Web framework

From Wikipedia, the free encyclopedia

  (Redirected from [Web application framework](https://en.wikipedia.org/w/index.php?title=Web_application_framework&redirect=no))

[Jump to navigation](https://en.wikipedia.org/wiki/Web_framework#mw-head)[Jump to search](https://en.wikipedia.org/wiki/Web_framework#searchInput)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | hide**This article has multiple issues.** Please help [**improve it**](https://en.wikipedia.org/w/index.php?title=Web_framework&action=edit) or discuss these issues on the [**talk page**](https://en.wikipedia.org/wiki/Talk:Web_framework). *(*[*Learn how and when to remove these template messages*](https://en.wikipedia.org/wiki/Help:Maintenance_template_removal)*)*   |  | | --- | | This article **needs additional citations for**[**verification**](https://en.wikipedia.org/wiki/Wikipedia:Verifiability). *(August 2008)* |  |  | | --- | | This article **possibly contains**[**original research**](https://en.wikipedia.org/wiki/Wikipedia:No_original_research). *(December 2010)* |  |  | | --- | | This article **needs attention from an expert on the subject**. *(January 2017)* | |

A **web framework** (**WF**) or **web application framework** (**WAF**) is a software framework that is designed to support the development of [web applications](https://en.wikipedia.org/wiki/Web_application) including web services, [web resources](https://en.wikipedia.org/wiki/Web_resource), and web APIs. Web frameworks provide a standard way to build and deploy web applications on the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web). Web frameworks aim to [automate](https://en.wikipedia.org/wiki/Automation) the overhead associated with common activities performed in [web development](https://en.wikipedia.org/wiki/Web_development). For example, many web frameworks provide [libraries](https://en.wikipedia.org/wiki/Library_(computing)) for [database](https://en.wikipedia.org/wiki/Database) access, [templating](https://en.wikipedia.org/wiki/Template_processor) frameworks, and [session](https://en.wikipedia.org/wiki/Session_(computer_science)) management, and they often promote [code reuse](https://en.wikipedia.org/wiki/Code_reuse).[[1]](https://en.wikipedia.org/wiki/Web_framework#cite_note-1) Although they often target development of [dynamic web sites](https://en.wikipedia.org/wiki/Dynamic_web_page), they are also applicable to [static websites](https://en.wikipedia.org/wiki/Static_web_page).[[2]](https://en.wikipedia.org/wiki/Web_framework#cite_note-2)



**Contents**

* [1History](https://en.wikipedia.org/wiki/Web_framework#History)
* [2Types of framework architectures](https://en.wikipedia.org/wiki/Web_framework#Types_of_framework_architectures)
  + [2.1Model–view–controller (MVC)](https://en.wikipedia.org/wiki/Web_framework#Model%E2%80%93view%E2%80%93controller_(MVC))
    - [2.1.1Push-based vs. pull-based](https://en.wikipedia.org/wiki/Web_framework#Push-based_vs._pull-based)
  + [2.2Three-tier organization](https://en.wikipedia.org/wiki/Web_framework#Three-tier_organization)
* [3Framework applications](https://en.wikipedia.org/wiki/Web_framework#Framework_applications)
  + [3.1General-purpose website frameworks](https://en.wikipedia.org/wiki/Web_framework#General-purpose_website_frameworks)
    - [3.1.1Server-side](https://en.wikipedia.org/wiki/Web_framework#Server-side)
    - [3.1.2Client-side](https://en.wikipedia.org/wiki/Web_framework#Client-side)
  + [3.2Discussion forums, wikis and weblogs](https://en.wikipedia.org/wiki/Web_framework#Discussion_forums,_wikis_and_weblogs)
* [4Features](https://en.wikipedia.org/wiki/Web_framework#Features)
  + [4.1Web template system](https://en.wikipedia.org/wiki/Web_framework#Web_template_system)
  + [4.2Caching](https://en.wikipedia.org/wiki/Web_framework#Caching)
  + [4.3Security](https://en.wikipedia.org/wiki/Web_framework#Security)
  + [4.4Database access, mapping and configuration](https://en.wikipedia.org/wiki/Web_framework#Database_access,_mapping_and_configuration)
  + [4.5URL mapping](https://en.wikipedia.org/wiki/Web_framework#URL_mapping)
  + [4.6AJAX](https://en.wikipedia.org/wiki/Web_framework#AJAX)
  + [4.7Web services](https://en.wikipedia.org/wiki/Web_framework#Web_services)
  + [4.8Web resources](https://en.wikipedia.org/wiki/Web_framework#Web_resources)
* [5See also](https://en.wikipedia.org/wiki/Web_framework#See_also)
* [6References](https://en.wikipedia.org/wiki/Web_framework#References)
* [7Bibliography](https://en.wikipedia.org/wiki/Web_framework#Bibliography)

