.*

**:param** *params: (optional) Dictionary or bytes to be sent in the query string for the :class:`Request`.*

**:param** *\\*\\*kwargs: Optional arguments that ``request`` takes.*

**:return:****:class:***`Response <Response>` object*

**:rtype:** *requests.Response*

*"""*

kwargs.setdefault(*'allow\_redirects'*, True)

return request(*'get'*, url, params=params, \*\*kwargs)

参数：

url:请求地址

params：(可选)在查询字符串中发送的字典或字节

kwargs:requests可获取的可选参数

返回值：返回一个*Response实例*对象

def **request**(method, url, \*\*kwargs):

*"""Constructs and sends a :class:`Request <Request>`.*

**:param** *method: method for the new :class:`Request` object.*

**:param** *url: URL for the new :class:`Request` object.*

**:param** *params: (optional) Dictionary or bytes to be sent in the query string for the :class:`Request`.*

**:param** *data: (optional) Dictionary or list of tuples ``[(key, value)]`` (will be form-encoded), bytes, or file-like object to send in the body of the :class:`Request`.*

**:param** *json: (optional) json data to send in the body of the :class:`Request`.*

**:param** *headers: (optional) Dictionary of HTTP Headers to send with the :class:`Request`.*

**:param** *cookies: (optional) Dict or CookieJar object to send with the :class:`Request`.*

**:param** *files: (optional) Dictionary of ``'name': file-like-objects`` (or ``{'name': file-tuple}``) for multipart encoding upload.*

*``file-tuple`` can be a 2-tuple ``('filename', fileobj)``, 3-tuple ``('filename', fileobj, 'content\_type')``*

*or a 4-tuple ``('filename', fileobj, 'content\_type', custom\_headers)``, where ``'content-type'`` is a string*

*defining the content type of the given file and ``custom\_headers`` a dict-like object containing additional headers*

*to add for the file.*

**:param** *auth: (optional) Auth tuple to enable Basic/Digest/Custom HTTP Auth.*

**:param** *timeout: (optional) How many seconds to wait for the server to send data*

*before giving up, as a float, or a :ref:`(connect timeout, read*

*timeout) <timeouts>` tuple.*

**:type** *timeout: float or tuple*

**:param** *allow\_redirects: (optional) Boolean. Enable/disable GET/OPTIONS/POST/PUT/PATCH/DELETE/HEAD redirection. Defaults to ``True``.*

**:type** *allow\_redirects: bool*

**:param** *proxies: (optional) Dictionary mapping protocol to the URL of the proxy.*

**:param** *verify: (optional) Either a boolean, in which case it controls whether we verify*

*the server's TLS certificate, or a string, in which case it must be a path*

*to a CA bundle to use. Defaults to ``True``.*

**:param** *stream: (optional) if ``False``, the response content will be immediately downloaded.*

**:param** *cert: (optional) if String, path to ssl client cert file (.pem). If Tuple, ('cert', 'key') pair.*

**:return:****:class:***`Response <Response>` object*

**:rtype:** *requests.Response*

*Usage::*

*>>> import requests*

*>>> req = requests.request('GET', 'http://httpbin.org/get')*

*<Response [200]>*

*"""*

# By using the 'with' statement we are sure the session is closed, thus we

# avoid leaving sockets open which can trigger a ResourceWarning in some

# cases, and look like a memory leak in others.

with sessions.Session() as session:

return session.request(method=method, url=url, \*\*kwargs)

参数：

*method：请求方法*

*url：请求地址*

*params：传递给Request类的查询字符串或字典*

*data：（可选）字典或者元组（将被表单编码），字节或者类似于文件的对象，在Request事例中被发送*

*json：（可选）Request实例中被发送的json数据*

*headers：（可选）Request实例中被发送的HTTP头*

*cookies：（可选）Request实例中被发送的字典或CookieJar对象*

*files：（可选）可发送的文件*

*auth：*

*timeout：等待服务器时间，参数类型为float或tuple*

*allow\_redirects：允许重定向，默认True*

*proxies：（可选）字典映射到url的代理*

*verify：（可选）默认为True*

*stream：（可选）如果是False，reponse内容将被立即下载。*

*返回值：是一个requests.Response*

### 7.1.2 Request类

class **Request**(RequestHooksMixin):

*"""A user-created :class:`Request <Request>` object.*

*Used to prepare a :class:`PreparedRequest <PreparedRequest>`, which is sent to the server.*

**:param** *method: HTTP method to use.*

**:param** *url: URL to send.*

**:param** *headers: dictionary of headers to send.*

**:param** *files: dictionary of {filename: fileobject} files to multipart upload.*

**:param** *data: the body to attach to the request. If a dictionary is provided, form-encoding will take place.*

**:param** *json: json for the body to attach to the request (if files or data is not specified).*

**:param** *params: dictionary of URL parameters to append to the URL.*

**:param** *auth: Auth handler or (user, pass) tuple.*

**:param** *cookies: dictionary or CookieJar of cookies to attach to this request.*

**:param** *hooks: dictionary of callback hooks, for internal usage.*

*Usage::*

*>>> import requests*

*>>> req = requests.Request('GET', 'http://httpbin.org/get')*

*>>> req.prepare()*

*<PreparedRequest [GET]>*

*"""*

### 7.1.3 Reponse类

class **Response**(object):

*"""The :class:`Response <Response>` object, which contains a*

*server's response to an HTTP request.*

*"""*

\_\_attrs\_\_ = [

*'\_content'*, *'status\_code'*, *'headers'*, *'url'*, *'history'*,

*'encoding'*, *'reason'*, *'cookies'*, *'elapsed'*, *'request'*

]

## 7.2 requests模块使用

### 7.2.1 发送请求