Django

STDC

May 3, 2025

Table of contents

Server-side website programming

- Django is an extremely popular and fully featured server-side web framework, written in Python.
- Web browsers communicate with web servers using the HyperText Transfer Protocol (HTTP).
- When click a link on a web page, submit a form, or run a search, an HTTP request is sent from your browser to the target server.

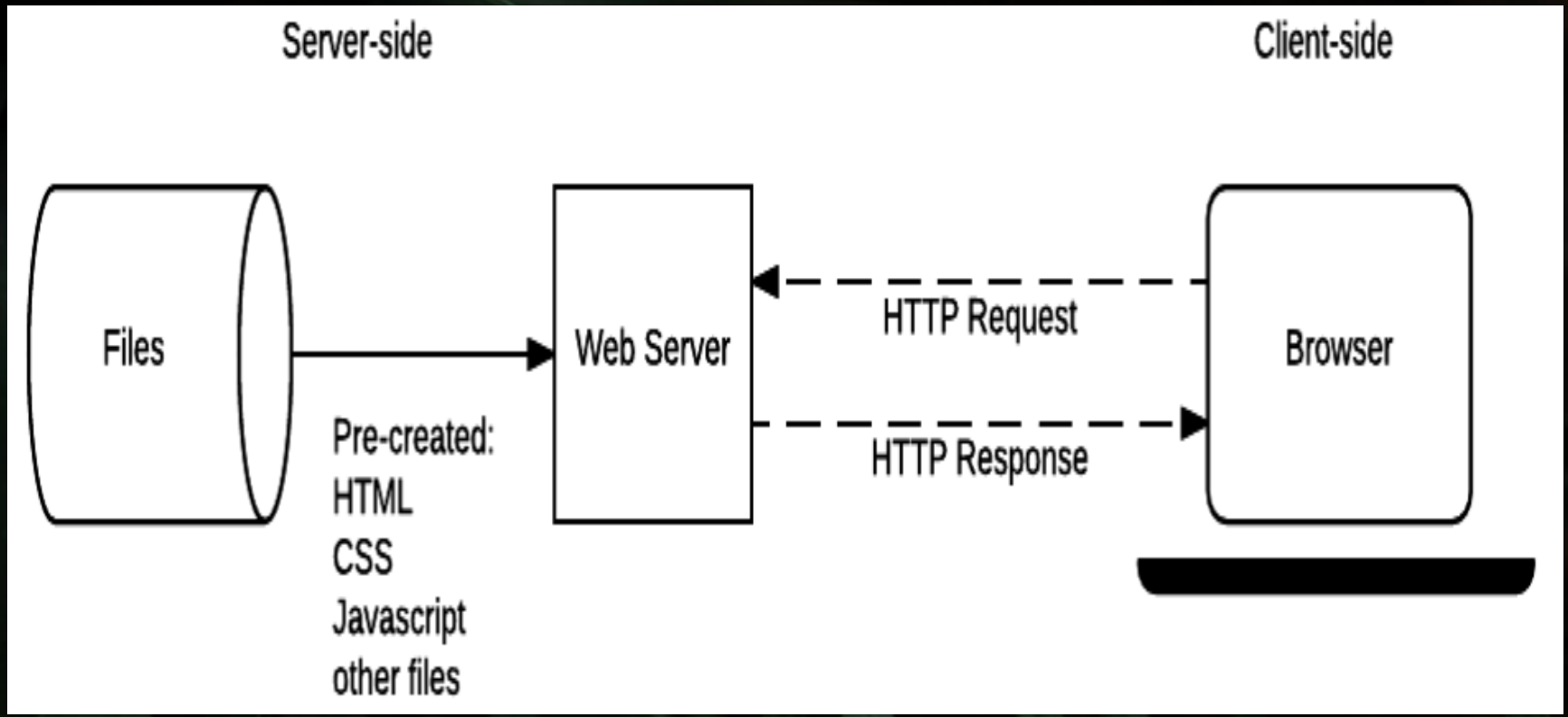
Server-side website programming

- The request includes a URL identifying the affected resource, a method that defines the required action (for example to get, delete, or post the resource), and may include additional information encoded in URL parameters (the field-value pairs sent via a query string), as POST data (data sent by the HTTP POST method), or in associated cookies.