History[[edit](https://en.wikipedia.org/w/index.php?title=Web_framework&action=edit&section=1)]

|  |  |
| --- | --- |
|  | This section **is missing information about the date some of the ideas arose**. Please expand the section to include this information. Further details may exist on the [talk page](https://en.wikipedia.org/wiki/Talk:Web_framework). *(June 2013)* |

*Further information:*[*Rich web application*](https://en.wikipedia.org/wiki/Rich_web_application)

As the design of the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web) was not inherently dynamic, early [hypertext](https://en.wikipedia.org/wiki/Hypertext) consisted of hand-coded [HTML](https://en.wikipedia.org/wiki/HTML) that was published on [web servers](https://en.wikipedia.org/wiki/Web_servers). Any modifications to published pages needed to be performed by the pages' author. In 1993, the [Common Gateway Interface](https://en.wikipedia.org/wiki/Common_Gateway_Interface) (CGI) standard was introduced for interfacing external applications with web servers, to provide a [dynamic web page](https://en.wikipedia.org/wiki/Dynamic_web_page) that reflected user inputs.[[3]](https://en.wikipedia.org/wiki/Web_framework#cite_note-3)

Original implementations of the CGI interface typically had adverse effects on the server load however, because each request started a separate [process](https://en.wikipedia.org/wiki/Process_(computing)).[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)] More recent implementations utilize persistent processes amongst other techniques to reduce the footprint in the server's resources and offer a general performance boost.

In 1995, fully integrated server/language development environments first emerged and new web-specific languages were introduced, such as [ColdFusion](https://en.wikipedia.org/wiki/ColdFusion), [PHP](https://en.wikipedia.org/wiki/PHP), and [Active Server Pages](https://en.wikipedia.org/wiki/Active_Server_Pages).[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

Although the vast majority of languages for creating dynamic web pages have [libraries](https://en.wikipedia.org/wiki/Library_(computing)) to help with common tasks, [web applications](https://en.wikipedia.org/wiki/Web_applications) often require specific libraries for particular tasks, such as creating [HTML](https://en.wikipedia.org/wiki/HTML) (for example, [JavaServer Faces](https://en.wikipedia.org/wiki/JavaServer_Faces)).[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

In the late 1990s, mature, "full stack" frameworks began to appear, that often gathered multiple libraries useful for [web development](https://en.wikipedia.org/wiki/Web_development) into a single cohesive [software stack](https://en.wikipedia.org/wiki/Software_stack) for web developers to use. Examples of this include [ASP.NET](https://en.wikipedia.org/wiki/ASP.NET), [Java EE](https://en.wikipedia.org/wiki/Java_Platform,_Enterprise_Edition), [WebObjects](https://en.wikipedia.org/wiki/WebObjects), [web2py](https://en.wikipedia.org/wiki/Web2py), [OpenACS](https://en.wikipedia.org/wiki/OpenACS), [Catalyst](https://en.wikipedia.org/wiki/Catalyst_(software)), [Mojolicious](https://en.wikipedia.org/wiki/Mojolicious), [Ruby on Rails](https://en.wikipedia.org/wiki/Ruby_on_Rails), [Laravel](https://en.wikipedia.org/wiki/Laravel), [Grails](https://en.wikipedia.org/wiki/Grails_(Framework)), [Django](https://en.wikipedia.org/wiki/Django_(web_framework)), [Zend Framework](https://en.wikipedia.org/wiki/Zend_Framework), [Sails.js](https://en.wikipedia.org/wiki/Sails.js), [Yii](https://en.wikipedia.org/wiki/Yii),[[4]](https://en.wikipedia.org/wiki/Web_framework#cite_note-4) [CakePHP](https://en.wikipedia.org/wiki/CakePHP),[[5]](https://en.wikipedia.org/wiki/Web_framework#cite_note-5) and [Symfony](https://en.wikipedia.org/wiki/Symfony).[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

Types of framework architectures[[edit](https://en.wikipedia.org/w/index.php?title=Web_framework&action=edit&section=2)]

Most web frameworks are based on the [model–view–controller](https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller) (MVC) [pattern](https://en.wikipedia.org/wiki/Software_design_pattern).[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

**Model–view–controller (MVC)**[[edit](https://en.wikipedia.org/w/index.php?title=Web_framework&action=edit&section=3)]

*Main article:*[*Model–view–controller*](https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller)

