Трехзвенная архитектура веб-приложения

JavaScript-фреймворки

Выполняют следующие основные функции:

1. Работа с DOM-деревом.

2. AJAX-запросы.

3. Удобные компоненты интерфейса (HTML-форм).

4. ООП (классы) JS поддерживает прототипное наследование.

Различные библиотеки корнями выросли из различных функций:

1. jQuery (DOM), jQueryUI, jQueryMobile – Twitter Bootstrap
2. Ext JS – наиболее мощная и универсальная, но платная.
3. Yahoo! UI (YUI) – интерфейс пользователя.
4. Dojo (додзё) – интерфейс пользователя. Позволяет создавать модульные элементы пользовательского интерфейса – dijit.
5. MooTools (ООП).

Дополнительно большое количество библиотек для создания графиков и диаграмм.

Данные библиотеки не являются новыми языками. Не смотря на то, что синтаксис jQuery кажется непохожим на обычный JavaScript, программа на jQuery является правильной JavaScript-программой. Никакой дополнительной компиляции (преобразований) не используется. Это связано с гибкостью синтаксиса JavaScript.

NODE.JS

JavaScript является серверным языком, разработка серверных приложений на JavaScript. Основан на «быстром» JavaScript-движке V8.

Новые языки, которые компилируются в JavaScript:

- CoffeeScript
- TypeScript
- Dart
- Другие языки (диалекты ЛИСПа)

Проект Node-WebKit – комбинация серверного и клиентского компонентов. Можно разрабатывать desktop-приложения на JavaScript.

4 поколения средств веб-разработки

1. Интерфейс CGI (Common Gateway Interface – общий интерфейс шлюзов)

2. Протокол программирования серверных приложений ISAPI

3. Технологии «серверных страниц»

- ASP – Active Server Pages (ASP.NET, ASP.NET MVC)
- PHP – Personal Home Pages
- JSP – Java Server Pages

4. MVC-фреймворки

Эта последовательность представляет собой основной путь развития технологий, хотя существуют и технологии, которые построены по другим принципам, например веб-технологии компании Lotus, NODE.JS.

1. Интерфейс CGI

Тим Бернерс-Ли создал только три основные компоненты веба:

- язык гипертекстовой разметки документов HTML (HyperText Markup Language);
- универсальный способ адресации ресурсов URI (Universal [Uniform] Resource Identifier);
- протокол обмена гипертекстовой информацией HTTP (HyperText Transfer Protocol – протокол передачи гипертекста).

Изначально веб был хранилищем статических документов.

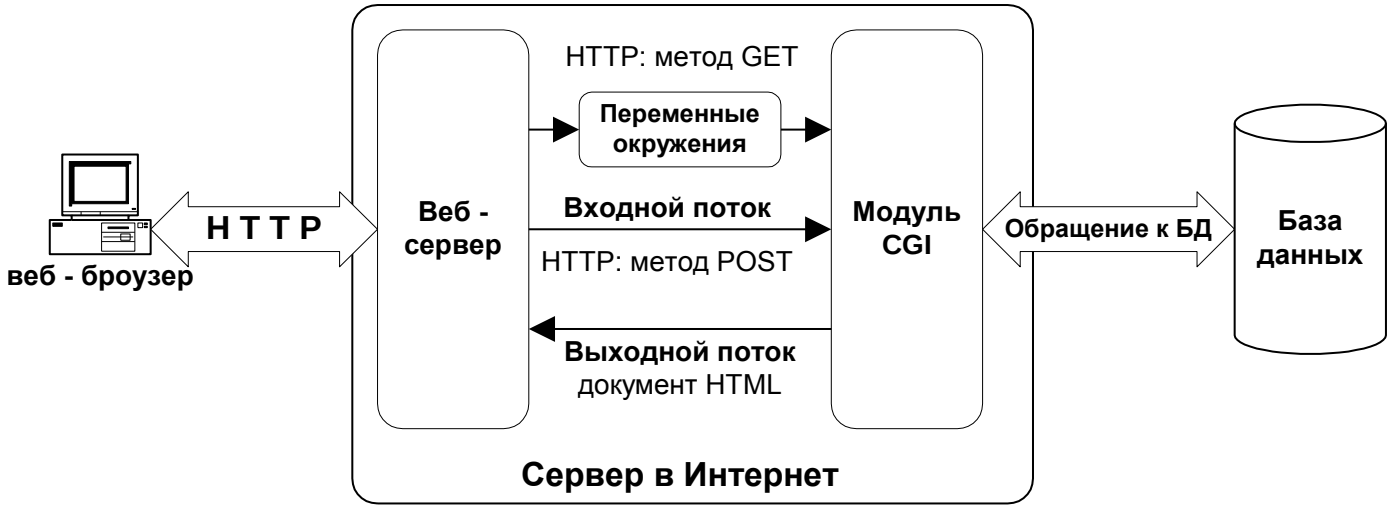
Позже, при разработке Mosaic, NCSA добавила к этим трем компонентам четвертый: универсальный интерфейс шлюзов CGI (Common Gateway Interface)