Static sites

- When a user wants to navigate to a page, the browser sends an HTTP "GET" request specifying its URL.
- The server retrieves the requested document from its file system and returns an HTTP response containing the document and a success status (usually 200 OK).
- If the file cannot be retrieved for some reason, an error status is returned.

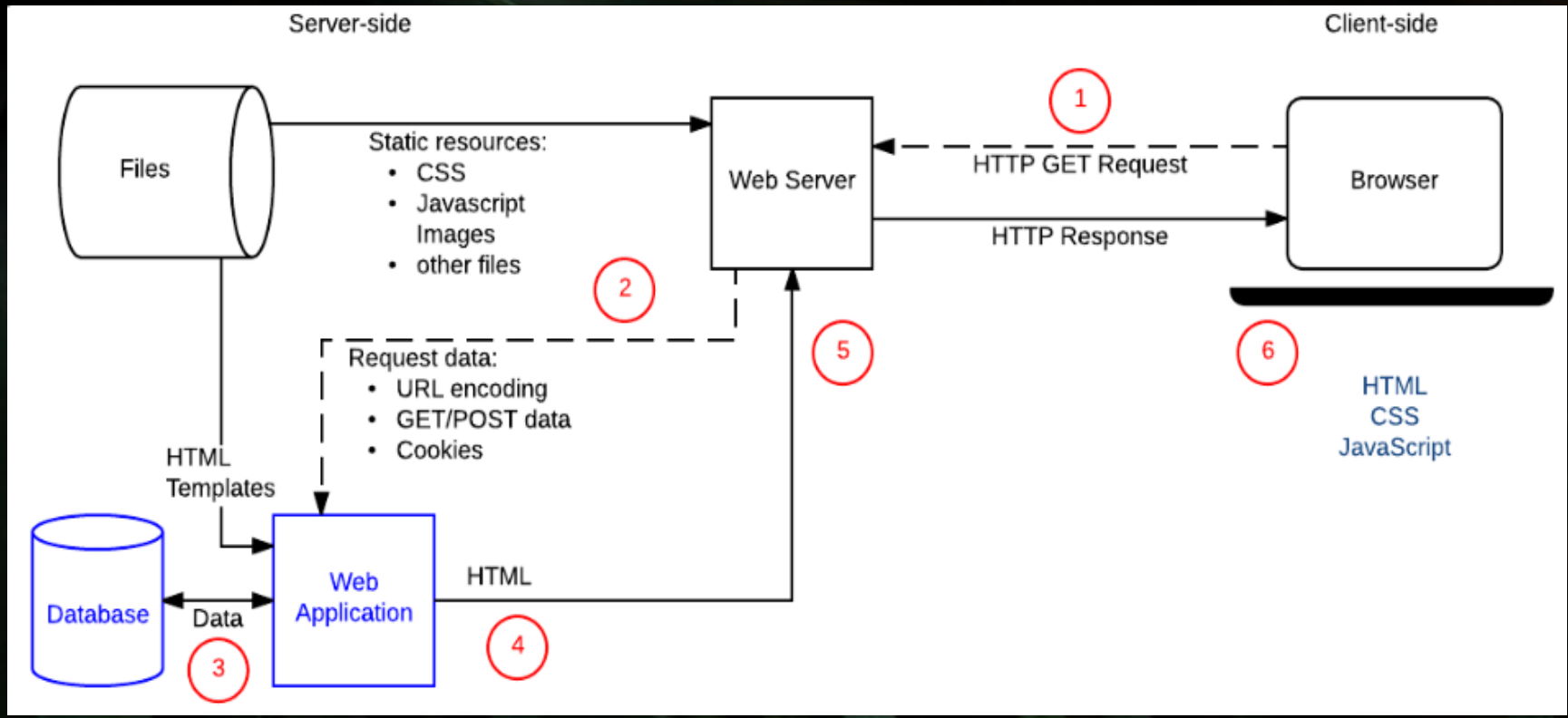
Static sites



Dynamic sites

- A dynamic website is one where some of the response content is generated dynamically, only when needed.
- On a dynamic website, HTML pages are normally created by inserting data from a database into placeholders in HTML templates.
- A dynamic site can return different data for a URL based on information provided by the user or stored preferences and can perform other operations as part of returning a response.
- Most of the code to support a dynamic website must run on the server. Creating this code is known as "server-side programming".