Many frameworks follow the MVC [architectural pattern](https://en.wikipedia.org/wiki/Architectural_pattern_(computer_science)) to separate the [data model](https://en.wikipedia.org/wiki/Data_model) with [business rules](https://en.wikipedia.org/wiki/Business_rule) from the [user interface](https://en.wikipedia.org/wiki/User_interface). This is generally considered a good practice as it modularizes code, promotes [code reuse](https://en.wikipedia.org/wiki/Code_reuse), and allows multiple interfaces to be applied. In web applications, this permits different views to be presented, such as [web pages](https://en.wikipedia.org/wiki/Web_page) for humans, and [web service](https://en.wikipedia.org/wiki/Web_service) interfaces for remote applications.[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

**Push-based vs. pull-based**[[edit](https://en.wikipedia.org/w/index.php?title=Web_framework&action=edit&section=4)]

Most MVC frameworks follow a push-based architecture also called "action-based". These frameworks use actions that do the required processing, and then "push" the data to the view layer to render the results.[[6]](https://en.wikipedia.org/wiki/Web_framework#cite_note-6) [Django](https://en.wikipedia.org/wiki/Django_(web_framework)), [Ruby on Rails](https://en.wikipedia.org/wiki/Ruby_on_Rails), [Symfony](https://en.wikipedia.org/wiki/Symfony), [Spring MVC](https://en.wikipedia.org/wiki/Spring_MVC), [Stripes](https://en.wikipedia.org/wiki/Stripes_(framework)), [Sails.js](https://en.wikipedia.org/wiki/Sails.js), [Diamond](https://en.wikipedia.org/w/index.php?title=Diamond_MVC&action=edit&redlink=1), [CodeIgniter](https://en.wikipedia.org/wiki/CodeIgniter)[[7]](https://en.wikipedia.org/wiki/Web_framework#cite_note-7) are good examples of this architecture. An alternative to this is pull-based architecture, sometimes also called "component-based". These frameworks start with the view layer, which can then "pull" results from multiple controllers as needed. In this architecture, multiple controllers can be involved with a single view. [Lift](https://en.wikipedia.org/wiki/Lift_(web_framework)), [Tapestry](https://en.wikipedia.org/wiki/Tapestry_(programming)), [JBoss Seam](https://en.wikipedia.org/wiki/JBoss_Seam), [JavaServer Faces](https://en.wikipedia.org/wiki/JavaServer_Faces), [(µ)Micro](http://micro-docs.simplegames.ca/), and [Wicket](https://en.wikipedia.org/wiki/Apache_Wicket) are examples of pull-based architectures. [Play](https://en.wikipedia.org/wiki/Play_Framework), [Struts](https://en.wikipedia.org/wiki/Apache_Struts), [RIFE](https://en.wikipedia.org/wiki/RIFE), and [ZK](https://en.wikipedia.org/wiki/ZK_(framework)) have support for both push- and pull-based application controller calls.[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

**Three-tier organization**[[edit](https://en.wikipedia.org/w/index.php?title=Web_framework&action=edit&section=5)]

In [three-tier organization](https://en.wikipedia.org/wiki/Three-tier_architecture), applications are structured around three physical tiers: client, application, and database.[[8]](https://en.wikipedia.org/wiki/Web_framework#cite_note-8)[[9]](https://en.wikipedia.org/wiki/Web_framework#cite_note-9)[[10]](https://en.wikipedia.org/wiki/Web_framework#cite_note-10)[[11]](https://en.wikipedia.org/wiki/Web_framework#cite_note-11) The database is normally an [RDBMS](https://en.wikipedia.org/wiki/Relational_database_management_system). The application contains the business logic, running on a server and communicates with the client using [HTTP](https://en.wikipedia.org/wiki/HTTP).[[12]](https://en.wikipedia.org/wiki/Web_framework#cite_note-12) The client on web applications is a web browser that runs HTML generated by the application layer.[[13]](https://en.wikipedia.org/wiki/Web_framework#cite_note-13)[[14]](https://en.wikipedia.org/wiki/Web_framework#cite_note-14) The term should not be confused with MVC, where, unlike in three-tier architecture, it is considered a good practice to keep business logic away from the controller, the "middle layer".[[15]](https://en.wikipedia.org/wiki/Web_framework#cite_note-15)[[16]](https://en.wikipedia.org/wiki/Web_framework#cite_note-16)

Framework applications[[edit](https://en.wikipedia.org/w/index.php?title=Web_framework&action=edit&section=6)]

Frameworks are built to support the construction of internet applications based on a single programming language, ranging in focus from general purpose tools such as Zend Framework and Ruby on Rails, which augment the capabilities of a specific language, to native-language programmable packages built around a specific user application, such as [content management systems](https://en.wikipedia.org/wiki/Content_management_system), some mobile development tools and some portal tools. [[17]](https://en.wikipedia.org/wiki/Web_framework#cite_note-17)

