

rex career portal integration via Reverse Proxy

rex Suite

rex systems GmbH

Suederstraße 75-79

D-20097 Hamburg

+49 (0) 40 / 8 900 800

www.rexx-systems.com

Contents

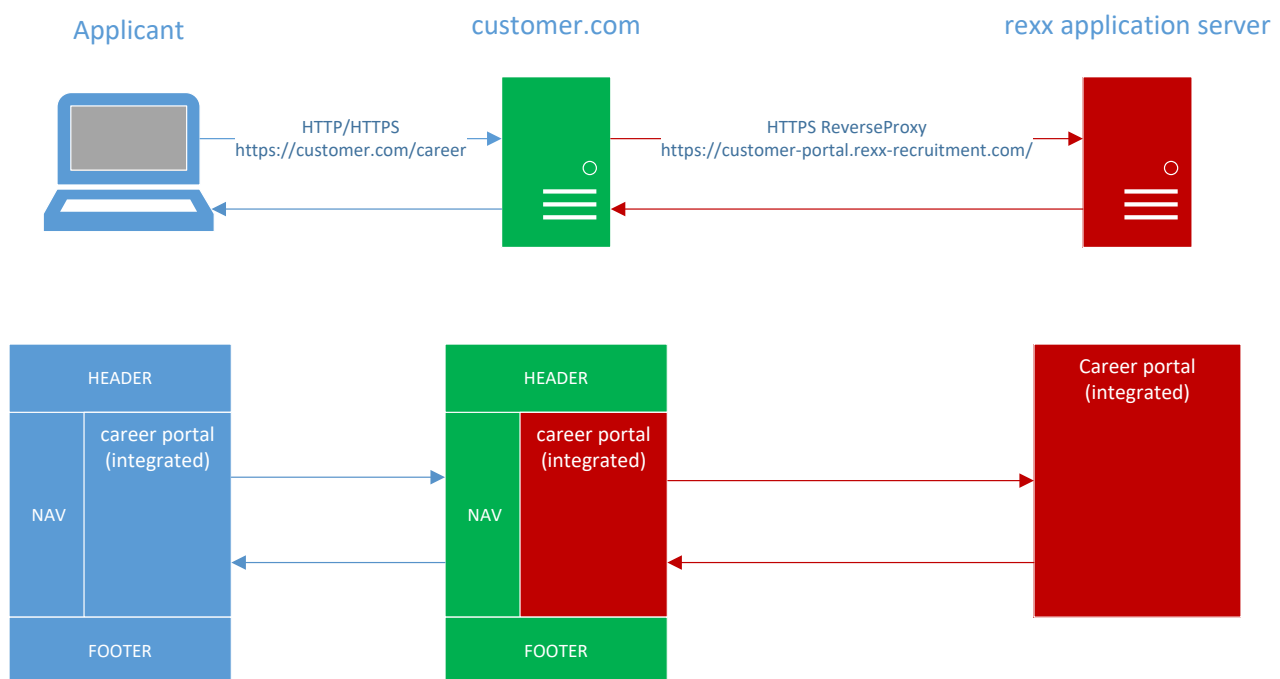
1.	INTRODUCTION.....	3
2.	FUNCTIONALITY	3
3.	REQUIREMENTS	4
4.	CONFIGURATION	5
4.1	Nginx >=1.6	5
4.2	Apache 2.4	6
4.3	Microsoft IIS.....	7

1. Introduction

If the rexx career portal is built into the customer's website using an iframe (hereinafter referred to as **customer.com**), then this can prove unfavorable to the relevance in search engines as well as the circulation of the job links to persons or services. In order to optimize the links and job ads for search engines, there is the option of using a reverse proxy to integrate the grabber or job portal into the existing website **customer.com**. Thereby content is no longer requested from the rexx server, rather directly via the webserver from **customer.com**. All links will therefore be generated starting with e.g. <https://customer.com/career>.