Dynamic sites



Dynamic sites

- Server-side programming allows sites to restrict access to authorized users and serve only the information that a user is permitted to see.
- Server-side programming allows developers to make use of sessions — a mechanism that allows a server to store information associated with the current user of a site and send different responses based on that information.

Web servers and HTTP

- Web browsers communicate with web servers using the HyperText Transfer Protocol (HTTP).
- When you click a link on a web page, submit a form, or run a search, the browser sends an HTTP Request to the server.

Web servers and HTTP

- The request includes:
  - When you click a link on a web page, submit a form, or run a search, the browser sends an HTTP Request to the server.
  - A method that defines the required action (for example, to get a file or to save or update some data).

Web servers and HTTP

- GET: Get a specific resource.
- POST: Create a new resource.
- HEAD: Get the metadata information about a specific resource without getting the body like GET would.
- PUT: Update an existing resource.
- DELETE: Delete the specified resource.

Web servers and HTTP

- The web browser creates an HTTP GET request to the server using the base URL for the resource and encoding the team and player number either as URL parameters or as part of the URL pattern.
- A GET request is used because the request is only fetching data (not modifying data).

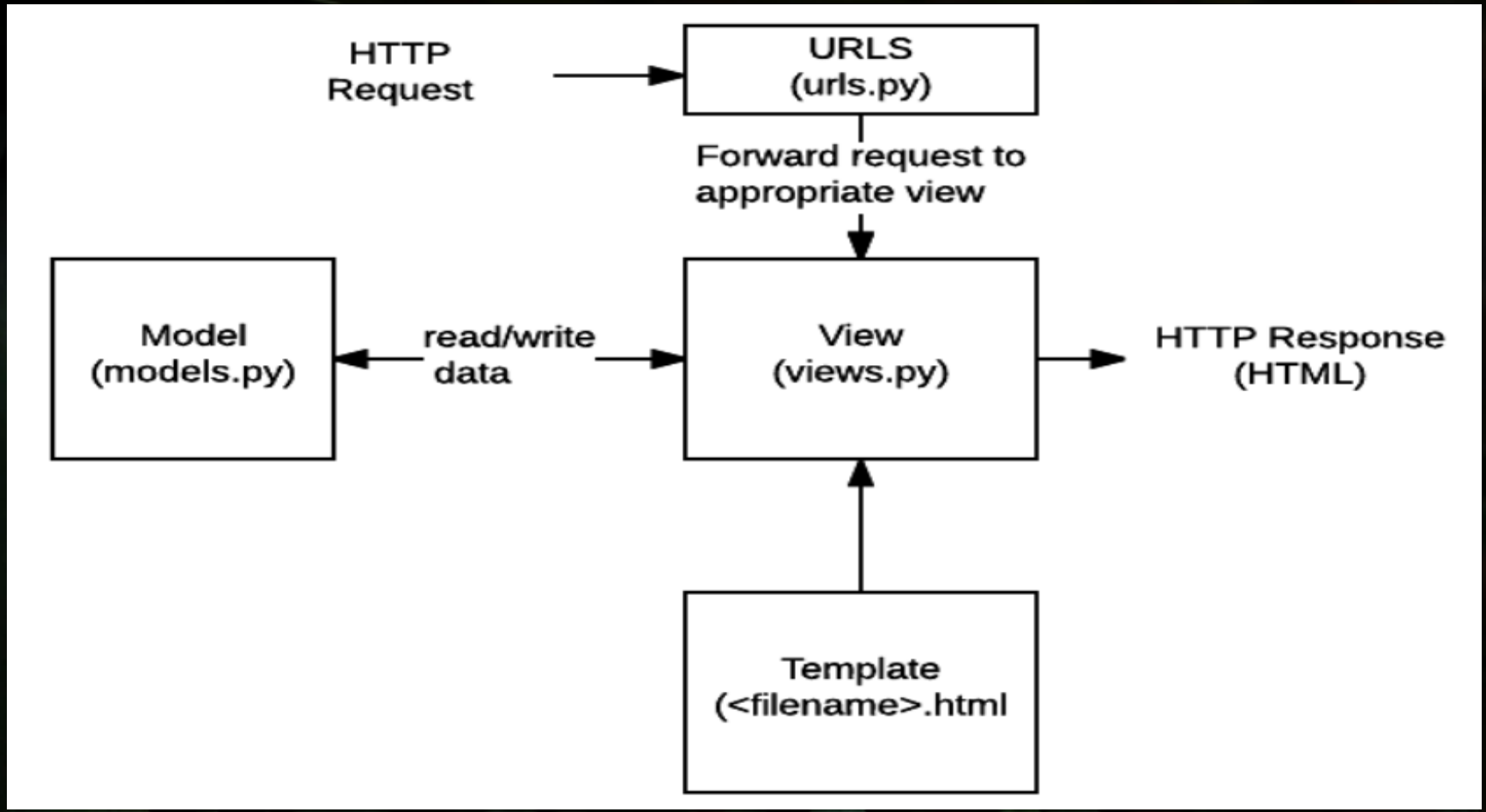
Web servers and HTTP

- The Web Application identifies that the intention of the request is to get the "best team list" based on the URL (/best/) and finds out the required team name and number of players from the URL.
- The Web Application then gets the required information from the database.

A web framework - Django

- Web frameworks provide tools and libraries to simplify common web development operations.
- Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design.
- Popular sites using Django (from Django home page) include: Disqus, Instagram, Knight Foundation, MacArthur Foundation, Mozilla, National Geographic, Open Knowledge Foundation, Pinterest, Open Stack.

A web framework - Django



Django-URLs

- While it is possible to process requests from every single URL via a single function, it is much more maintainable to write a separate view function to handle each resource.
- A URL mapper is used to redirect HTTP requests to the appropriate view based on the request URL.