**General-purpose website frameworks**[[edit](https://en.wikipedia.org/w/index.php?title=Web_framework&action=edit&section=7)]

Web frameworks must function according to the architectural rules of browsers and [web protocols](https://en.wikipedia.org/wiki/Web_protocol) such as [HTTP](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol), which is [stateless](https://en.wikipedia.org/wiki/Stateless_protocol). Webpages are served up by a [server](https://en.wikipedia.org/wiki/Server_(computing)) and can then be modified by the browser using [JavaScript](https://en.wikipedia.org/wiki/JavaScript). Either approach has its advantages and disadvantages.[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

Server-side page changes typically require that the page be refreshed, but allow any language to be used and more computing power to be utilized. Client-side changes allow the page to be updated in small chunks which feels like a desktop application, but are limited to JavaScript and run in the user's browser, which may have limited computing power. Some mix of the two is typically used.[[18]](https://en.wikipedia.org/wiki/Web_framework#cite_note-18) Applications which make heavy use of JavaScript are called [single-page applications](https://en.wikipedia.org/wiki/Single-page_application) and typically make use of a client-side JavaScript web framework to organize the code.[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

**Server-side**[[edit](https://en.wikipedia.org/w/index.php?title=Web_framework&action=edit&section=8)]

* [Apache Wicket](https://en.wikipedia.org/wiki/Apache_Wicket)
* [ASP.NET Core](https://en.wikipedia.org/wiki/ASP.NET_Core)
* [CakePHP](https://en.wikipedia.org/wiki/CakePHP)
* [Catalyst](https://en.wikipedia.org/wiki/Catalyst_(software))
* [CodeIgniter](https://en.wikipedia.org/wiki/CodeIgniter)
* [CppCMS](https://en.wikipedia.org/wiki/CppCMS)
* [Django](https://en.wikipedia.org/wiki/Django_(web_framework))
* [Express.js](https://en.wikipedia.org/wiki/Express.js)
* [The Gravity Framework](https://www.thegravityframework.com/)
* [Gridfyx PHP](https://github.com/JacksonMangallay/gridfyx)
* [Yii](https://en.wikipedia.org/wiki/Yii)
* [Laravel](https://en.wikipedia.org/wiki/Laravel)
* [Mojolicious](https://en.wikipedia.org/wiki/Mojolicious)
* [Ruby on Rails](https://en.wikipedia.org/wiki/Ruby_on_Rails)
* [Sails.js](https://en.wikipedia.org/wiki/Sails.js)
* [Symfony](https://en.wikipedia.org/wiki/Symfony)
* [Spring MVC](https://en.wikipedia.org/wiki/Spring_MVC)
* [Wt (web toolkit)](https://en.wikipedia.org/wiki/Wt_(web_toolkit))
* [Zend Framework](https://en.wikipedia.org/wiki/Zend_Framework)[[19]](https://en.wikipedia.org/wiki/Web_framework#cite_note-19)
* [QCObjects](https://en.wikipedia.org/w/index.php?title=QCObjects&action=edit&redlink=1)
* [Phoenix](https://www.phoenixframework.org/)

**Client-side**[[edit](https://en.wikipedia.org/w/index.php?title=Web_framework&action=edit&section=9)]

*Main article:*[*Single-page application*](https://en.wikipedia.org/wiki/Single-page_application)

Examples include [Backbone.js](https://en.wikipedia.org/wiki/Backbone.js), [AngularJS](https://en.wikipedia.org/wiki/AngularJS), [Angular](https://en.wikipedia.org/wiki/Angular_(application_platform)), [QCObjects](https://en.wikipedia.org/w/index.php?title=QCObjects&action=edit&redlink=1), [EmberJS](https://en.wikipedia.org/wiki/EmberJS), [ReactJS](https://en.wikipedia.org/wiki/React_(JavaScript_library)) and [Vue.js](https://en.wikipedia.org/wiki/Vue.js).[[20]](https://en.wikipedia.org/wiki/Web_framework#cite_note-20)

**Discussion forums, wikis and weblogs**[[edit](https://en.wikipedia.org/w/index.php?title=Web_framework&action=edit&section=10)]

* WikiBase/[WikiWikiWeb](https://en.wikipedia.org/wiki/WikiWikiWeb)[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