2. Functionality

In order to fully integrate the rexx career portal into the customer's website (e.g. **customer.com**), the web server that operates the website **customer.com** has to integrate the rexx career portal using a reverse proxy. However, since the rexx job portal represents only a portion of the website, the remaining available content, such as for example left navigation, header and footer of the outlay must be added. This occurs with other modules of the web server.



Adding additional content, as indicated above, can be carried out on both web servers. When content added to **customer.com**, it is possible to automatically generate the HTML code dynamically via the utilized CMS system. We refer to this as dynamic content. When content is added to the rexx application server, we refer to this as statistical content, because the HTML code must first be submitted to the rexx application server. An automated periodical transfer of the HTML code is possible.

3. REQUIREMENTS

For a dynamic integration to **customer.com** the Apache webserver or the Nginx webserver is mandatory.

If Microsoft IIS is being used on **customer.com**, the integration can only take place statically on the rexx application server. The rexx career portal needs its own namespace, because of dynamically generated links. Working: "customer.com/career/", not working: "/career/vacancies.html".

Dynamic integration:

Minimal requirement is Nginx 1.6.xx or Apache 2.4.xx. Full access to apaches, or nginx's configuration is needed. To implement the configuration using a ".htaccess file" is not possible!

Nginx (>=1.6.xx) with the following modules:

- ngx_http_ssi_module
- ngx_http_sub_module (ngx_http_substitutions_filter_module)
- ngx_http_proxy_module

Apache 2.4.xx with the following modules:

- mod_proxy
- mod_proxy_http (Submodul von mod_proxy)
- mod_include (ehemals mod_ssi)
- mod_headers
- mod_rewrite
- mod_sed

If it is not possible to activate or install the aforementioned modules on the web server **customer.com**, then the integration can only take place statically via the rexx application server.

Static integration:

If the content is statically integrated, Microsofts IIS, Apache webserver or Nginx webserver can be used.

In Microsofts IIS at least version 7.0 is required, which was supplied with Windows Server 2008 R8.

Additionally the following modules have to be retrospectively installed:

- URL Rewrite Module 2.0 ([Link to Module download on iis.net](#))
- Web Farm Framework 2.2 ([Link to Module download on iis.net](#))
- Application Request Routing (>ARR 2.0) ([Link to Module download \(ARR 3.0\) on iis.net](#))

We do not recommend installing the modules individually, rather to carry out the installations via Microsoft's web platform installer, as in this case dependencies are installed.

If the website customer.com uses and includes jQuery, this must be prevented under the namespace of the jobportal , i.e. “https://customer.com/career”.

4. Configuration

4.1 Nginx >=1.6

The following directives need be added to the server section.

```
# SSL/TLS configuration of "kunde.de"
server {

    location /career/ {
        proxy_pass https://customer-portal.rexx-systems.com/;
        proxy_redirect default;
        proxy_set_header Host $http_host;
        subs_filter_types text/html;
        subs_filter '<!--#[a-z].*' ' ' ri;
        subs_filter '<body(.*)>' '<body$1><!--#include virtual="/rex-layout/body-top.inc.php"-->'
ir;
        subs_filter '</head>' '<!--#include virtual="/rex-layout/head.inc.php"--></head>' i;
        subs_filter '</body>' '<!--#include virtual="/rex-layout/body-bottom.inc.php"--></body>'
i;
        proxy_set_header Accept-Encoding "";
        ssi on;
    }
}
```

