You can manage $language\_suffix by this setting when you cannot add AcceptLanguageModule module into your system.

rewrite (.\*) $1/$http\_accept\_language

A more resilient approach would use a map:

map $http\_accept\_language $lang {

default en;

~es es;

~fr fr;

}

...

rewrite (.\*) $1/$lang;

This mapping does not honor the preferences. To imporove this mapping you can change the regex from ~es es; to ~\*^es es; and this is going to map the language only to the first language appearing on the Accept-Language header (assumming the browser delivers the prefered language first, wich I think is very frequent)

prefer Lua because it can be installed easily in debian based distros:

apt-get install nginx-extras

The downside of using AcceptLanguageModule is you cannot rely on automatic system updates anymore. And with every nginx update (even security one), you have to compile Nginx yourself. The second downside is that module assumes that the accept-language is sorted by quality values already. I rather prefer Lua because it can be installed easily in debian based distros:

apt-get install nginx-extras

My colleague Fillipo made great [nginx-http-accept-lang](https://github.com/fghibellini/nginx-http-accept-lang) script in Lua. It correctly handles quality values and does redirect user accordingly. I've made [small modification](https://gist.github.com/mauron85/47ed1075262d9e020fe2) to that script. It accepts supported languages as input parameter and returns the most qualified language according to Accept-Language header. With returned value you can do whatever you want. It can be used for rewrites, setting lang cookie ...

I'm only using language determination for root path only (location = /). And user lang cookie has preference over browser. My nginx conf looks like this:

map $cookie\_lang $pref\_lang {

default "";

~en en;

~sk sk;

}

server {

listen 80 default\_server;

root /usr/share/nginx/html;

index index.html index.htm;

# Make site accessible from http://localhost/

server\_name localhost;

location = / {

# $lang\_sup holds comma separated languages supported by site

set $lang\_sup "en,sk";

set\_by\_lua\_file $lang /etc/nginx/lang.lua $lang\_sup;

if ($pref\_lang) {

set $lang $pref\_lang;

}

add\_header Set-Cookie lang=$lang;

rewrite (.\*) $scheme://$server\_name/$lang$1;

}

location / {

# First attempt to serve request as file, then

# as directory, then fall back to displaying a 404.

try\_files $uri $uri/ =404;

}

}