



**draft-barnes-acme**

```
# apt-get install acme
```

and then...

# Requirements

<https://buy.wosign.com/free/>

SSL Request a Free SSL Certificate | Request a Free Email Certificate

**Step 1: Set certificate parameter**

Domain name:  ✓

1. The first one will be deemed as the main domain name to be listed in the certificate. 100 domain names at most.  
2. If you would like to bind more than one domain name in a certificate, please click "Enter" to start a new line after each domain name.

It's Free to Bind the domain name: www.ietf.org

Period:  1 year  2 year  3 year

Language:  Chinese (Displayed Chinese)  English (Displayed English)

Algorithm:  SHA1 (According to WebTrust BFI, CA can't issue SHA-1 certificate validity period greater than January 1, 2017.)  
 SHA2 (more Secure, but WinXP SP2 don't support)

**Step 2: Domain name control verification**

Not Validated Domain: ietf.org [Validate now](#)

**Step 3: Please Select Certificate Signing Request method**

Option 1: Generated by the system Quick and handy 😊  Option 2: Generated by myself

Please paste Certificate Signing Request

Please paste Certificate Signing Request file

**Step 4: Enter Email to receive the certificate collection link**

Email:

Please make sure it is correct, otherwise you can't get the issued SSL certificate.  
If you can't receive certificate collection email, you can login your WoSign account to get the certificate.

Account password:

If you have WoSign account, this certificate will be included in your account after you have collected this certificate. We don't reset your account password to this one.

Confirm password:

Captcha: Enter the characters [change it](#)

I have read and agree to [WoSign Terms of Use Agreement](#)

Domain name  
(other cert. contents)

Verify domain control

PKCS#10 CSR

Contact + auth

CAPTCHA?

Subscriber Agreement

Contact + auth  
Agreement

Domain name

PKCS#10 CSR



"next"

"next"

new-reg ----+----> new-authz ----+----> new-cert

.

|

.

|

.

.

|

.

|

.

v

|

v

|

v

reg\* -----+

authz -----+

cert

. ^

. | "up"

v |

challenge



Verify domain control

No CAPTCHA  
(Standard DOS mitigation  
applies)

# Registration

POST /acme/new-reg HTTP/1.1

Host: example.com

```
{  
  "contact": [  
    "mailto:cert-admin@example.com",  
    "tel:+12025551212"  
  ],  
  "agreement": "https://example.com/terms"  
}
```

*/\* Signed as JWS with account key pair \*/*

HTTP/1.1 201 Created

Content-Type: application/json

Location: https://example.com/reg/asdf

```
{  
  "contact": /* copied */,  
  "agreement": /* copied */,  
  "key": /* JWS signature key */  
}
```

# Authorization

POST /acme/new-Authz HTTP/1.1

Host: example.com

```
{
  "identifier": {
    "type": "domain",
    "value": "example.org"
  }
}
/* Signed as JWS */
```

HTTP/1.1 201 Created

Content-Type: application/json

Location: https://example.com/authz/asdf

```
{
  "status": "pending",
  "identifier": {
    "type": "domain",
    "value": "example.org"
  },
  "key": { /* JWK from JWS header */ },
  "challenges": [ /* next slide */ ],
}
```

# Challenges

```
/* Challenge: CA ---> Applicant */  
{  
  "type": "simpleHttps",  
  "uri": "https://example.com/authz/asdf/0",  
  "token": "I1irfxKKXAsHtmzK29Pj8A"  
}
```

To prove you control `example.org` please provision a file containing the string `I1irfxKKXAsHtmzK29Pj8A` in the directory:

`https://example.org/.well-known/acme-challenge/`

```
/* Response: Applicant ---> CA */  
{  
  "path": "Hf5GrX4Q7EB"  
}  
/* Signed as JWS */
```

OK, I provisioned it to:

`https://example.org/.well-known/acme-challenge/Hf5GrX4Q7EB`

# Certificates

```
POST /acme/new-cert HTTP/1.1
Host: example.com
Accept: application/pkix-cert
```

```
{
  "csr": "5jNudRx6Ye4H...FS6aKdZeGsyso",
  "authorizations": [
    "https://example.com/acme/authz/asdf"
  ]
}
/* Signed as JWS */
```

```
HTTP/1.1 201 Created
Content-Type: application/pkix-cert Location:
https://example.com/acme/cert/asdf
```

```
[DER-encoded certificate]
```

# Summary

POST new-registration — registration + agreement URL

POST registration + agreement — OK

POST new-authorization — authorization + challenges

POST challenge + response — accepted

GET authorization — authorization valid

POST new-certificate — certificate

# Links

Specification:

<https://tools.ietf.org/html/draft-barnes-acme-01>

<https://github.com/letsencrypt/acme-spec>

Implementation:

<https://github.com/letsencrypt/boulder>