- The URL mapper can also match particular patterns of strings or digits that appear in a URL and pass these to a view function as data.

Django-View

- A view is a request handler function, which receives HTTP requests and returns HTTP responses.
- Views access the data needed to satisfy requests via models, and delegate the formatting of the response to templates.

Django-Models

- Models are Python objects that define the structure of an application's data, and provide mechanisms to manage (add, modify, delete) and query records in the database.

Django-Templates

- A template is a text file defining the structure or layout of a file (such as an HTML page), with placeholders used to represent actual content.
- A view can dynamically create an HTML page using an HTML template, populating it with data from a model.
- A template can be used to define the structure of any type of file; it doesn't have to be HTML!

HTML

- HTML is the standard markup language for creating Web pages.
- HTML stands for Hyper Text Markup Language
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

HTML

- The < !DOCTYPE html> declaration defines that this document is an HTML5 document
- The <html> element is the root element of an HTML page
- The <head> element contains meta information about the HTML page
- The <title> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
- The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
- The <h1> element defines a large heading
- The <p> element defines a paragraph

Formatting

- : Creates a new paragraph
-
: AInserts a line break (carriage return)
- <blockquote> </blockquote>: Puts content in a quote - indents text from both sides
- : Used to format block content with CSS
- : Used to format inline content with CSS

Body attributes

- <body bgcolor=?>: Sets background color, using name or hex value
- <body text=?>: Sets text color, using name or hex value
- <body link=?>: Sets color of links, using name or hex value
- <body vlink=?>: Sets color of visited links, using name or hex value
- <body alink=?>: Sets color of active links (while mouse-clicking)

Text Tags

- <pre> </pre> Creates preformatted text
- <h1> </h1> –> <h6> </h6> Creates headlines – H1=largest, H6=smallest
-   Creates bold text (should use instead)
-   Creates italicized text (should use instead)
- <tt> </tt> Creates typewriter-style text
- <code> </code> Used to define source code, usually monospace
- <cite> </cite> Creates a citation, usually processed in italics
- <address> </address> Creates address section, usually processed in italics