Features[[edit](https://en.wikipedia.org/w/index.php?title=Web_framework&action=edit&section=11)]

|  |  |
| --- | --- |
|  | This section **does not**[**cite**](https://en.wikipedia.org/wiki/Wikipedia:Citing_sources)**any**[**sources**](https://en.wikipedia.org/wiki/Wikipedia:Verifiability). Please help [improve this section](https://en.wikipedia.org/w/index.php?title=Web_framework&action=edit) by [adding citations to reliable sources](https://en.wikipedia.org/wiki/Help:Referencing_for_beginners). Unsourced material may be challenged and [removed](https://en.wikipedia.org/wiki/Wikipedia:Verifiability#Burden_of_evidence). *(February 2013) (*[*Learn how and when to remove this template message*](https://en.wikipedia.org/wiki/Help:Maintenance_template_removal)*)* |

Frameworks typically set the [control flow](https://en.wikipedia.org/wiki/Control_flow) of a program and allow the user of the framework to "hook into" that flow by exposing various events.[[21]](https://en.wikipedia.org/wiki/Web_framework#cite_note-:0-21) This "[inversion of control](https://en.wikipedia.org/wiki/Inversion_of_control)" design pattern is considered to be a defining principle of a framework, and benefits the code by enforcing a common flow for a team which everyone can customize in similar ways.[[21]](https://en.wikipedia.org/wiki/Web_framework#cite_note-:0-21) For example, some popular "microframeworks" such as Ruby's [Sinatra](https://en.wikipedia.org/wiki/Sinatra_(software)) (which inspired [Express.js](https://en.wikipedia.org/wiki/Express.js)) allow for "middleware" hooks prior to and after HTTP requests. These middleware functions can be anything, and allow the user to define logging, authentication and session management, and redirecting.[[22]](https://en.wikipedia.org/wiki/Web_framework#cite_note-22)

**Web template system**[[edit](https://en.wikipedia.org/w/index.php?title=Web_framework&action=edit&section=12)]

*Main article:*[*Web template system*](https://en.wikipedia.org/wiki/Web_template_system)

**Caching**[[edit](https://en.wikipedia.org/w/index.php?title=Web_framework&action=edit&section=13)]

*Main article:*[*Web cache*](https://en.wikipedia.org/wiki/Web_cache)

Web caching is the [caching](https://en.wikipedia.org/wiki/Web_cache) of [web](https://en.wikipedia.org/wiki/World_wide_web) [documents](https://en.wikipedia.org/wiki/Electronic_document) in order to reduce [bandwidth](https://en.wikipedia.org/wiki/Bandwidth_(computing)) usage, [server](https://en.wikipedia.org/wiki/Web_server) [load](https://en.wikipedia.org/wiki/Load_(computing)), and perceived "[lag](https://en.wikipedia.org/wiki/Lag)". A web cache stores copies of documents passing through it; subsequent requests may be satisfied from the cache if certain conditions are met. Some application frameworks provide mechanisms for caching documents and bypassing various stages of the page's preparation, such as database access or template interpretation.[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

**Security**[[edit](https://en.wikipedia.org/w/index.php?title=Web_framework&action=edit&section=14)]

*Main article:*[*Web application security*](https://en.wikipedia.org/wiki/Web_application_security)

Some web frameworks come with [authentication](https://en.wikipedia.org/wiki/Authentication) and [authorization](https://en.wikipedia.org/wiki/Authorization) frameworks, that enable the [web server](https://en.wikipedia.org/wiki/Web_server) to identify the users of the application, and restrict access to functions based on some defined criteria. Drupal is one example that provides role-based access to pages, and provides a web-based interface for creating users and assigning them roles.[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

**Database access, mapping and configuration**[[edit](https://en.wikipedia.org/w/index.php?title=Web_framework&action=edit&section=15)]

Many web frameworks create a unified [API](https://en.wikipedia.org/wiki/API) to a database backend, enabling web applications to work with a variety of databases with no code changes, and allowing programmers to work with higher-level concepts. Additionally, some [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) frameworks contain mapping tools to provide [object-relational mapping](https://en.wikipedia.org/wiki/Object-relational_mapping), which maps [objects](https://en.wikipedia.org/wiki/Object-oriented_programming) to [tuples](https://en.wikipedia.org/wiki/Tuple).[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

Some frameworks minimize web application configuration through the use of [introspection](https://en.wikipedia.org/wiki/Introspection_(computer_science)) and/or following well-known conventions. For example, many Java frameworks use [Hibernate](https://en.wikipedia.org/wiki/Hibernate_(Java)) as a persistence layer, which can generate a database schema at runtime capable of persisting the necessary information. This allows the application designer to design business objects without needing to explicitly define a database schema. Frameworks such as [Ruby on Rails](https://en.wikipedia.org/wiki/Ruby_on_Rails) can also work in reverse, that is, define properties of model objects at runtime based on a database schema.[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

