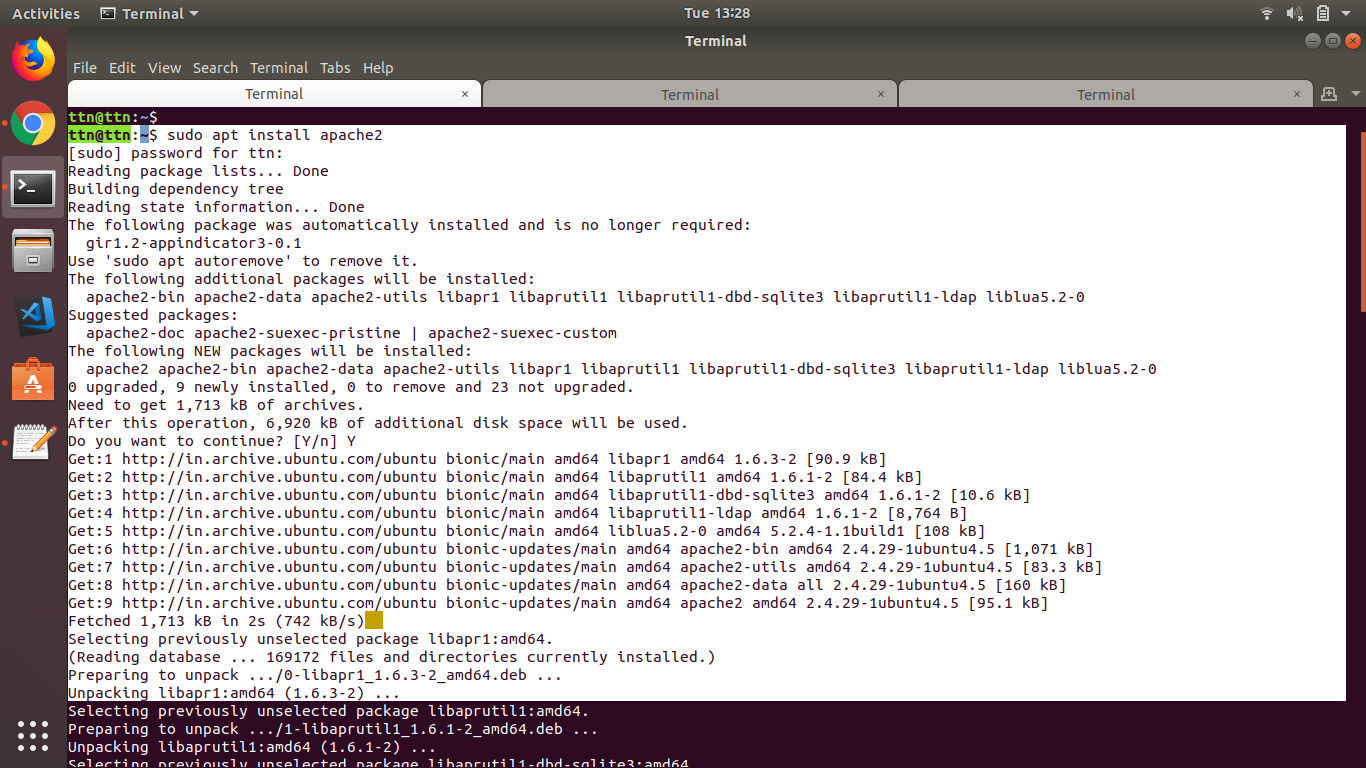
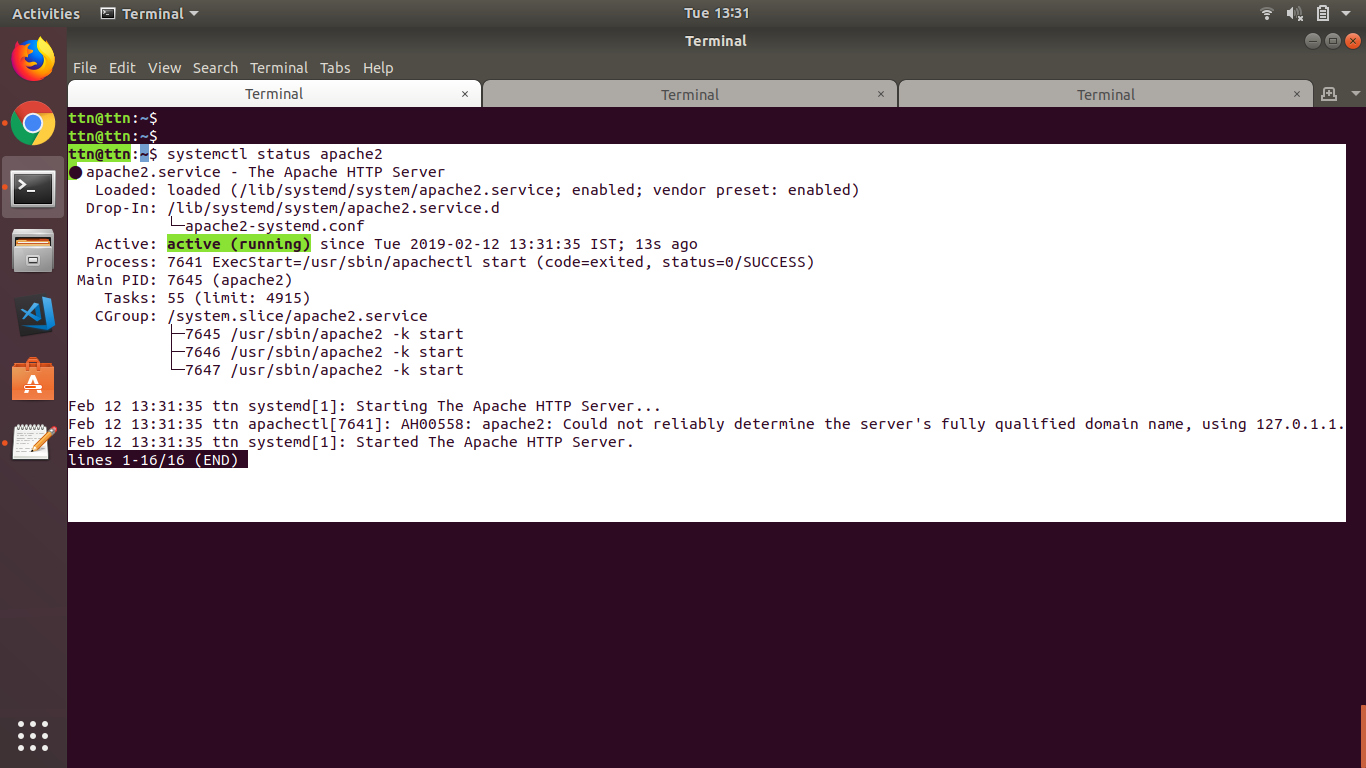
