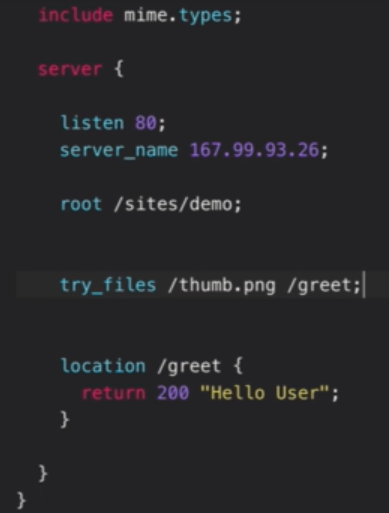
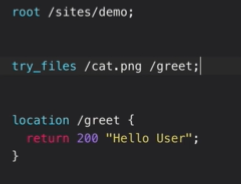
**Try files:**

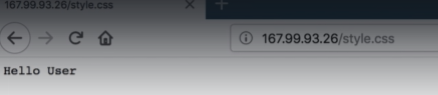




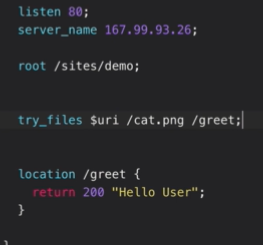
* As above, try file works like first it checks for the **“/thumb.png”.** if it exists, nginx will serve the request. If it doesn’t then it will check for **“/greet”**
* It doesn’t matter what the URI we are giving the URL, even if we search with **“/something”,** it will give the **“thumb.png”** image in browser.
* And even if we search with example of **“style.css”** which does exist in the code. But still nginx will serve the request with **“thumb.png”**



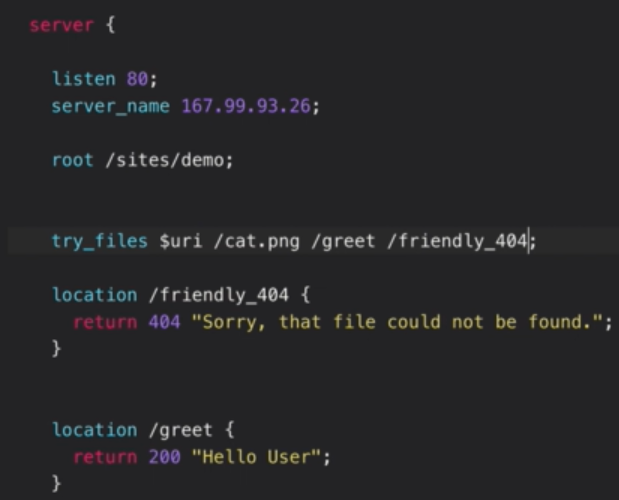
* As if we change the try file with something that doesn’t exist, then it will for **“/greet”** because **“/cat.png”** doesn’t exist.



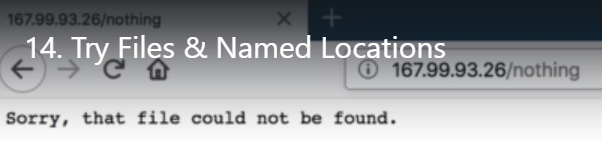
* Try files also rewrites the request, it doesn’t redirect the URL but it internally mutates the location.



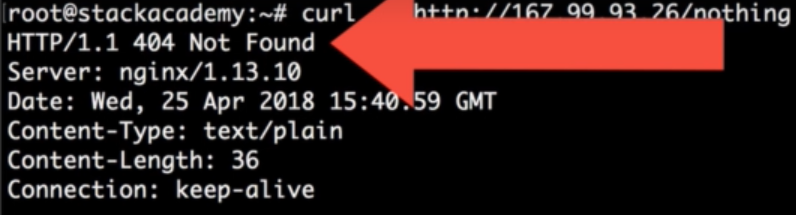
* As if we pass **“$uri”** as above, which means before checking anything it will confirm gets the data with uri we have given in URL. If that exists like **“thumb.png” or style.css” or “index.html”.** it will give the data as we requested
* If that uri doesn’t exists, then only it will go for the other condition in try\_files.



* As above, we can also give some friendly message to the client. If all the conditions fail as per the try\_files. Then it would give 404 error. We can add a message to the error.

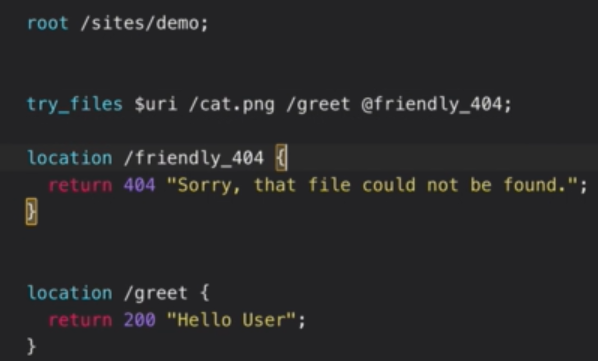


* When we curl the same URL. We get normal 404 error.



**Named locations:**

* Naming locations means simply giving names to the location contexts. And use them with try\_files.



* Same as normal try files but we need to add the conditions with **“@”** location name. and if is found then the request will be servers as per the location context.
* As above, $uri reads whatever the context the client is giving in the URL will be read over here and if that location is exists then the it will redirect to that location.