Other features web frameworks may provide include [transactional support](https://en.wikipedia.org/wiki/Database_transaction) and [database migration tools](https://en.wikipedia.org/wiki/Data_migration).[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

**URL mapping**[[edit](https://en.wikipedia.org/w/index.php?title=Web_framework&action=edit&section=16)]

*See also:*[*Front controller*](https://en.wikipedia.org/wiki/Front_controller)*and*[*URL redirection*](https://en.wikipedia.org/wiki/URL_redirection)

A framework's [URL](https://en.wikipedia.org/wiki/Uniform_Resource_Locator) mapping or routing facility is the mechanism by which the framework interprets URLs. Some frameworks, such as Drupal and Django, match the provided URL against pre-determined patterns using [regular expressions](https://en.wikipedia.org/wiki/Regular_expression), while some others use [rewriting](https://en.wikipedia.org/wiki/Rewriting) techniques to translate the provided URL into one that the underlying engine will recognize. Another technique is that of [graph traversal](https://en.wikipedia.org/wiki/Graph_traversal) such as used by [Zope](https://en.wikipedia.org/wiki/Zope), where a URL is decomposed in steps that traverse an object graph (of models and views).[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

A URL mapping system that uses pattern matching or rewriting to route and handle requests allows for [shorter](https://en.wikipedia.org/wiki/URL_shortening) more "[friendly URLs](https://en.wikipedia.org/wiki/Friendly_URL)" to be used, increasing the simplicity of the site and allowing for better indexing by search engines. For example, a URL that ends with "/page.cgi?cat=science&topic=physics" could be changed to simply "/page/science/physics". This makes the URL easier for people to remember, read and write, and provides search engines with better information about the structural layout of the site. A graph traversal approach also tends to result in the creation of friendly URLs. A shorter URL such as "/page/science" tends to exist by default as that is simply a shorter form of the longer traversal to "/page/science/physics".[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

**AJAX**[[edit](https://en.wikipedia.org/w/index.php?title=Web_framework&action=edit&section=17)]

*Main article:*[*Ajax framework*](https://en.wikipedia.org/wiki/Ajax_framework)

[Ajax](https://en.wikipedia.org/wiki/Ajax_(programming)), shorthand for "[*Asynchronous*](https://en.wikipedia.org/wiki/Asynchronous_I/O)[*JavaScript*](https://en.wikipedia.org/wiki/JavaScript)*and*[*XML*](https://en.wikipedia.org/wiki/XML)", is a web development technique for creating web applications. The intent is to make web pages feel more responsive by exchanging small amounts of data with the server behind the scenes, so that the entire web page does not have to be reloaded each time the user requests a change. This is intended to increase a web page's interactivity, speed, and [usability](https://en.wikipedia.org/wiki/Usability).[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

Due to the complexity of Ajax programming in JavaScript, there are numerous [Ajax frameworks](https://en.wikipedia.org/wiki/Ajax_framework) that exclusively deal with Ajax support. Some Ajax frameworks are even embedded as a part of larger frameworks. For example, the [jQuery](https://en.wikipedia.org/wiki/JQuery) [JavaScript library](https://en.wikipedia.org/wiki/JavaScript_library) is included in Ruby on Rails.[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

With the increased interest in developing "[Web 2.0](https://en.wikipedia.org/wiki/Web_2.0)" [rich media](https://en.wikipedia.org/wiki/Rich_media) applications, the complexity of programming directly in Ajax and JavaScript has become so apparent that compiler technology has stepped in, to allow developers to code in high-level languages such as Java, Python and Ruby. The first of these compilers was [Morfik](https://en.wikipedia.org/wiki/Morfik) followed by [Google Web Toolkit](https://en.wikipedia.org/wiki/Google_Web_Toolkit), with ports to Python and Ruby in the form of [Pyjs](https://en.wikipedia.org/wiki/Pyjs) and RubyJS following some time after. These compilers and their associated widget set libraries make the development of rich media Ajax applications much more akin to that of developing desktop applications.[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

**Web services**[[edit](https://en.wikipedia.org/w/index.php?title=Web_framework&action=edit&section=18)]

*Main article:*[*Web service*](https://en.wikipedia.org/wiki/Web_service)

Some frameworks provide tools for creating and providing web services. These utilities may offer similar tools as the rest of the web application.[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

**Web resources**[[edit](https://en.wikipedia.org/w/index.php?title=Web_framework&action=edit&section=19)]

*Main article:*[*Web resource*](https://en.wikipedia.org/wiki/Web_resource)

