

**NAME**

OCSP\_REQUEST\_new, OCSP\_REQUEST\_free, OCSP\_request\_add0\_id, OCSP\_request\_sign, OCSP\_request\_add1\_cert, OCSP\_request\_onereq\_count, OCSP\_request\_onereq\_get0 - OCSP request functions

**SYNOPSIS**

```
#include <openssl/ocsp.h>
```

```
OCSP_REQUEST *OCSP_REQUEST_new(void);  
void OCSP_REQUEST_free(OCSP_REQUEST *req);
```

```
OCSP_ONEREQ *OCSP_request_add0_id(OCSP_REQUEST *req, OCSP_CERTID *cid);
```

```
int OCSP_request_sign(OCSP_REQUEST *req,  
                      X509 *signer, EVP_PKEY *key, const EVP_MD *dgst,  
                      STACK_OF(X509) *certs, unsigned long flags);
```

```
int OCSP_request_add1_cert(OCSP_REQUEST *req, X509 *cert);
```

```
int OCSP_request_onereq_count(OCSP_REQUEST *req);  
OCSP_ONEREQ *OCSP_request_onereq_get0(OCSP_REQUEST *req, int i);
```

**DESCRIPTION**

**OCSP\_REQUEST\_new()** allocates and returns an empty **OCSP\_REQUEST** structure.

**OCSP\_REQUEST\_free()** frees up the request structure **req**.

**OCSP\_request\_add0\_id()** adds certificate ID **cid** to **req**. It returns the **OCSP\_ONEREQ** structure added so an application can add additional extensions to the request. The **id** parameter **MUST NOT** be freed up after the operation.

**OCSP\_request\_sign()** signs OCSP request **req** using certificate **signer**, private key **key**, digest **dgst** and additional certificates **certs**. If the **flags** option **OCSP\_NOCERTS** is set then no certificates will be included in the request.

**OCSP\_request\_add1\_cert()** adds certificate **cert** to request **req**. The application is responsible for freeing up **cert** after use.

**OCSP\_request\_onereq\_count()** returns the total number of **OCSP\_ONEREQ** structures in **req**.

**OCSP\_request\_onereq\_get0()** returns an internal pointer to the **OCSP\_ONEREQ** contained in **req** of index **i**. The index value **i** runs from 0 to **OCSP\_request\_onereq\_count(req) - 1**.

## RETURN VALUES

**OCSP\_REQUEST\_new()** returns an empty **OCSP\_REQUEST** structure or **NULL** if an error occurred.

**OCSP\_request\_add0\_id()** returns the **OCSP\_ONEREQ** structure containing **cid** or **NULL** if an error occurred.

**OCSP\_request\_sign()** and **OCSP\_request\_add1\_cert()** return 1 for success and 0 for failure.

**OCSP\_request\_onereq\_count()** returns the total number of **OCSP\_ONEREQ** structures in **req** and -1 on error.

**OCSP\_request\_onereq\_get0()** returns a pointer to an **OCSP\_ONEREQ** structure or **NULL** if the index value is out of range.

## NOTES

An OCSP request structure contains one or more **OCSP\_ONEREQ** structures corresponding to each certificate.

**OCSP\_request\_onereq\_count()** and **OCSP\_request\_onereq\_get0()** are mainly used by OCSP responders.

## EXAMPLES

Create an **OCSP\_REQUEST** structure for certificate **cert** with issuer **issuer**:

```
OCSP_REQUEST *req;
OCSP_ID *cid;

req = OCSP_REQUEST_new();
if (req == NULL)
    /* error */
cid = OCSP_cert_to_id(EVP_sha1(), cert, issuer);
if (cid == NULL)
    /* error */

if (OCSP_REQUEST_add0_id(req, cid) == NULL)
    /* error */
```

```
/* Do something with req, e.g. query responder */
```

```
OCSP_REQUEST_free(req);
```

**SEE ALSO**

**crypto(7), OCSP\_cert\_to\_id(3), OCSP\_request\_add1\_nonce(3), OCSP\_resp\_find\_status(3), OCSP\_response\_status(3), OCSP\_sendreq\_new(3)**

**COPYRIGHT**

Copyright 2015-2016 The OpenSSL Project Authors. All Rights Reserved.

Licensed under the Apache License 2.0 (the "License"). You may not use this file except in compliance with the License. You can obtain a copy in the file LICENSE in the source distribution or at <https://www.openssl.org/source/license.html>.