Text Tags

-   Emphasizes a word (usually processed in italics)
-   Emphasizes a word (usually processed in bold)
-   Sets size of font - 1 to 7 (should use CSS instead)
-   Sets font color (should use CSS instead)
-   Defines the font used (should use CSS instead)

Lists

- : Creates an unordered list
- <ol start=?> : Creates an ordered list (start=xx, where xx is a counting number)
- : Encompasses each list item
- <dl> </dl>: Creates a definition list
- <dt>: Precedes each defintion term
- <dd>: Precedes each definition

Links

- clickable text: Creates a hyperlink to a Uniform Resource Locator
- clickable text: Creates a hyperlink to an email address
- <a name="NAME">: Creates a target location within a document
- clickable text: Creates a link to that target location

Graphical elements

- <hr>: Inserts a horizontal rule
- <hr size=?>: Sets size (height) of horizontal rule
- <hr width=?>: Sets width of rule (as a % or absolute pixel length)
- <hr noshade>: Creates a horizontal rule without a shadow
- : Adds image; it is a separate file located at the URL
- <img src="URL" align=?>: Aligns image left/right/center/bottom/top/middle (use CSS)

Graphical elements

- <img src="URL" border=?>: Sets size of border surrounding image (use CSS)
- <img src="URL" height=?>: Sets height of image, in pixels
- <img src="URL" width=?>: Sets width of image, in pixels
- <img src="URL" alt=?>: Sets the alternate text for browsers that can't process images

Forms

- <form> </form>: Defines a form
- <select multiple name=? size=?> </select>: Creates a scrolling menu. Size sets the number of menu items visible before user needs to scroll.
- <select name=?> </select>: Creates a pulldown menu
- <option>: Sets off each menu item
- <textarea name=? cols="x" rows="y"></textarea>: Creates a text box area. Columns set the width; rows set the height.

Forms

- <input type="checkbox" name=? value=?>: Creates a checkbox.
- <input type="checkbox" name=? value=? checked>: Creates a checkbox which is pre-checked.
- <input type="radio" name=? value=?>: Creates a radio button.
- <input type="radio" name=? value=? checked>: Creates a radio button which is pre-checked.
- <input type="text" name=? size=?>: Creates a one-line text area. Size sets length, in characters.

Forms

- <input type="submit" value=?>: Creates a submit button. Value sets the text in the submit button.
- <input type="image" name=? src=? border=? alt=?>: Creates a submit button using an image.
- <input type="reset">: Creates a reset button

Tables

- <table> </table>: Creates a table
- <tr> </tr>: Sets off each row in a table
- <td> </td>: Sets off each cell in a row
- <th> </th>: Sets off the table header (a normal cell with bold, centered text)

Table attributes

- <table border=?>: Sets the width of the border around table cells
- <table cellspacing=?>: Sets amount of space between table cells
- <table cellpadding=?>: Sets amount of space between a cell's border and its contents
- <table width=?>: Sets width of the table in pixels or as a percentage

Table attributes

- <tr align=?>: Sets alignment for cells within the row (left/center/right)
- <td align=?>: Sets alignment for cells (left/center/right)
- <tr valign=?>: Sets vertical alignment for cells within the row (top/middle/bottom)
- <td valign=?>: Sets vertical alignment for cell (top/middle/bottom)
- <td rowspan=?>: Sets number of rows a cell should span (default=1)
- <td colspan=?>: Sets number of columns a cell should span
- <td nowrap>: Prevents lines within a cell from being broken to fit

HTML5 input tag attributes

- <input type="email" name=?> : Sets a single-line textbox for email addresses
- <input type="url" name=?> : Sets a single-line textbox for URLs
- <input type="number" name=?>: Sets a single-line textbox for a number
- <input type="range" name=?> : Sets a single-line text box for a range of numbers
- <input type="date/month/week/time" name=?>: Sets a single-line text box with a calendar showing the date/month/week/time
- <input type="search" name=?> : Sets a single-line text box for searching
- <input type="color" name=?> : Sets a single-line text box for picking a color