A number of newer [Web 2.0](https://en.wikipedia.org/wiki/Web_2.0) [RESTful](https://en.wikipedia.org/wiki/Representational_state_transfer) frameworks are now providing [resource-oriented architecture](https://en.wikipedia.org/wiki/Resource-oriented_architecture) (ROA) infrastructure for building collections of resources in a sort of [Semantic Web](https://en.wikipedia.org/wiki/Semantic_Web) [ontology](https://en.wikipedia.org/wiki/Ontology_engineering), based on concepts from [Resource Description Framework](https://en.wikipedia.org/wiki/Resource_Description_Framework) (RDF).[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)]

See also[[edit](https://en.wikipedia.org/w/index.php?title=Web_framework&action=edit&section=20)]

* [Comparison of web frameworks](https://en.wikipedia.org/wiki/Comparison_of_web_frameworks)
* [Rich Internet application](https://en.wikipedia.org/wiki/Rich_Internet_application)
* [List of rich Internet application frameworks](https://en.wikipedia.org/wiki/List_of_rich_Internet_application_frameworks)
* [List of web service frameworks](https://en.wikipedia.org/wiki/List_of_web_service_frameworks)
* [Application server](https://en.wikipedia.org/wiki/Application_server)
* [Comparison of application servers](https://en.wikipedia.org/wiki/Comparison_of_application_servers)
* [Application framework](https://en.wikipedia.org/wiki/Application_framework)
* [Application security](https://en.wikipedia.org/wiki/Application_security)
* [Convention over configuration](https://en.wikipedia.org/wiki/Convention_over_configuration)
* [Don't repeat yourself](https://en.wikipedia.org/wiki/Don%27t_repeat_yourself) (DRY)
* [Solution stack](https://en.wikipedia.org/wiki/Solution_stack)
* [Multiple phone web-based application framework](https://en.wikipedia.org/wiki/Multiple_phone_web-based_application_framework)

References[[edit](https://en.wikipedia.org/w/index.php?title=Web_framework&action=edit&section=21)]