Время создания – около 1993 года.

CGI представляет собой интерфейс запуска внешней программы, которая может динамически формировать HTML-документы.

CGI является наиболее старой технологией, ее поддерживает большинство веб-серверов (даже самые простые).

Схема взаимодействия веб-сервера и модуля CGI:



Модуль CGI представляет собой исполняемый файл, вызываемый севером для обработки данных и генерации документа HTML.

Обмен данными между сервером и модулем CGI происходит через стандартные входной и выходной потоки.

В случае использования метода GET для получения параметров запроса используются переменные окружения операционной системы.

При использовании метода POST данные передаются через стандартный поток ввода.

В процессе работы модуль CGI генерирует документ HTML, который помещается в стандартный поток вывода, затем передается на сторону браузера, и браузер показывает его пользователю.

Для написания сценариев CGI используются как интерпретируемые языки (например Perl), так и компилируемые, C/C++, Pascal, Basic и т.д.

Основным достоинством технологии CGI является то, что появилась возможность динамически создавать документы HTML. Другим достоинством является то, что данная технология поддерживается большинством веб-серверов.

Основным недостатком технологии CGI является проблема увеличения нагрузки на веб-сервер. Каждая копия модуля CGI запускается на сервере в отдельном процессе. Таким образом, при одновременном поступлении запроса от 1000 пользователей будет одновременно запущено 1000 копий программы CGI.

Для решения данной проблемы был создан стандарт Fast-CGI, который позволяет избегать запуска лишних процессов, но он не нашел очень широкого применения.

Также для платформы Windows разработан стандарт Windows CGI (WinCGI). Основное отличие от CGI состоит в том, что вместо переменных окружения используется INI-файл.

Python использует технологию WSGI (Web Server Gateway Interface) – разновидность Fast-CGI.

На платформе Java существует технология сервлетов (Java servlet), которая по своему принципу напоминает технологию CGI. Java servlet – это программа на языке Java, которая выводит в выходной поток документ HTML, точно так же, как это делает программа CGI (но при этом CGI – интерфейс не используется, используются внутренние механизмы платформы Java).

2. Протокол программирования серверных приложений ISAPI

Этот протокол был разработан для преодоления ограничений и недостатков протокола CGI.

В отличие от CGI-технологии, технология ISAPI (Internet Server API) предусматривает реализацию модулей расширения веб-сервера в виде библиотек DLL.

При обработке запроса не создается отдельный процесс, как в технологии CGI, а вызывается модуль ISAPI, который загружается в память при первом обращении.

Существует несколько вариантов реализации протоколов серверных приложений. Чаще всего используются ISAPI (Microsoft) и NSAPI (Netscape).

Модули расширения ISAPI, как и модули CGI, принимают данные, передаваемые из браузера, обрабатывают эти данные и динамически формируют HTML-документ. Но вместо чтения переменных окружения и работы с входным и выходным потоками, модуль ISAPI обращается к специальным API-функциям для чтения параметров и вывода документа HTML.

Модуль ISAPI реализуется в виде библиотеки DLL, которая при первом обращении загружается в одно адресное пространство с веб-сервером. Это позволяет значительно повысить производительность сервера по сравнению с использованием технологии CGI.

Недостатки технологии ISAPI:

1. В отличие от CGI, технология ISAPI поддерживается только конкретным веб-сервером – Microsoft Internet Information Services (IIS).