4.2 Apache 2.4

The following directives need to be added to the virtual host of the website “customer.com”.

```
<VirtualHost *:443>
    ProxyPreserveHost on
    SSLProxyEngine on
    SSLProxyVerify off
    SSLProxyCheckPeerCN off
    SSLProxyCheckPeerName off
    SSLProxyCheckPeerExpire off
    ProxyPass /career https://customer-portal.rexx-systems.com
    ProxyPassReverse /career https://customer-portal.rexx-systems.com

    # Pre used rewrite rules must be stopped by the following rule
    # RewriteRule ^(@career)($|/) - [L]

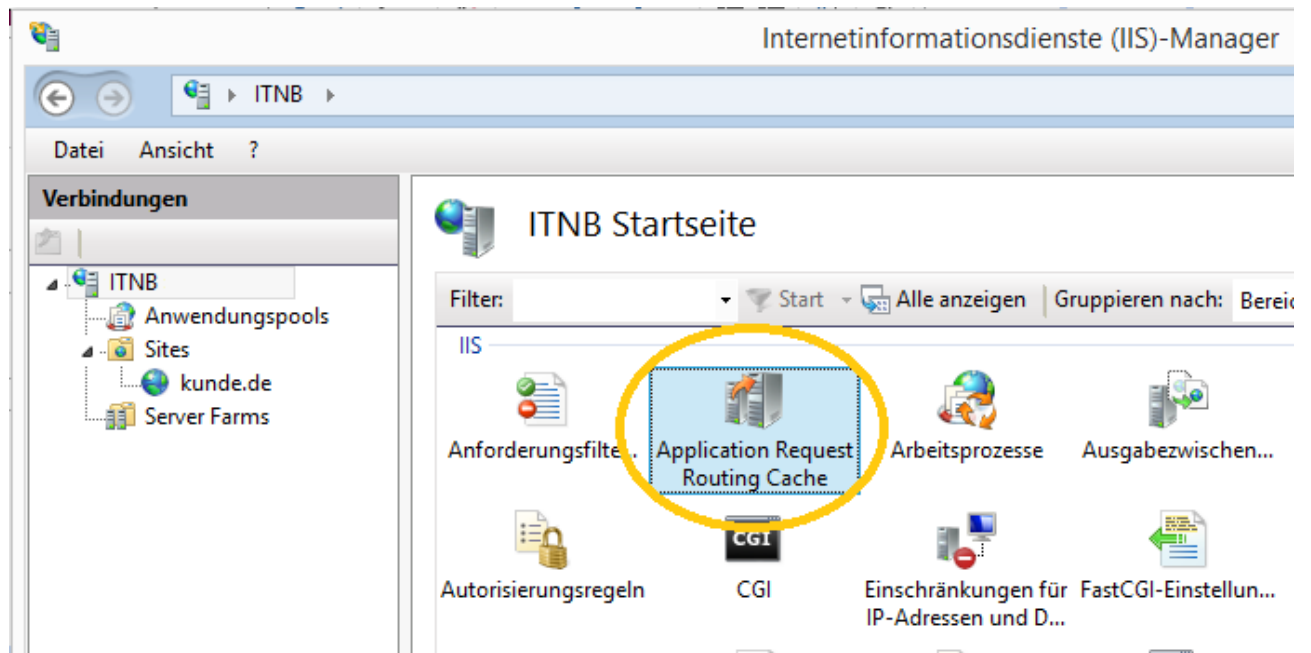
    <Location /karriere>
        <IfModule mod_headers.c>
            RequestHeader unset accept-encoding
            Header unset Content-Length
        </IfModule>
        Options +Includes
        AddOutputFilterByType Sed;INCLUDES text/html

        # exclude attachment preview
        <If "%{REQUEST_URI} !~ /dsa0.php/">
            OutputSed "/<!--#[a-zA-Z]/d"
            OutputSed "s/\\(<\\head>\\)/\\<!--#include virtual=\\\"\\rexx-layout\\head.inc.php\\\"--"
        </If>
    </Location>
</VirtualHost>
```

As with Apache 2.2xx the above configuration specifies that the file “head.inc.php” with HTML meta-tags, the file “body-top.inc.php” with the HTML code for the header area (e.g. navigation, header) and the file body-bottom.inc.php for the HTML code for the footer area are stored in the directory “rex-layout”.