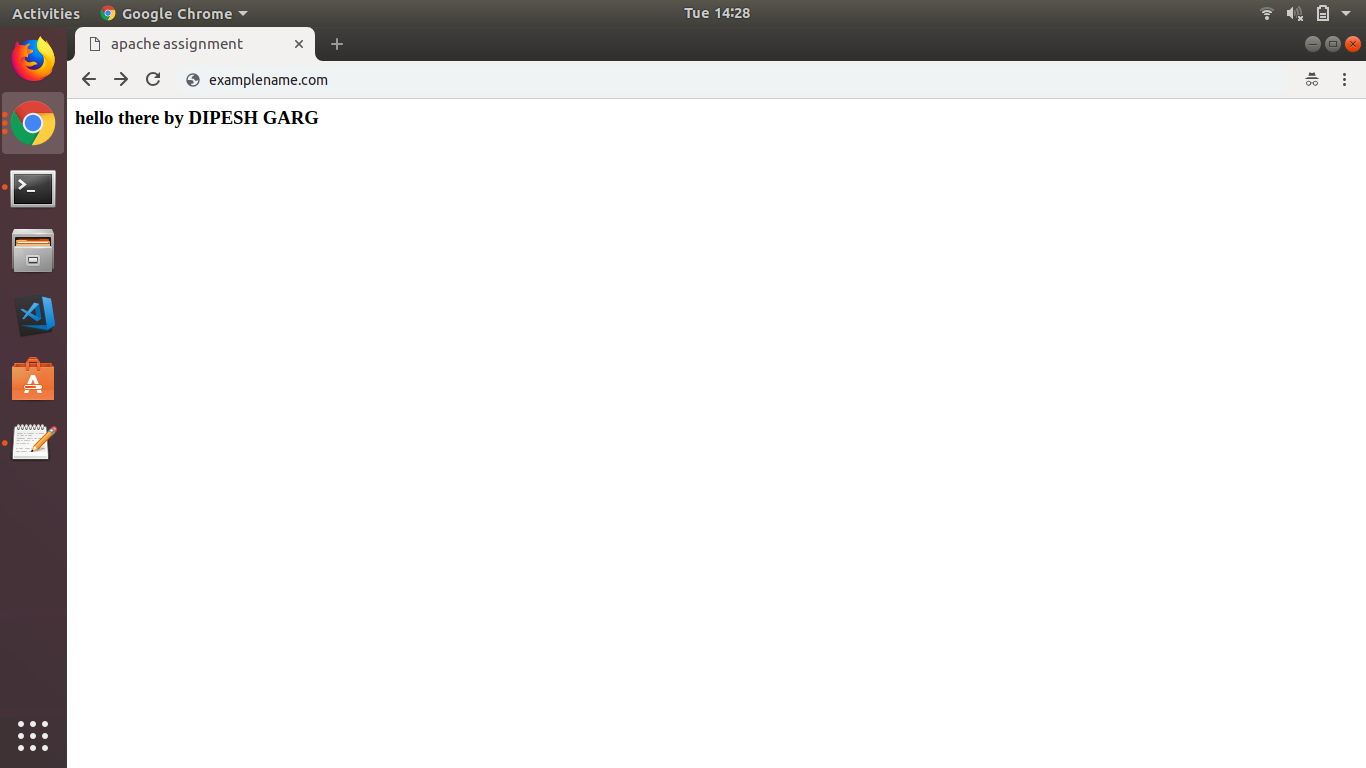
* **Install Apache**



