https://developer.mozilla.org/en-US/docs/Learn/Server-side/Configuring_server_MIME_types

Background

MIME types describe the media type of content either in email or served by web servers or web applications and are intended to help guide a web browser in how the content is to be processed and displayed.

Examples of MIME types:

- `text/html` for HTML documents
- `text/plain` for plain text
- `text/css` for Cascading Style Sheets
- `text/javascript` for scripts
- `text/markdown` for Markdown files
- `application/octet-stream` for binary files where user action is expected

The default configurations of servers vary wildly and will provide different default values if there is no default content type defined.

Versions of the Apache Web Server before **before 2.2.7** were configured to report a MIME type of `text/plain` or `application/octet-stream` for unknown content types. The current version of Apache reports none for files with unknown content types.

Nginx will report `text/plain` if you don't define a default content type.

As new content types are invented or added to web servers, web administrators may fail to add the new MIME types to their web server's configuration. This is a major source of problems for users of browsers that respect the MIME types reported by web servers and applications.

Why are correct MIME types important?

According to the HTTP specification, If the Example of an incorrect MIME type result web server or application reports an incorrect MIME type for content, a web browser has no way, of knowing the author's intentions and will serve the content with default MIME types that will work in unexpected ways.

Some web browsers, such as Internet Explorer, tries to *guess* the correct

MIME type, allowing misconfigured web servers and applications to continue working for Intenet Explorer but not other browsers that support the standard.

Serving content using the correct MIME type can also be important for security reasons; it's possible for malicious content to affect the user's computer by pretending to be a safe type of document when it is in fact something else that can put your computer at risk.

Javascript Legacy MIME Types

When looking for information about Javascript MIME types, you may see several MIME types that reference Javascript. Some of these MIME Types include:

- application/javascript
- application/ecmascript
- application/x-ecmascript
- application/x-javascript
- text/ecmascript
- text/javascript1.0
- text/javascript1.1
- text/javascript1.2
- text/javascript1.3
- text/javascript1.4
- text/javascript1.5
- text/x-ecmascript
- text/x-javascript

While browsers may support any, some, or all of these alternative MIME types, you should only use `text/javascript` to indicate the mimetype of Javascript files.

See MIME types (IANA media types) for more information

Why browsers should not guess MIME types

Apart from violating the HTTP specification, it is a bad strategy for browsers to guess MIME types for the following reasons:

Loss of control

If the browser ignores the reported MIME type, web administrators and authors no longer have control over how their content is to be processed.

For example, a web site oriented for web developers might wish to send certain example HTML documents as either `text/html` or `text/plain` in order to have the documents either processed and displayed as HTML or as source code. If the browser guesses the MIME type, this option is no longer available to the author.

Security

Some content types, such as executable programs, are inherently unsafe. For this reason these MIME types are usually restricted in terms of what actions a web browser will take when given content of that type. An executable program should not be executed on the user's computer and at most should cause a dialog to appear **asking the user** if they wish to download the file.

MIME type guessing has led to security exploits in Internet Explorer which were based upon a malicious author incorrectly reporting a MIME type of a dangerous file as a safe type. This bypassed the normal download dialog resulting in Internet Explorer guessing that the content was an executable program and then running it on the user's computer.

How to determine the MIME type sent by a server

- In Firefox
  - Load the file and use **go to Tools | Page Info to get the content type for the page you accessed**
  - You can also go to **Tools | Web Developer | Network** and reload the page. The request tab gives you a list of all the resources the page loaded. Clicking on any resource will list all the information available, including **Content-Type**
- In Chrome
  - Load the file and go to **View | Developer | Developer Tools** and choose the network tab. Reload the page and select the resource you want to inspect. Under headers look for **Content-Type** and it will report the content type of the resource
- Look for a `meta` tag that gives the MIME type such as `<meta http-equiv="Content-Type" content="text/html">`

- According to the standards, the `meta` tag that gives the MIME type should be ignored if there's a {{HTTPHeader("Content-Type")}} line in the header

IANA keeps a list of registered MIME Media Types. The HTTP specification defines a superset of MIME which is used to describe the media types used on the web.

How to determine the correct MIME type for your content

There are several ways to determine the correct MIME type value to be used for your content.

- If your content was created using commerical software, read the vendor's documentation to see what MIME types should be reported for the application
- Look in IANA's MIME Media Types registry which contains all registered MIME types
- Search for the file extension in FILExt or File extensions reference to see what MIME types are associated with that extension. Pay close attention as the application may have multiple MIME types that are different in only one letter

How to set up your server to send the correct MIME types

The goal is to configure your server to send the correct {{HTTPHeader("Content-Type")}} HTTP header for each document.

- If you're using the Apache web server, simply copy this sample .htaccess file to the directory that contains the files that you want to work with or the parent directory if there are many such directories.
- If you're using NGINX, look at the NGINX configuration snipets. NGINX does not have a .htaccess equivalent tool, so all changes will go into the main configuration file
- If you're using a server-side script or framework to generate content, the way to indicate the content type will depend on the tool you're using. Check the framework or library's documentation

Related Links

- Incorrect MIME Type for CSS Files
- IANA | MIME Media Types
- Hypertext Transfer Protocol — HTTP/1.1
- MIME types (IANA media types)
- Apache vs Nginx: Practical Considerations
- Migrate Apache .htaccess to NGINX server block
- Microsoft - 293336 - INFO: WebCast: MIME Type Handling in Microsoft Internet Explorer

{{QuickLinksWithSubpages("/en-US/docs/Web/Security")}}