4.3 Microsoft IIS

Activation of the proxy function in the ARR module





Application Request Routing

Use this feature to configure proxy settings for Application Request Routing

☒ Enable proxy

Proxy Setting

HTTP version:

Pass through

☒ Keep alive

Time-out (seconds):

120

☒ Reverse rewrite host in response headers

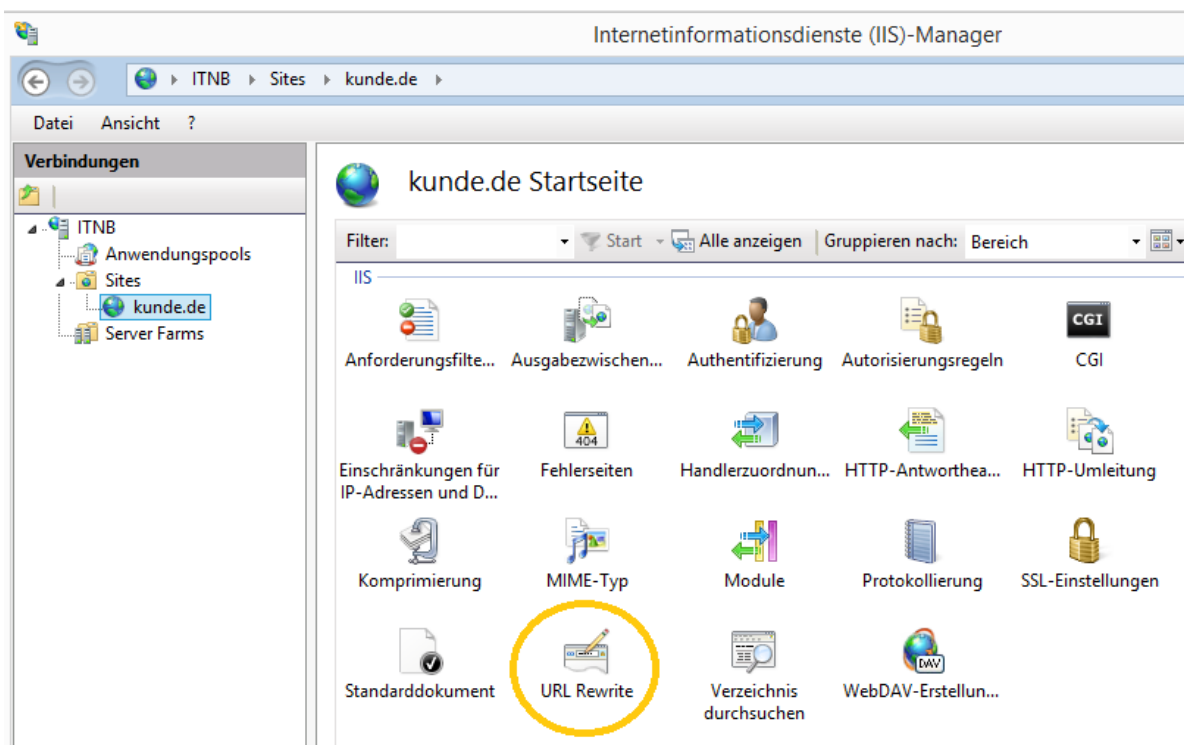
Custom Headers

Preserve client IP in the following header:

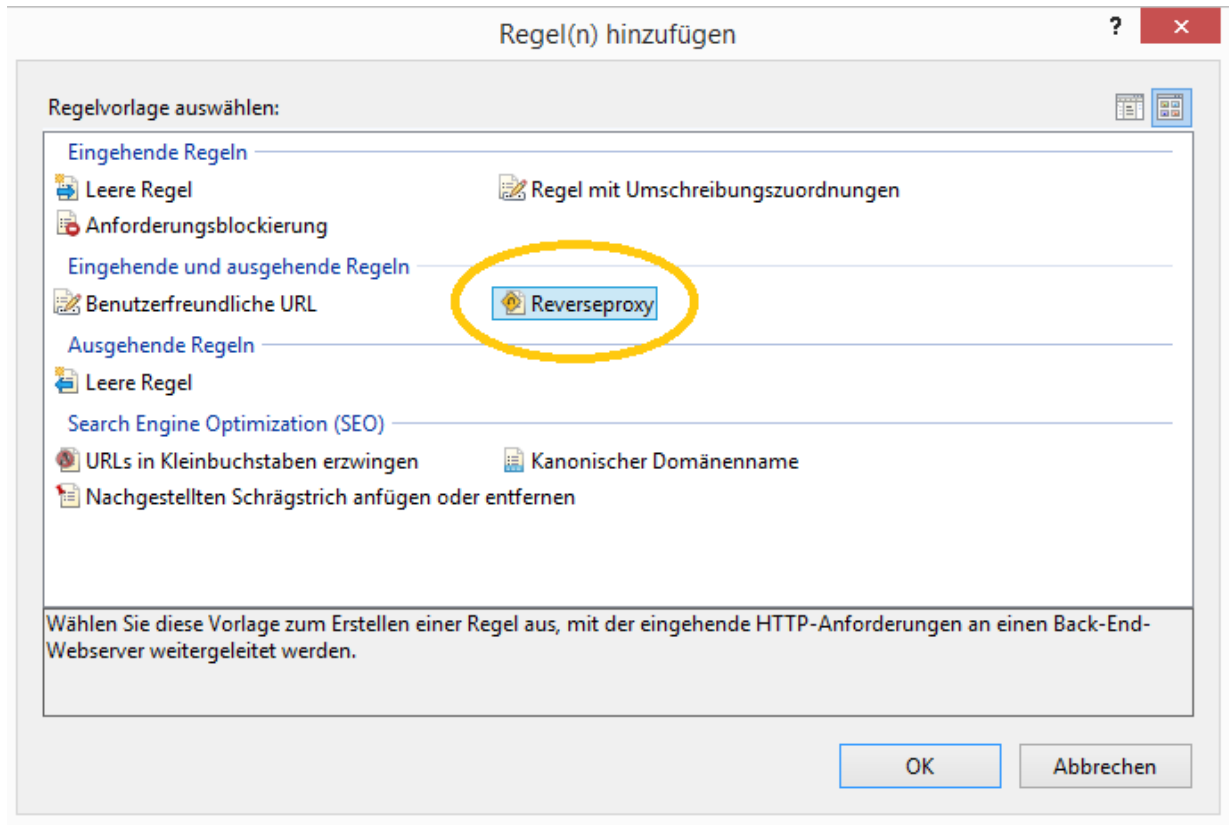
X-Forwarded-For

☒ Include TCP port from client IP