* 1. [**^**](https://en.wikipedia.org/wiki/Web_framework#cite_ref-1) *Multiple (wiki).*[*"Web application framework"*](https://web.archive.org/web/20150723163302/http:/docforge.com/wiki/Web_application_framework)*. Docforge. Archived from*[*the original*](http://docforge.com/wiki/Web_application_framework)*on 2015-07-23.*
  2. [**^**](https://en.wikipedia.org/wiki/Web_framework#cite_ref-2) [*"Top Open-Source Static Site Generators"*](https://www.staticgen.com/)*. StaticGen.*
  3. [**^**](https://en.wikipedia.org/wiki/Web_framework#cite_ref-3) [*"CGI: Common Gateway Interface"*](https://web.archive.org/web/20090409213905/http:/hoohoo.ncsa.uiuc.edu/cgi/intro.html)*. Archived from*[*the original*](http://hoohoo.ncsa.uiuc.edu/cgi/intro.html)*on 2009-04-09.*
  4. [**^**](https://en.wikipedia.org/wiki/Web_framework#cite_ref-4) [*"Yii PHP Framework"*](http://www.yiiframework.com/)*.*
  5. [**^**](https://en.wikipedia.org/wiki/Web_framework#cite_ref-5) [*"CakePHP"*](http://cakephp.org/)*.*
  6. [**^**](https://en.wikipedia.org/wiki/Web_framework#cite_ref-6) *Thomson, Kris (2003-10-29).*[*"Clarification on MVC= Pull and MVC Push"*](http://www.theserverside.com/patterns/thread.tss?thread_id=22143)*. Retrieved 2007-07-29.*
  7. [**^**](https://en.wikipedia.org/wiki/Web_framework#cite_ref-7) [*"What are the fundamental differences between Struts and JSF"*](http://struts.apache.org/2.2.1/docs/what-are-the-fundamental-differences-between-struts-and-jsf.html)*. Struts.apache.org. 2011-02-14. Retrieved 2013-06-14.*
  8. [**^**](https://en.wikipedia.org/wiki/Web_framework#cite_ref-8) *Microsoft.*[*"Three-tiered distribution"*](http://msdn.microsoft.com/en-us/library/ff647546.aspx)*. Retrieved 2011-09-19.*
  9. [**^**](https://en.wikipedia.org/wiki/Web_framework#cite_ref-9) *Oracle.*[*"clustering\_concepts\_10en"*](http://download.oracle.com/docs/cd/E10316_01/cs/cs_doc_10/documentation/integrator/clustering_concepts_10en.pdf)*(PDF). Retrieved 2011-09-19.*
  10. [**^**](https://en.wikipedia.org/wiki/Web_framework#cite_ref-10) *Robert R. Perkoski.*[*"Introduction to Web Development"*](https://web.archive.org/web/20131107052241/http:/www.sis.pitt.edu/~perkoski/is1092/slides/chapter01.ppt)*. Archived from*[*the original*](http://www.sis.pitt.edu/~perkoski/is1092/slides/chapter01.ppt)*on 2013-11-07.*
  11. [**^**](https://en.wikipedia.org/wiki/Web_framework#cite_ref-11) *IBM.*[*"Using Client Access Express in a three tier environment"*](http://publib.boulder.ibm.com/html/as400/v5r1/ic2933/index.htm?info/rzaii/rzaiithreetier.htm)*. Retrieved 2011-09-19.*
  12. [**^**](https://en.wikipedia.org/wiki/Web_framework#cite_ref-12) *Oracle.*[*"Understanding the Three-Tier Architecture"*](http://download.oracle.com/docs/cd/B25221_05/web.1013/b13593/undtldev010.htm)*. Retrieved 2011-09-19.*
  13. [**^**](https://en.wikipedia.org/wiki/Web_framework#cite_ref-13) *Microsoft.*[*"Pragmatic Architecture: Layering"*](http://msdn.microsoft.com/en-us/library/aa905336.aspx)*. Retrieved 2011-09-19.*
  14. [**^**](https://en.wikipedia.org/wiki/Web_framework#cite_ref-14) *Arokia.*[*"3-Tier Web Architecture"*](http://www.arokiait.com/3-tire-web-architechture.htm)*. Retrieved 2011-09-19.*
  15. [**^**](https://en.wikipedia.org/wiki/Web_framework#cite_ref-15) [*"ASP.NET MVC Controller Best Practices"*](https://web.archive.org/web/20111011174215/http:/www.arrangeactassert.com:80/asp-net-mvc-controller-best-practices-%E2%80%93-skinny-controllers/)*. Archived from*[*the original*](http://www.arrangeactassert.com/asp-net-mvc-controller-best-practices-%E2%80%93-skinny-controllers/)*on 2011-10-11. Retrieved 2011-09-19.*
  16. [**^**](https://en.wikipedia.org/wiki/Web_framework#cite_ref-16) *Jamis Buck.*[*"Skinny Controller, Fat Model"*](https://web.archive.org/web/20150516021854/http:/www.arrangeactassert.com/asp-net-mvc-controller-best-practices-%E2%80%93-skinny-controllers/)*. Archived from*[*the original*](http://weblog.jamisbuck.org/2006/10/18/skinny-controller-fat-model)*on 2015-05-16.*
  17. [**^**](https://en.wikipedia.org/wiki/Web_framework#cite_ref-17) [*"Getting Started With Web Frameworks"*](https://www.wired.com/2010/02/get_started_with_web_frameworks/)*.*[*Wired Magazine*](https://en.wikipedia.org/wiki/Wired_Magazine)*. Retrieved 2018-04-02.*
  18. [**^**](https://en.wikipedia.org/wiki/Web_framework#cite_ref-18) *KLIMUSHYN, Mel.*[*"Web Application Architecture – Client-Side vs. Server-Side"*](https://spin.atomicobject.com/2015/04/06/web-app-client-side-server-side/)*. Atomic Spin. Retrieved 2016-03-06.*
  19. [**^**](https://en.wikipedia.org/wiki/Web_framework#cite_ref-19) [*"Home - Zend Framework"*](https://framework.zend.com/)*. framework.zend.com. Retrieved 2020-02-09.*
  20. [**^**](https://en.wikipedia.org/wiki/Web_framework#cite_ref-20) [*"AngularJS vs. Backbone.js vs. Ember.js"*](https://www.airpair.com/js/javascript-framework-comparison)*. www.airpair.com. Retrieved 2016-06-04.*
  21. ^ [Jump up to:***a***](https://en.wikipedia.org/wiki/Web_framework#cite_ref-:0_21-0) [***b***](https://en.wikipedia.org/wiki/Web_framework#cite_ref-:0_21-1) *Fowler, Martin.*[*"bliki: InversionOfControl"*](http://martinfowler.com/bliki/InversionOfControl.html)*. martinfowler.com. Retrieved 2016-03-06.*
  22. [**^**](https://en.wikipedia.org/wiki/Web_framework#cite_ref-22) *Xue, Qiang.*[*"Capital One Engineering – Philosophies that Shaped Successful Frameworks"*](http://www.capitalone.io/blog/successful-framework-philosophies/)*. www.capitalone.io. Retrieved 2016-03-06.*