Changed port to 8080 from 80 by vim /etc/apache2/ports.conf

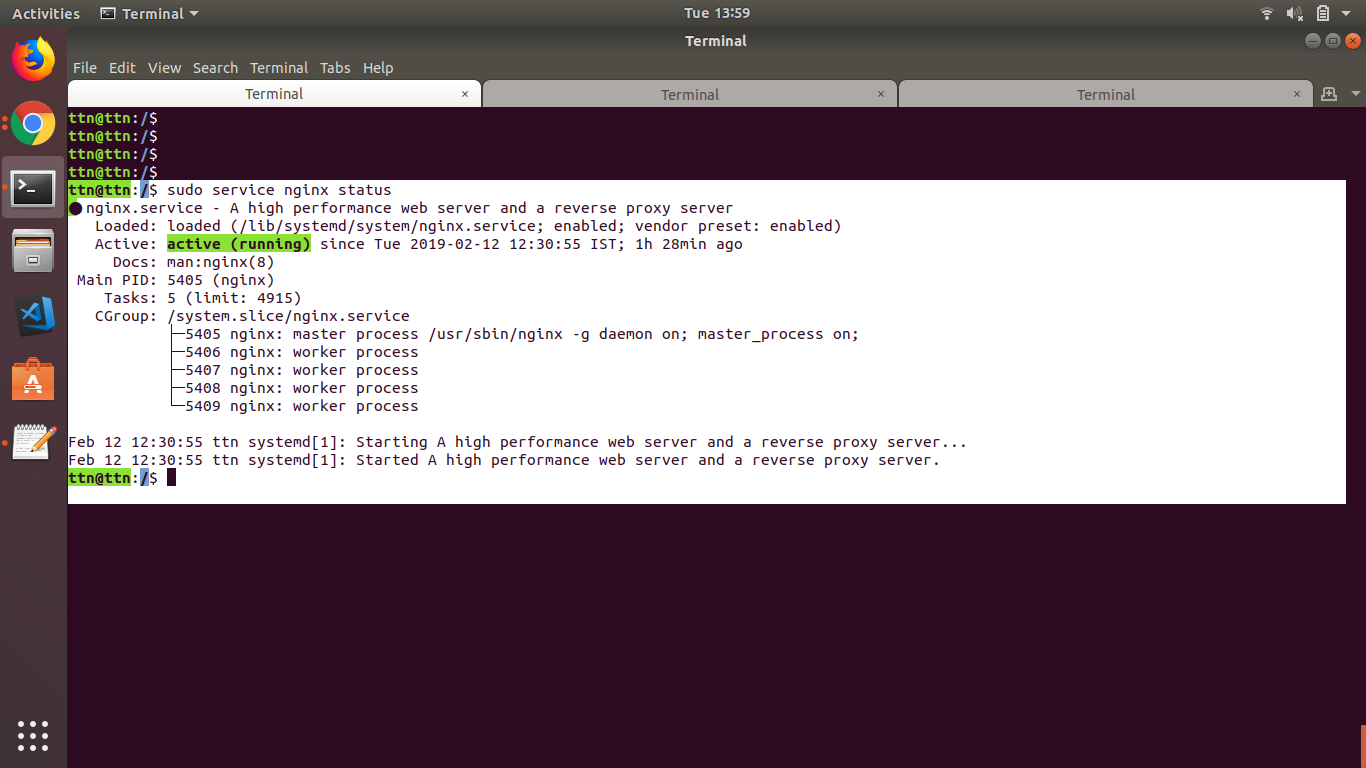


* + Create custom HTML PAGE, Print your name on it?



* **Install Nginx**

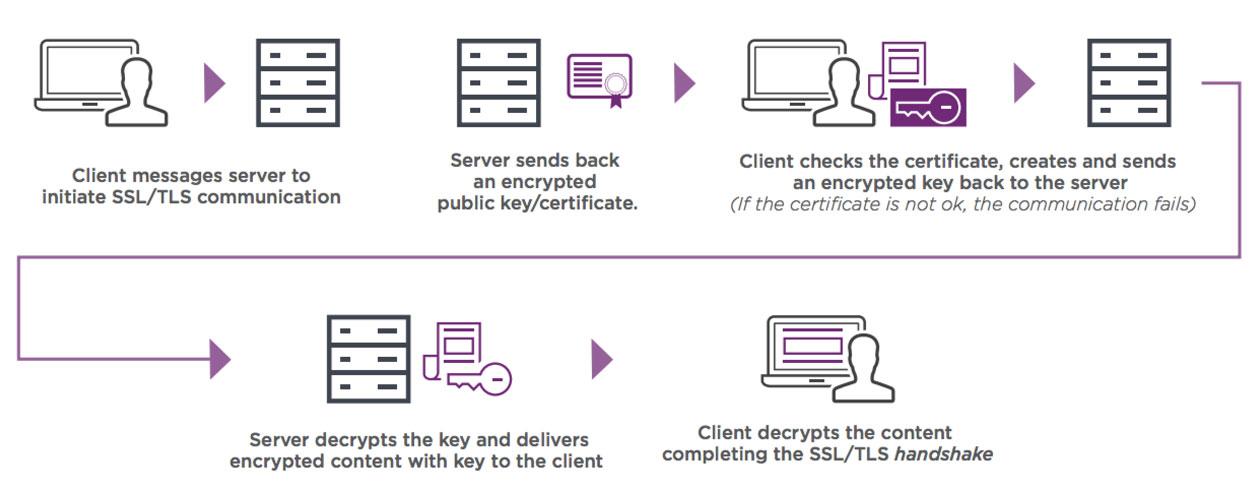
**sudo apt install nginx**

****

* **PUT Vs POST**

|  |  |
| --- | --- |
| **PUT** | **POST** |
| PUT method requests for the enclosed entity be stored under the supplied Request-URI. If the Request-URI refers to an already existing resource – an update operation will happen, otherwise create operation should happen if Request-URI is a valid resource URI (assuming client is allowed to determine resource identifier). | The POST method is used to request that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line. It essentially means that POST request-URI should be of a collection URI. |
| PUT /questions/{question-id} | POST /questions |
| PUT method is idempotent. So if you send retry a request multiple times, that should be equivalent to single request modification. | POST is NOT idempotent. So if you retry the request N times, you will end up having N resources with N different URIs created on server. |
| you can cache the response. | Responses to this method are not cacheable, unless the response includes appropriate Cache-Control or Expires header fields. |

* **How Does SSL work?**
* A browser or server attempts to connect to a website (i.e. a web server) secured with SSL. The browser/server requests that the web server identify itself.
* The web server sends the browser/server a copy of its SSL certificate.
* The browser/server checks to see whether or not it trusts the SSL certificate. If so, it sends a message to the web server.
* The web server sends back a digitally signed acknowledgement to start an SSL encrypted session.
* Encrypted data is shared between the browser/server and the web server.