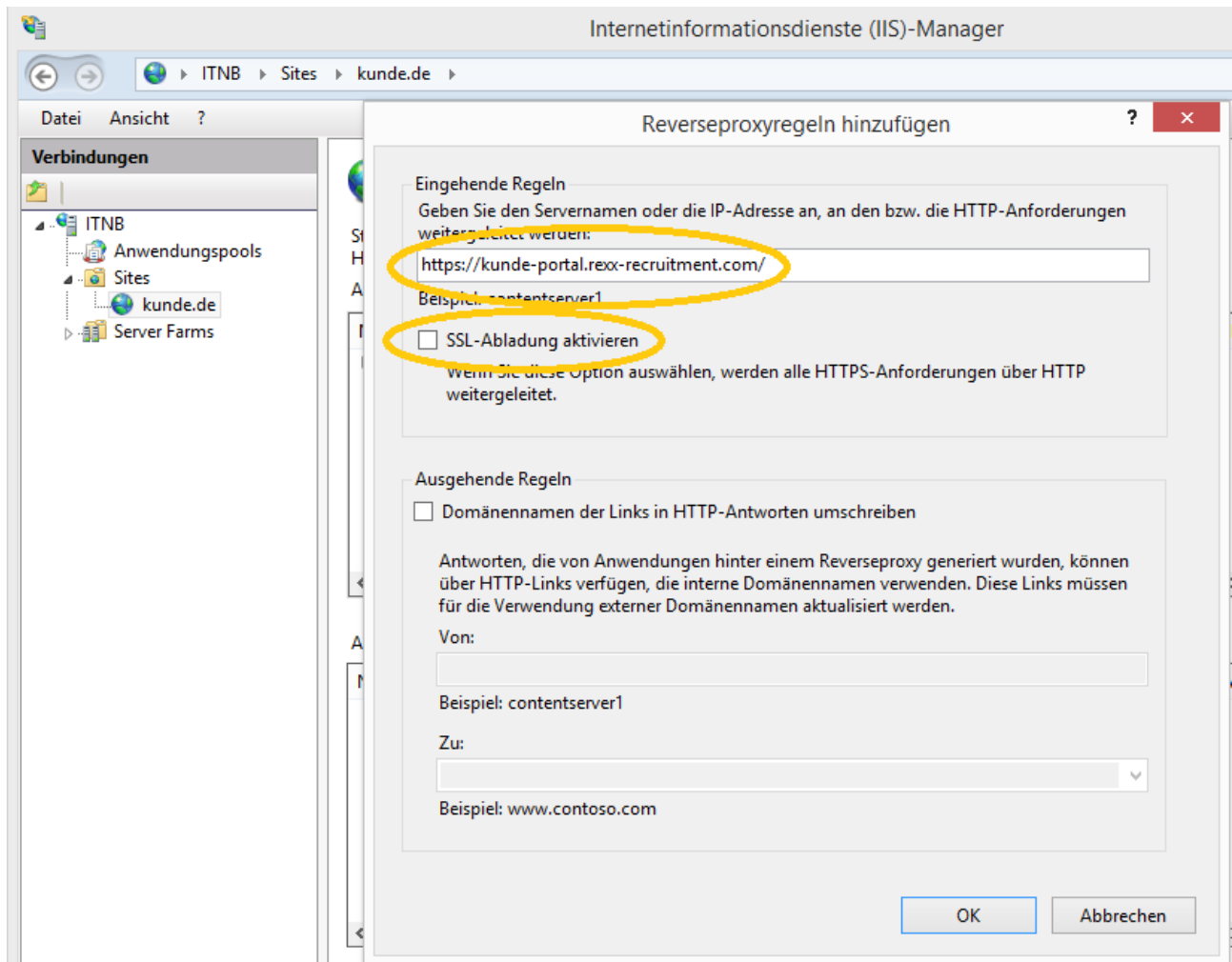
Switch the settings of the URL rewrite module on the website "customer.com"



