# URL Shortener

## Use Case:

HTTP URLs can sometime grow to long length such that you might want to introduce the shorter version of the URL. The other reason would be to hide the details in a long URL.

## F5 iRules:

```
when RULE_INIT {
  set static::small_url_timeout 86400
  set static::small_url_lifetime 86400
  set static::small_url_response_header "<!DOCTYPE HTML PUBLIC \"-//IETF//DTD HTML 2.0//EN\"><html><head> \
    <title>Small URL Generator</title></head><body><center><h1>Small URL Generator</h1> \
    "
  set static::small_url_response_footer "</center></body></html>"
}


when HTTP_REQUEST {
  if { ([HTTP::uri] starts_with "/create?") and ([HTTP::query] ne "") } {
    set url [URI::decode [string tolower [URI::query [HTTP::uri] url]]]
    set custom_url_key [string tolower [URI::query [HTTP::uri] custom_url_key]]


    if { $custom_url_key ne "" } {
      if { ([table lookup -subtable small_url $custom_url_key] ne "") } {
        HTTP::respond 200 content "$static::small_url_response_header <b><font color=\"ff0000\"> \
          Error: the custom Small URL <a href=\"http://[HTTP::host]/$custom_url_key\"> \
          http://[HTTP::host]/$custom_url_key</a> has already been taken. Please try again. \
          </font></b> $static::small_url_response_footer"
      } else {
```

```
        set url_key $custom_url_key

        log local0. "Custom Small URL created for $url with custom key $url_ke
y"      }

    } else {

      switch -glob [string length $url] {

        {[1-9]} { set url_key_length 3 }

        {1[0-9]} { set url_key_length 3 }

        {2[0-9]} { set url_key_length 4 }

        {3[0-9]} { set url_key_length 5 }

        default { set url_key_length 6 }

      }


      set url_key [string tolower [scan [string map {/ "" + ""} [b64encode [md
5 $url]]] "%${url_key_length}s"]]

    }


    if { ([table lookup -subtable small_url $url_key] eq "") } {

      table add -subtable small_url $url_key $url $static::small_url_timeout $
static::small_url_lifetime

      log local0. "Small URL created for $url with key $url_key"

    } else {

      log local0. "Small URL for $url already exists with key $url_key"

    }


    HTTP::respond 200 content "$static::small_url_response_header The Small UR
L for \

      <a href=\"$url\">$url</a> is <a href=\"http://[HTTP::host]/$url_key\"> \

      http://[HTTP::host]/$url_key</a> $static::small_url_response_footer"

  } else {

    set url_key [string map {/ ""} [HTTP::path]]

    set url [table lookup -subtable small_url $url_key]


    if { [string length $url] != 0 } {

      log local0. "Found key $url_key, redirecting to $url"

      HTTP::redirect $url
```

```
    } else {
      HTTP::respond 200 content "$static::small_url_response_header <form acti
on=\"/create\" \
        method=\"get\"><input type=\"text\" name=\"url\">  \
        <input type=\"submit\" value=\"make small!\"><h4>Make it custom! \
        (optional)</h4>http://[HTTP::host]/<input type=\"text\" name=\"custom_
url_key\"></form> \
        $static::small_url_response_footer"
    }
  }
}
```

URL: https://devcentral.f5.com/codeshare/url-shortener

## NetScaler Solution:

```
add ns variable url_map -type map(text(1000), text(5000), 10000) -
expires 86400


add ns assignment assign_map -variable
$url_map[HTTP.REQ.URL.QUERY.VALUE("url").DIGEST(MD5)] -set
HTTP.REQ.URL.QUERY.VALUE("url")


add responder policy create_entries_pol
'HTTP.REQ.URL.PATH.EQ("/create") &&
HTTP.REQ.URL.QUERY.VALUE("url").LENGTH.GT(0)' assign_map


add responder action redirect_small_url_act redirect
'"http://"+HTTP.REQ.HOSTNAME+"/"+HTTP.REQ.URL.QUERY.VALUE("url").DIG
EST(MD5)' -bypassSafetyCheck YES


add responder policy redirect_small_url_pol
'HTTP.REQ.URL.PATH.EQ("/create") &&
HTTP.REQ.URL.QUERY.VALUE("url").LENGTH.GT(0)' redirect_small_url_act


Bind the responder policies "create_entries_pol" with a Higher
priority than the responder policy "redirect_small_url_pol" and
gotoPriorityExpression as NEXT
```

The policies can be bound to vserver or to Global Bind Point.

```
add rewrite action translate_url_act replace HTTP.REQ.URL
$url_map[HTTP.REQ.URL] -bypassSafetyCheck YES
```

```
add rewrite policy translate_url_pol
$url_map.valueExists(HTTP.REQ.URL) translate_url_act
```

Bind Rewrite policy to a vserver or global request bind point. Here we are creating a URL map in runtime and translating original URL with the mapped one.