* **List out five difference in Apache and Nginx**

|  |  |
| --- | --- |
| **Apache** | **Nginx** |
| Highly customizable architecture. | Difficult to customize the modules suitable to server due to complex base architecture. |
| Slower. | 2.5x faster than apache. |
| Apache runs on all kinds of Unix-like systems (e.g., Linux or BSD) and has full support for Microsoft Windows. | NGINX also runs on several modern Unix-like systems and has some support for Windows, but its Windows performance is not as strong as that of other platforms. |
| Excellent community through widespread user base. | NGINX has community support through mailing lists, IRC, Stack Overflow, and a foru. |
| A single thread can only process one connection. | A single thread can handle multiple connections. |

* **What is Virtual HOST?**

The term Virtual Host refers to the practice of running more than one web site (such as company1.example.com and company2.example.com) on a single machine. Virtual hosts can be "IP-based", meaning that you have a different IP address for every web site, or "name-based", meaning that you have multiple names running on each IP address. The fact that they are running on the same physical server is not apparent to the end user.

Name-based virtual hostings are used for hosting various virtual websites to one IP address. For configuring such type virtual hostings, it is required to install IP address through which the Apache requests will be received from all necessary websites. One may add as many websites to the name-based virtual hostings as it will be required.

In order to launch IP-based virtual hosting, it will need more than one address with the configuration on the required server. Thus, the number of hostings will depend on the IP address numbers. For example, if the server has 5 IP addresses, then one can create 5 virtual hostings on their IP base.

<VirtualHost>

ServerName

ServerAlias

ServerPath

DocumentRoot

</VirtualHost>

* **What is Document ROOT?**

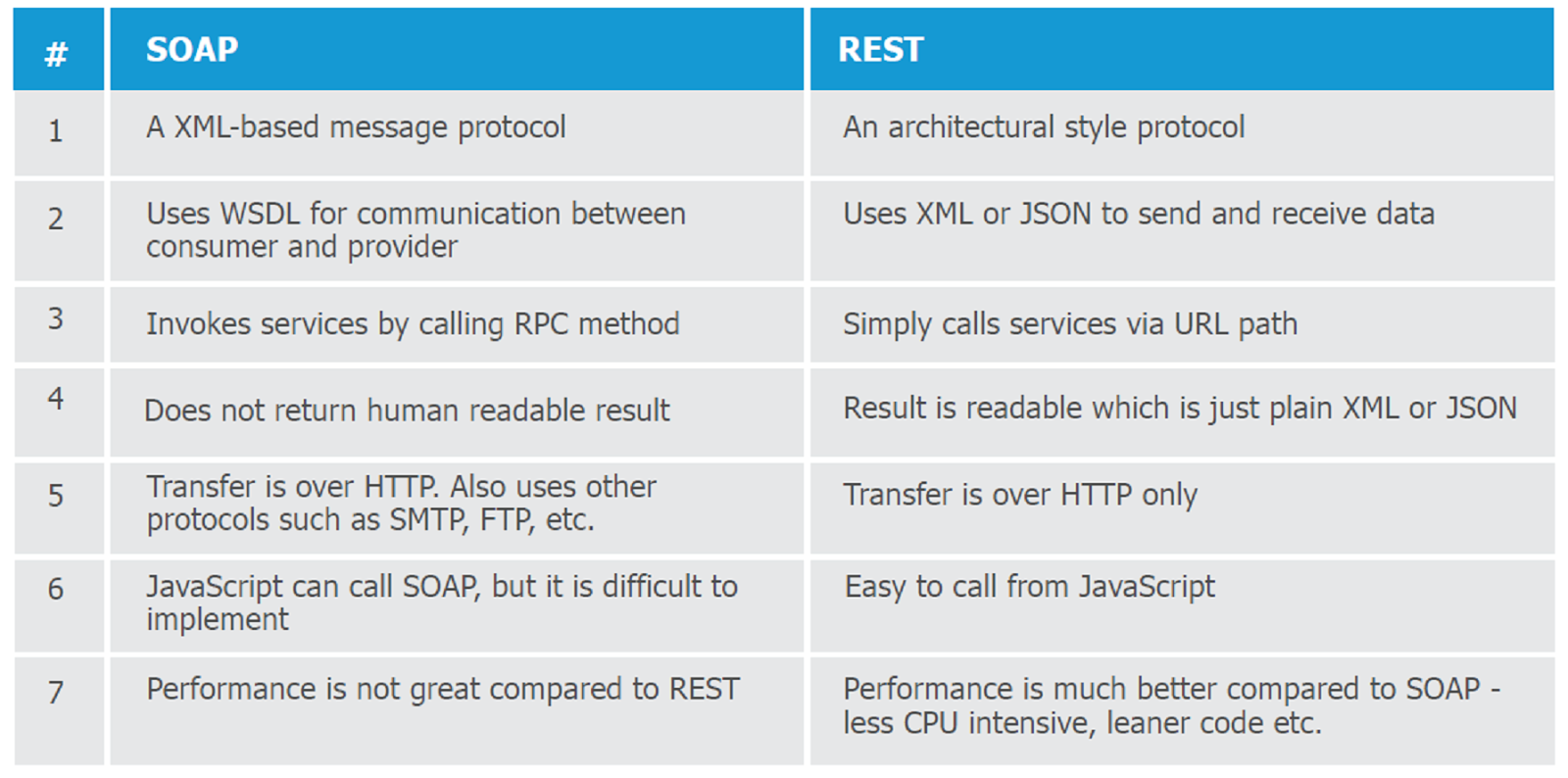
The document root is a directory (a folder) that is stored on your host's servers and that is designated for holding web pages. When someone else looks at your web site, this is the location they will be accessing.