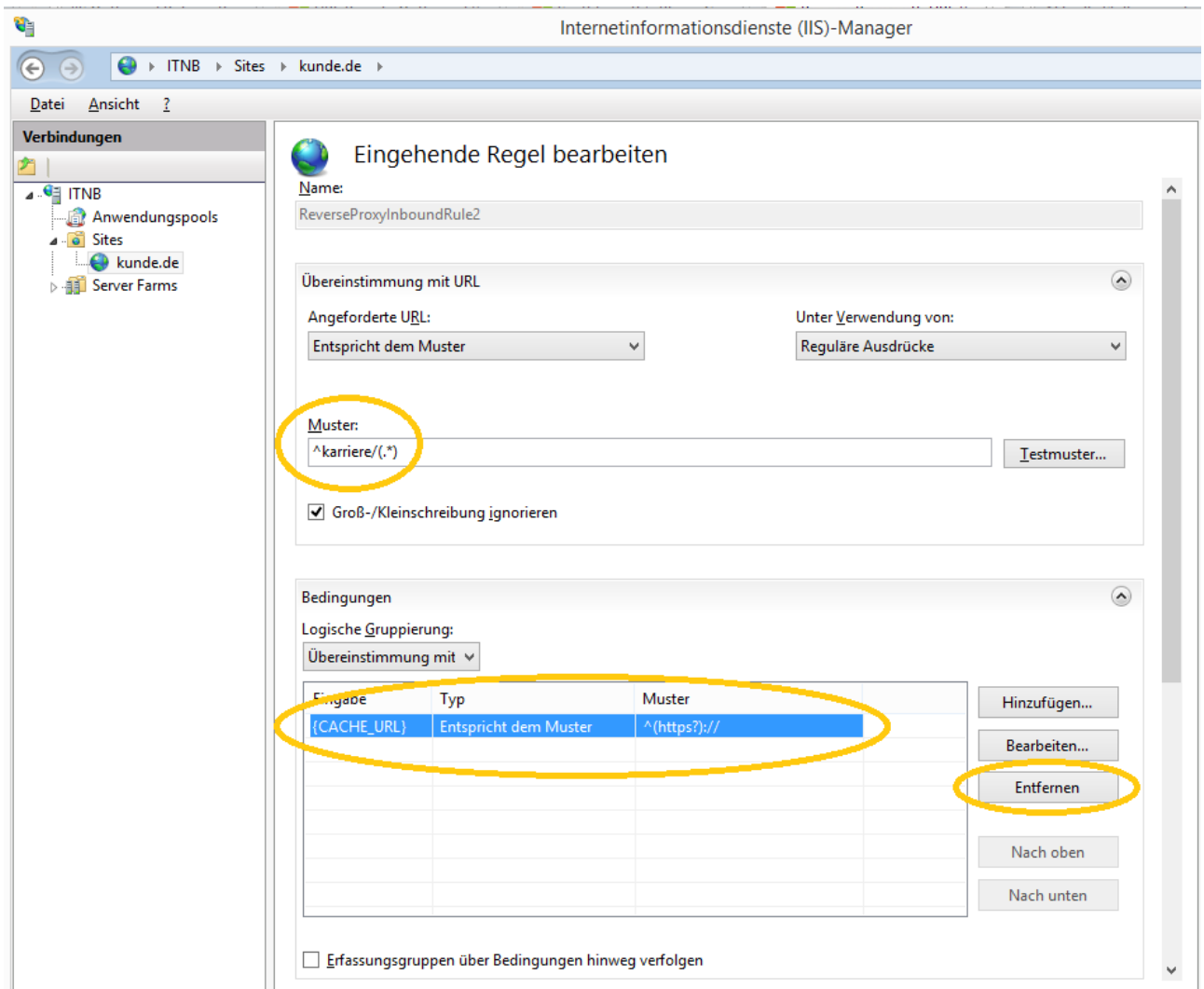
About „**Add new rule**“ add a reverse proxy rule



Consign target address „<https://kunde-portal.rexx-systems.com/>“ and deactivate **SSL-Upload**



Subsequently edit incoming rules – edit design and remove condition



Internetinformationsdienste (IIS)-Manager

ITNB > Sites > kunde.de >

Verbindungen

ITNB

- Anwendungspools
- Sites
 - kunde.de
- Server Farms

Eingehende Regel bearbeiten

Name: ReverseProxyInboundRule2

Übereinstimmung mit URL

Angeforderte URL: Entspricht dem Muster

Unter Verwendung von: Reguläre Ausdrücke

Muster: ^karriere/(.*) Testmuster...

