# http://www.findpatent.ru/patent/222/2227372.html Патентный поиск

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Предмет поиска | Страна выдачи и номер охранного документа, классификационный индекс | Заявитель с указанием страны, номер заявки, дата приоритета, конвенционный приоритет, дата публикации | Сущность заявленного технического решения и цели создания (по описанию изобретения или опубликованной заявке) | Сведения действительного документа или причина аннулирования |
| 1 | 2 | 3 | 4 | 5 |
|  | USA  US 7653652 B2  [1/1](https://www.google.com/url?id=FM8LBgABERAJ&q=http://www.uspto.gov/web/patents/classification/uspc001/defs001.htm&usg=AFQjCNG37WdtIPCwBOhdSwF0dt2rkf76Nw#C001S001000), [707/999.103](https://www.google.com/url?id=FM8LBgABERAJ&q=http://www.uspto.gov/web/patents/classification/uspc707/defs707.htm&usg=AFQjCNE7Q7Bg2eD2wcE_fXEcdOe7Yesevw#C707S999103) | Raxit A. Kagalwala, John Patrick Thompson  Stated 17.08.2004  US 10/919,682  Publ. 26.01.2010 | **DATABASE SCHEMA FOR STRUCTURED QUERY LANGUAGE (SQL) SERVER**  This invention relates to databases and to database management schemas.  A database schema described herein is an extension of the CIM core model. It defines classes, properties, methods, and associations for a SQL (structured query language) database. Although a specific embodiment is disclosed herein, it should be recognized that variations of the described embodiment are possible while still remaining within the scope of the appended claims. |  |
|  | USA  US 8676769 B2 707/693 | Marco Costaglio  Stated 27.04.2009  US 12/430,156  Publ. 18.03.2014 | **METHOD FOR DEFRAGMENTING INDEXES IN A RELATIONAL DATABASE AND SYSTEM FOR PERFORMING THE METHOD**  A method and a system defragment indexes in a relational database. The relational database contains a set of tables, in which each table contains a set of indexes, in which, for each index, a fragmentation parameter is associated indicating the fragmentation level of the index. The method includes the following steps: getting a list of the set of tables; for each table of the list, getting the fragmentation parameter associated to each index of the table; and depending on the value of the fragmentation parameter, defragmenting the associated index. |  |
|  | USA  US 9355077 B2 G06F17/30 | Ivan Osmak  Stated 13.03.2013  US 13/799,431  Publ. 31.05.2016 | **CONTENT MANAGEMENT SYSTEM EMPLOYING A HYBRID WEB APPLICATION FRAMEWORK**  A method may include receiving, via a network, a request for browser-renderable content, and determining, by a processor of a computing device, a web framework, where the web framework includes at least one ASP.NET™ control and at least one ASP.NET™ MVC controller. The method may include determining, by the processor, first instructions, where the first instructions correspond to the at least one ASP.NET™ control, and the first instructions include a browser-renderable language. The method may include determining, by the processor, second instructions, where the second instructions correspond to the at least one ASP.NET™ MVC controller, and the second instructions include the browser-renderable language The method may include combining, by the processor, the first instructions and the second instructions to determine the browser-renderable content, and providing, via the network, the browser-renderable content. |  |
|  | Germany  EP 2485457 A1 [H04L29/08](https://www.google.com/url?id=8yCQBwABERAJ&q=http://web2.wipo.int/ipcpub/&usg=AFQjCNER44F5jlVoswCkvW3YEcB5lW4moA#refresh=page&notion=scheme&version=20130101&symbol=H04L0029080000) | Marco Fioritoni, Nadia Giusti  Stated 4.02.2011  EP20110153373  Publ. 8.08.2011 | **A METHOD FOR KEEPING A WEB SESSION ALIVE IN A WEB APPLICATION**  There is a need for an improved method to keep a web session in a web application alive for a determinable amout of time allowing to keep a limited amount of data on the Internet Information Server and manage individually the session timeout characteriscs of the web session in the web application.  This method therefore allows to automaticall refresh the web pages in the background behind without any further user interaction. Thereby, the portion of code added effectively creates a hidden postback that keeps alive the session's state. By means of the class library, the user is enabled at engineering level to manage the session timeout state and to eliminate a premature timeout thereby eliminating any risk of possible data loss. |  |
|  | USA  US 20140068557 A1 717/116 | Mikus Vanags, Arturs Licis, Janis Justs  Stated 22.02.2013  US 13/773,662  Publ. 6.03.2014 | **STRONGLY TYPED METADATA ACCESS IN OBJECT ORIENTED PROGRAMMING LANGUAGES WITH REFLECTION SUPPORT**  Type safety is important property of any type system. Modern programming languages support different mechanisms to work in type safe manner, e.g., properties, methods, events, attributes (annotations) and other structures, but none of the existing, general purpose, programming languages which support reflection provide type safe type (class/structure) member metadata access. Existing solutions provide no or limited type safety which are complex and processed at runtime which by definition is not built-in type-safe metadata access, but only more or less type safe workarounds called “best practices”. Problem can be solved by introducing method for type safe type member metadata access which could be processed at compile time. |  |