2. Уменьшение надежности по сравнению с CGI.

При возникновении ошибки в модуле ISAPI может произойти аварийное завершение всего веб-сервера, а при возникновении ошибки в модуле CGI будет завершен только процесс CGI.

3. Достаточно большая трудоемкость разработки и отладки модулей ISAPI.

Технология ISAPI имеет достаточно узкую нишу: с ее помощью разрабатываются дополнительные компоненты веб-сервера IIS. Например, интерпретатор ASP-страниц разработан с использованием технологии ISAPI.

3. Технологии «серверных страниц»

Технологии «серверных страниц» объединяют в себе надежность технологии CGI и производительность, близкую к производительности ISAPI. При этом создание серверной страницы проще, чем создание CGI-модуля (и тем более модуля ISAPI).

При использовании технологии из группы технологий «серверных страниц» программа на сервере представляет собой документ HTML, в который встроены «серверные сценарии». В результате выполнения вместо исходного текста сценария в документ подставляются результаты работы сценария. Полученный статический HTML-документ передается в броузер.

4. MVC-фреймворки

Как правило является расширением технологии «серверных страниц». В некоторых технологиях являются расширением CGI, например Python использует технологию WSGI.

Наиболее известные примеры MVC-фреймворков:

- Django (Python)
- Ruby on Rails

- ASP.NET MVC

«M» - модель. Как правило, это отдельная библиотека (подфреймворк), который обеспечивает удобное взаимодействие с БД.

Часто используется технология ORM (Object-relational mapping). Объекты языка программирования отображаются на записи таблиц БД.

В ASP.NET MVC используется Entity Framework. Для запросов к данным используется технология LINQ.

Entity Framework использует три подхода:

Model First. Сначала создается (рисуется в конструкторе) модель БД (файл в формате XML, модель трехуровневая [схема БД]-[уровень связи]-[свойства класса]). По данной модели генерируется код на C# и DDL-script.

Database First. На основе существующей БД создается модель и код на C#.

Code First. Разрабатывается код классов модели. На основе кода генерируется модель и DDL-script.

Миграция – переход между версиями БД при доработке веб-приложения. Реализован только для **Code First.**

«V» - view (вид, шаблон, представление). Фактически в этом качестве может использоваться любая технология «серверных страниц». Часто используются отдельные шаблонизаторы (template engine - шаблонные движки), например Smarty в PHP.

В ASP.NET MVC используются:

- ASPX – классический шаблонизатор, который применяется в ASP.NET.
- Razor (cshtml) – более простой синтаксис, похожий на код C#.

«C» - controller (контроллер). В различных фреймворках системы контроллеров реализованы различным образом. Основная задача контроллера – обработка действий пользователя.

В ASP.NET MVC любое нажатие пользователя на гиперссылку или кнопку отправки формы приводит к вызову какого-либо метода одного из классов-контроллеров. То есть при нажатии на гиперссылку вызывается серверный код на C#.

Функциональные языки для веб-разработки

Функциональный язык Erlang для разработки высоконагруженных систем. Elixir – Ruby-подобный язык на Erlang-машине.

Сервер СУБД

Используется классический реляционный подход или NoSQL.

Классическая реляционная СУБД – Oracle, MS SQL Server, MySQL, PostgreSQL.

ASP.NET MVC (Entity Framework) прежде всего ориентирован на работу с MS SQL Server.

Часто в веб-приложениях используются NoSQL СУБД.

Чаще всего применяются документо-ориентированные. Mongo DB, Couch DB. Содержат встроенный веб-сервер. Могут возвращать JSON. К таким СУБД возможно прямое обращение из JavaScript (jQuery).