☒ Groß-/Kleinschreibung ignorieren

Bedingungen

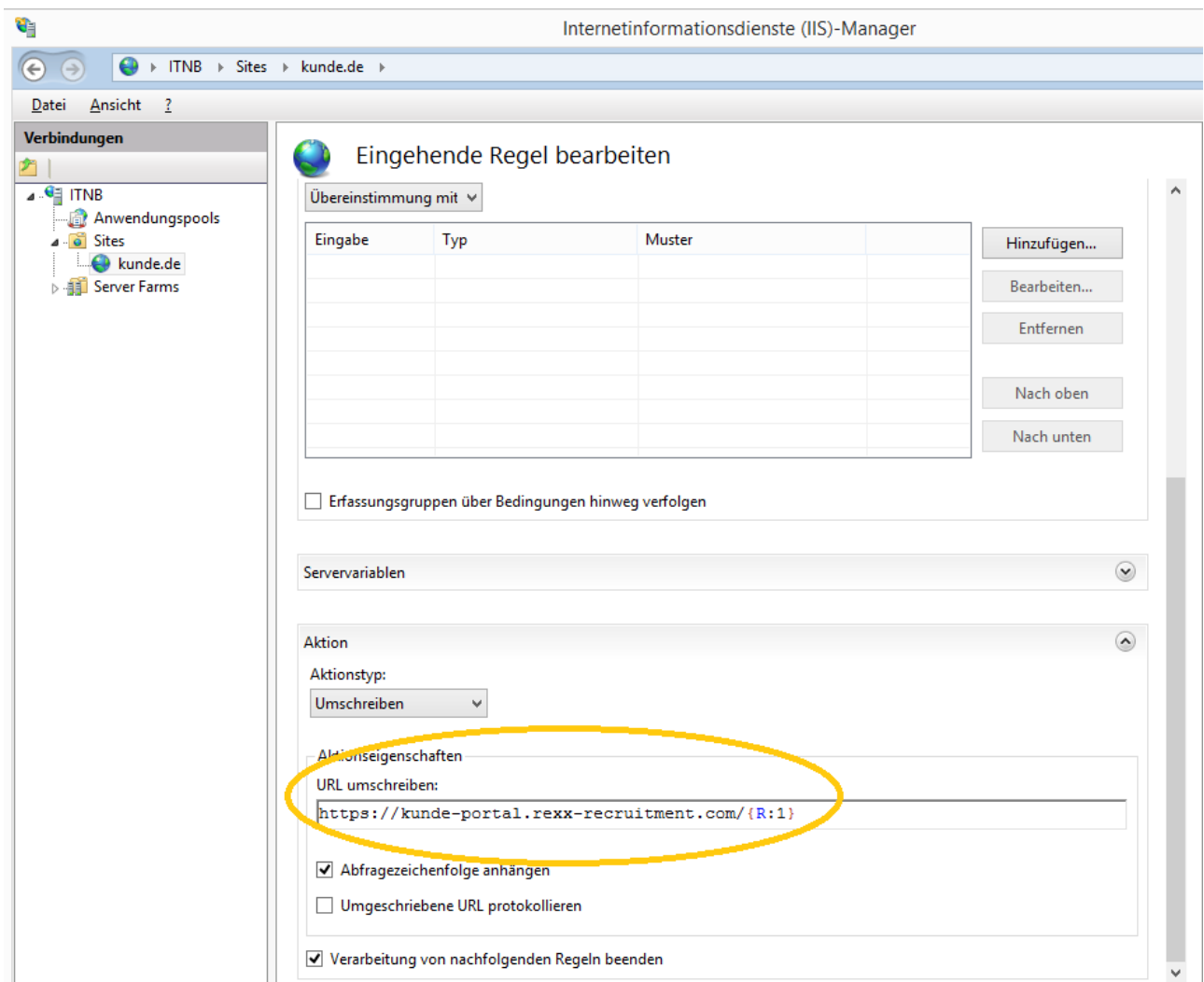
Logische Gruppierung: Übereinstimmung mit

Eingabe	Typ	Muster
CACHE_URL	Entspricht dem Muster	^(https?://

Hinzufügen...
Bearbeiten...
Entfernen
Nach oben
Nach unten

☐ Erfassungsgruppen über Bedingungen hinweg verfolgen

Subsequently edit incoming rules – customize Action URL re-write (as in Fig.)



Afterwards, the IIS-site “customer.com” must be restarted.

The directory of the page in the configuration file “web.config” must, at least upon further configurations, accordingly contain the following content:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <system.webServer>
    <rewrite>
      <rules>
        <rule name="ReverseProxyInboundRule1" stopProcessing="true">
          <match url="^karriere/(.*)" negate="true" />
          <action type="Rewrite" url=" https://customer-portal.rexx-systems.com
/{R:1}" />
        </rule>
      </rules>
    </rewrite>
  </system.webServer>
</configuration>
```

The loading of the page “<https://customer.com/career>” should now be visible in both the job portal and the job listings of the rex application server. The statically added HTML code now has to be passed on to the relevant rex consultant. The rex consultant then coordinates the integration into the rex career portal.