|  | USA  US 20050005158 A1 726/22 | Eyal Alaluf  Stated 29.06.2004  US 10/880,396  Publ. 06.01.2005 | **METHOD FOR COMPILING AN ACTIVE SERVER PAGE (ASP).NET WEB SERVICE INTO A JAVA COMPLIANT WEB SERVICE**  A method is disclosed for an attribute compiler for compiling an active server page (ASP).Net Web service into a Java compliant Web service and enabling execution of Web applications having at least one Web service attribute. The method includes detecting at least one Web method in a .Net assembly file of a respective .Net class. The method also includes generating a Web service file comprising at least one instance of a backend component and the Web method and generating the at least one instance of a backend component for implementing the Web method, such that the active server page (ASP).Net Web service can be compiled into a Java compliant Web service. |  |
|  | USA  US 7162723 B2 719/318 | Scott D. Guthrie, Dmitry Robsman  Stated 29.06.2001  US 09/894,828  Publ. 9.01.2007 | **ASP.NET HTTP RUNTIME**  An HTTP request handling runtime includes a context object and an event pipeline for processing an HTTP request received at a host application from a client application. The context object logically represents the HTTP request and encapsulates at least one property associated with the received HTTP request. The context object is processed by the event pipeline, which includes a plurality of synchronous and/or asynchronous request events. A callback is generated when a request event is raised and when at least one application and/or module is registered with the request event. Each callback instantiates each application and module that is registered with the request event for processing the context object. The plurality of request events can include events that are in a deterministic order and/or request events that are in a non-deterministic order, such as an error event. |  |
|  | USA  US 20150012809 A1 715/234 | Nishant Kaushik, Yash Kumar Gupta  Stated 13.07.2013  US 13/934,417  Publ. 8.01.2015 | **METHOD AND APPARATUS FOR TRANSLATING JAVA-SCRIPT ACROSS DIFFERENT HOST ENVIRONMENTS**  A computer implemented method and apparatus for translating JAVASCRIPT code across different host environments.  The method comprises accessing a document comprising JAVASCRIPT code of a first type; parsing the JAVASCRIPT code of the first type in the document; tracking a location of an event or action in the document in which the JAVASCRIPT code of the first type is present; translating the parsed JAVASCRIPT code of the first type into JAVASCRIPT code of a second type; merging the translated JAVASCRIPT code into a location in a translated JAVASCRIPT file, wherein the location corresponds to the location tracked in the document; and exporting the translated JAVASCRIPT file. |  |
|  | USA  US 8954989 B1 719/313 | Paul Colton, Uri Sarid, Kevin Edward Lindsey  Stated 18.11.2008  US 12/273,539  Publ. 10.02.2015 | **FLEXIBLE, EVENT-DRIVEN JAVASCRIPT SERVER ARCHI-TECTURE**  A script server architecture is disclosed herein. The script server architecture includes a core comprising means for creating a DOM node in a HTML document, a framework written in JavaScript, the framework comprising a plurality of registered event handlers for performing the business logic of the script server architecture and means for modifying the DOM of a HTML document, and means for bridging the core and the framework. |  |
|  | USA  US 6892200 B2 707/704 | Brendan Eich  Stated 30.10.2002  US 10/285,043  Publ. 10.05.2005 | **JAVASCRIPT ENGINE**  A JavaScript enhancement is provided that increases performance of the JavaScript by avoiding the use of locking mechanisms in multi-threaded sessions. To do so, an object is deemed owned by a thread exclusively until the script is ended or suspended. Because scripts are typically short and complete quickly, the overhead to start and end a request containing a script is much less than the overhead of locking each and every object access done by the script, on average. |  |
|  | USA  US 9424163 B2 G06F9/44 | Robert A. Paveza, Patrick Nelson, Samuel Ng, Gregg B. Miskelly, Gearard Boland  Stated 15.01.2014  US 14/155,923  Publ. 23.08.2016 | **EXCEPTION AND DEBUGGING BEHAVIORS FOR JAVASCRIPT DEBUGGING USING JUST MY CODE**  Just My Code debugging allows developers to work on problems within their own code without noise from libraries, plugins and other unrelated code. Typical debugger behaviors may be modified based upon the identification or characterization of JavaScript code as “My Code,” “Library Code,” or “Unrelated Code.” The debugger behaviors that may be modified depending upon the code's characterization include, for example, first-chance exception handling, unhandled exception handling, the behavior of the debugger at explicit user breakpoints, the behavior when a user directs the debugger to break upon the next statement, and the behavior of the debugger when the user executes a stepping gesture. The debugger's visualizes core elements of the program state, such as the call stack, depending upon the identification or characterization of the code. |  |