In order for a website to be accessible to visitors, it must be published to the correct directory, the "document root."

* **List out five difference in REST and SOAP web service?**

**SOAP** – SOAP is a protocol which was designed before REST and came into the picture. The main idea behind designing SOAP was to ensure that programs built on different platforms and programming languages could exchange data in an easy manner.

**REST** – This was designed specifically for working with components such as media components, files, or even objects on a particular hardware device. Any web service that is defined on the principles of REST can be called a RestFul web service. A Restful service would use the normal HTTP verbs of GET, POST, PUT and DELETE for working with the required components.



* **Explain all below mentioned HTTP status code in your words with examples?**
  + **HTTP STATUS CODE: 200, 300, 301, 302, 304, 307, 400, 401, 403, 404, 500, 503, 550**

### **200 OK**

The request has succeeded. The information returned with the response is dependent on the method used in the request

### **300 Multiple Choices**

The requested resource corresponds to any one of a set of representations, each with its own specific location, and agent- driven negotiation information is being provided so that the user (or user agent) can select a preferred representation and redirect its request to that location.

### **301 Moved Permanently**

The requested resource has been assigned a new permanent URI and any future references to this resource SHOULD use one of the returned URIs. Clients with link editing capabilities ought to automatically re-link references to the Request-URI to one or more of the new references returned by the server, where possible. This response is cacheable unless indicated otherwise.

### **302 Found**

The requested resource resides temporarily under a different URI. Since the redirection might be altered on occasion, the client SHOULD continue to use the Request-URI for future requests. This response is only cacheable if indicated by a Cache-Control or Expires header field.

### **304 Not Modified**

If the client has performed a conditional GET request and access is allowed, but the document has not been modified, the server SHOULD respond with this status code. The 304 response MUST NOT contain a message-body, and thus is always terminated by the first empty line after the header fields.

### **307 Temporary Redirect**

The requested resource resides temporarily under a different URI. Since the redirection MAY be altered on occasion, the client SHOULD continue to use the Request-URI for future requests. This response is only cacheable if indicated by a Cache-Control or Expires header field.

### **400 Bad Request**

The request could not be understood by the server due to malformed syntax. The client SHOULD NOT repeat the request without modifications.

### **401 Unauthorized**

The request requires user authentication.

### **403 Forbidden**

The server understood the request, but is refusing to fulfill it. Authorization will not help and the request SHOULD NOT be repeated.

### **404 Not Found**

The server has not found anything matching the Request-URI.

### **500 Internal Server Error**

The server encountered an unexpected condition which prevented it from fulfilling the request.

### **503 Service Unavailable**

The server is currently unable to handle the request due to a temporary overloading or maintenance of the server

### **HTTP 550 Permission Denied**

The server is stating the account you have currently logged in as does not have permission to perform the action you are attempting. You may be trying to upload to the wrong directory or trying to delete a file.