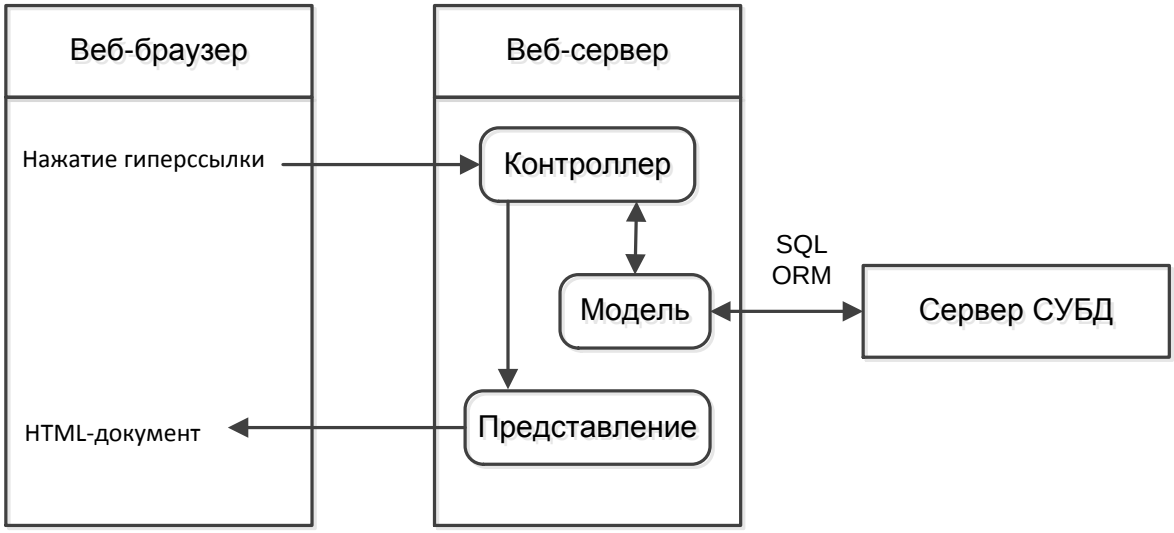
XML-СУБД. Содержат отдельные средства разработки веб приложений (XQuery), могут быть использованы с другими технологиями.

BigTable – в случае использования облачных технологий.

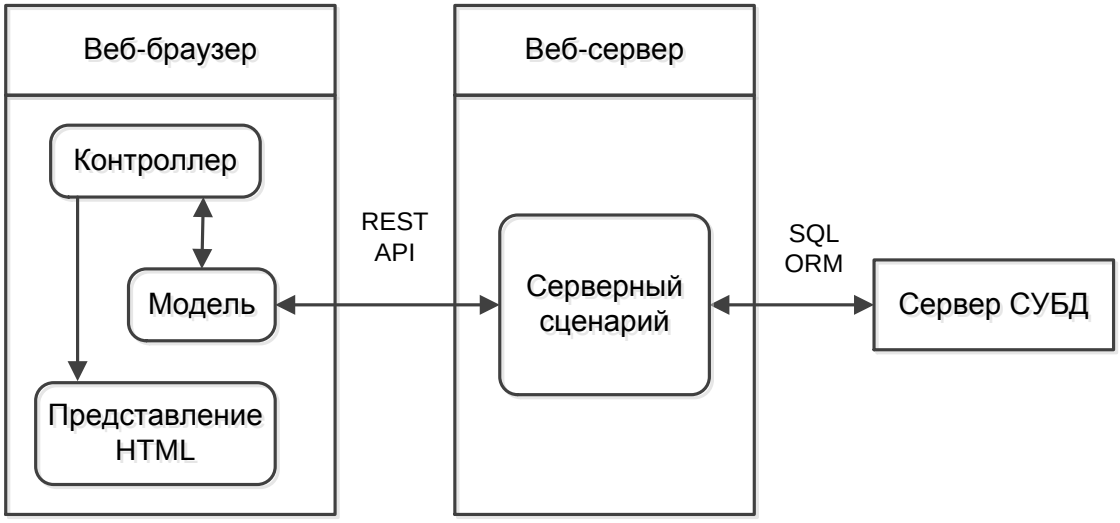
Графовые СУБД – в случае работы со знаниями.

Использование MVC-фреймворков на стороне клиента и сервера

Использование MVC-фреймворка на стороне веб-сервера. Схема с «толстым» сервером и «тонким» клиентом.



Использование MVC-фреймворка на стороне веб-браузера. Схема с «тонким» сервером и «толстым» клиентом.



Примеры MVC-фреймворков на стороне веб-браузера:

- AngularJS
- KnockoutJS
- Другие примеры на сайте - https://github.com/tastejs/todomvc