

[Documentation Home](#)

## Contents

- [Configuring SES URL's](#)
- [Modify Apache Config](#)
- [Tomcat web.xml Config](#)

## Configuring SES URL's

Search Engine Safe (SES) URL's are very cool for getting all of your site indexed in a meaningful way by a lot of search engines. It's also a handy way of linking to various aspects of your site in a way that looks good.

Using the Railo Installer along with SES URL's is a relatively simple process, once you understand how each processes involved works. Essentially, when working with SES URL's you have to remember that when a request first comes in, it hits Apache. If things are set up right, it will hit Apache, get passed to Tomcat, and subsequently get passed to Railo. You just need to make sure both Apache and Tomcat understand WHICH URL's to pass on to Railo in order for SES URL's to work.

## Modify Apache Config

You can control what URL patters are passed off to Tomcat by updating the main Apache Config file. This is usually httpd.conf in RedHat-based systems (like CentOS) or apache2.conf in Debian-based systems (like Ubuntu). Further, you'll need a part of the URL to "Key" off of. This "key" will tell Apache that this is a URL that needs to be passed off. CMS systems, such as [Mura](#), give you the ability to specify a "site" for your SES URL. This site name makes a perfect "key".

So, with this in mind, if your SES URL is suppose to look like this:

```
http://getrailo.org/site/contact/business/
```

Then your Apache config file might look something like this:

```
JkMount /*.cfm ajp13
JkMount /*.cfc ajp13
```

```
JkMount /*.do ajp13
JkMount /*.jsp ajp13
JkMount /*.cfchart ajp13
JkMount /*.cfm/* ajp13
JkMount /*.cfml/* ajp13

# ADD MY SES SITE KEY
JkMount /site/* ajp13

# Flex Gateway Mappings
# JkMount /flex2gateway/* ajp13
# JkMount /flashservices/gateway/* ajp13
# JkMount /messagebroker/* ajp13
JkMountCopy all
JkLogFile [log directory]/mod_jk.log
```

Note that in the above example, we're passing every request that comes in under the "site" key off to Tomcat. You'll probably still want Apache to handle the simple stuff like serving images and so forth, as it saves threads being started unnecessarily by Tomcat.

If you do not wish to use a site "Key", that's okay, but you will need to send EVERY request off to Tomcat. This means you will not be able to do things like process PHP pages, etc, from this server. You will need to choose how important having a site "Key" is compared to how important it is to processing PHP on the same server. Apache will serve little to no purpose in this scenario, as every request it gets will be passed to Tomcat.

There is a way around this, with a site-specific mod\_jk configuration, but I'm not going to go into the complex details of that unless there is an interest. Please post a request on the Railo mailing list if you're interested in knowing how to do this.

## Tomcat web.xml Config

For Tomcat, we will need to create a web.xml file that is site-specific, that way you can create mappings that only apply to a single site and not every site that's hosted by Tomcat. To do this, you will need to create the web.xml file within the WEB-INF file that railo creates for a new site. The process goes something like this:

1. The first step is to modify the server.xml file with the information for the new site [Explained Here](#)
2. Restart Tomcat so that the WEB-INF directory is created in the "DocBase" path, or create the directory yourself
3. Once the WEB-INF directory is created, create the web.xml file inside it. IE:  
/home/user/mysite.com/WEB-INF/web.xml  
Once you've created the file, add something similar to the following to create the site-

4.

specific mappings:

### **For Mura**

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
CFMLServlet  
/index.cfm/*  
/default/index.cfm/*
```

### **For Mango Blog**

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
CFMLServlet  
/blog/post.cfm/*  
/blog/page.cfm/*  
/blog/archives.cfm/*  
/blog/feeds/rss.cfm/*
```

You can add and remove additional attributes as needed by your specific application.

Once you've added the web.xml files to the site, you may need to restart Tomcat for the updated configs to take full effect.

[Documentation Home](#)