

Chapter 1 : CiteSeerX " Citation Query Effective Enterprise Java

Effective Enterprise Java calendrierdelascience.com, calendrierdelascience.com, calendrierdelascience.com, calendrierdelascience.com, calendrierdelascience.com Download Note: If you're looking for a free download links of Effective Enterprise Java pdf, epub, docx and torrent then this site is not for you.

The goals of J2EE. The ten fallacies of enterprise computing. Prefer components as the key element of development, deployment, and reuse. Prefer loose coupling across component boundaries. Differentiate layers from tiers. Keep data and processors close together. Remember that identity breeds contention. Use hook points to inject optimizations, customizations, or new functionality. Be robust in the face of failure. Define your performance and scalability goals. Restrict EJB to transactional processing. Never optimize without profiling first. Recognize the cost of vendor neutrality. Build in monitoring support. Build in administration support. Make deployment as simple as possible. Understand all your communications options. Consider your lookup carefully. Recognize the cost of network access. Prefer context-complete communication styles. Prefer data-driven communication over behavior-driven communication. Avoid waiting for remote service requests to respond. Consider partitioning components to avoid excessive load on any one machine. Consider using Web Services for open integration. Pass data in bulk. Consider rolling your own communication proxies. Prefer rules engines for complex state evaluation and execution. Prefer transactional processing for implicitly nonatomic failure scenarios. Differentiate user transactions from system transactions. Never cede control outside your component while holding locks. Understand EJB transactional affinity. Prefer local transactions to distributed ones. Consider using optimistic concurrency for better scalability. Consider using pessimistic concurrency for explicit concurrency control. Consider lower isolation levels for better transactional throughput. Use savepoints to keep partial work in the face of rollback. Replicate resources when possible to avoid lock regions. Favor the immutable, for it needs no locks. Use objects-first persistence to preserve your domain model. Use relational-first persistence to expose the power of the relational model. Use procedural-first persistence to create an encapsulation layer. Recognize the object-hierarchical impedance mismatch. Use in-process or local storage to avoid the network. Never assume you own the data or the database. Lazy-load infrequently used data. Eager-load frequently used data. Batch SQL work to avoid round-trips. Know your JDBC provider. Consider rich-client UI technologies. Separate presentation from processing. Keep style separate from content. Pregenerate content to minimize processing. Validate early, validate everything. Security is a process, not a product. Remember that security is not just prevention. Establish a threat model. Always validate user input. Turn on platform security. Use SignedObject to provide integrity of Serialized objects. Use SealedObject to provide confidentiality of Serializable objects. Use GuardedObject to provide access control on objects.

Chapter 2 : Effective Enterprise Java : Ted Neward :

*Effective Enterprise Java [Ted Neward] on calendrierdelascience.com *FREE* shipping on qualifying offers. With this book, Ted Neward helps you make the leap from being a good Java enterprise developer to a great developer! â€”John Crupi.*

The ten fallacies of enterprise computing. Prefer components as the key element of development, deployment, and reuse. Prefer loose coupling across component boundaries. Differentiate layers from tiers. Keep data and processors close together. Remember that identity breeds contention. Use hook points to inject optimizations, customizations, or new functionality. Be robust in the face of failure. Define your performance and scalability goals. Restrict EJB to transactional processing. Never optimize without profiling first. Recognize the cost of vendor neutrality. Build in monitoring support. Build in administration support. Make deployment as simple as possible. Understand all your communications options. Consider your lookup carefully. Recognize the cost of network access. Prefer context-complete communication styles. Prefer data-driven communication over behavior-driven communication. Avoid waiting for remote service requests to respond. Consider partitioning components to avoid excessive load on any one machine. Consider using Web Services for open integration. Pass data in bulk. Consider rolling your own communication proxies. Prefer rules engines for complex state evaluation and execution. Prefer transactional processing for implicitly nonatomic failure scenarios. Differentiate user transactions from system transactions. Never cede control outside your component while holding locks. Understand EJB transactional affinity. Prefer local transactions to distributed ones. Consider using optimistic concurrency for better scalability. Consider using pessimistic concurrency for explicit concurrency control. Consider lower isolation levels for better transactional throughput. Use savepoints to keep partial work in the face of rollback. Replicate resources when possible to avoid lock regions. Favor the immutable, for it needs no locks. Use objects-first persistence to preserve your domain model. Use relational-first persistence to expose the power of the relational model. Use procedural-first persistence to create an encapsulation layer. Recognize the object-hierarchical impedance mismatch. Use in-process or local storage to avoid the network. Never assume you own the data or the database. Lazy-load infrequently used data. Eager-load frequently used data. Batch SQL work to avoid round-trips. Know your JDBC provider. Consider rich-client UI technologies.

Chapter 3 : Effective Enterprise Java | InformIT

If you want to build better Java enterprise applications and work more efficiently, look no further. Inside, you will find an accessible guide to the nuances of Java 2 Platform, Enterpris "With this book, Ted Neward helps you make the leap from being a good Java enterprise developer to a great developer!"

There present variety of reasons behind it due to which the readers quit reading the eBooks at their first most effort to make use of them. However, there exist some techniques that could help the readers to have a nice and effective reading experience. Someone ought to fix the proper brightness of screen before reading the eBook. It is a most common problem that many of the people usually bear while using an eBook. As a result of this they suffer from eye sores and head aches. The very best alternative to overcome this acute difficulty is to reduce the brightness of the screens of eBook by making particular changes in the settings. A great eBook reader ought to be set up. You may also make use of free software that can provide the readers that have many functions to the reader than only a simple platform to read the desirable eBooks. Apart from offering a place to save all your valuable eBooks, the eBook reader software even provide you with a large number of characteristics as a way to boost your eBook reading experience in relation to the conventional paper books. You can even improve your eBook reading experience with help of choices provided by the software program such as the font size, full screen mode, the certain number of pages that need to be shown at once and also change the color of the backdrop. You should not use the eBook consistently for a lot of hours without breaks. You need to take appropriate breaks after specific intervals while reading. Nevertheless, this does not mean that you need to step away from the computer screen every now and then. Continuous reading your eBook on the computer screen for a long time without taking any break can cause you headache, cause your neck pain and suffer with eye sores and in addition cause night blindness. So, it is necessary to provide your eyes rest for a while by taking breaks after specific time intervals. This can help you to prevent the problems that otherwise you may face while reading an eBook always. While reading the eBooks, you need to prefer to read large text. Typically, you will note that the text of the eBook tends to be in moderate size. It is suggested to read the eBook with large text. So, raise the size of the text of the eBook while reading it at the display. Even though this will definitely mean that you will have less text on every page and greater amount of page turning, you will manage to read your wanted eBook with great convenience and have a great reading experience with better eBook screen. It is suggested not to go for reading the eBook in fullscreen mode. Though it might appear simple to read with full screen without turning the page of the eBook fairly often, it put ton of anxiety in your eyes while reading in this mode. Consistently prefer to read the eBook in exactly the same length that would be similar to the printed book. This is so, because your eyes are used to the span of the printed book and it would be comfy that you read in the same way. Try various shapes or sizes until you find one with which you will be comfortable to read eBook. By using different techniques of page turn you could also enhance your eBook encounter. You can try many ways to turn the pages of eBook to improve your reading experience. Check out whether you can turn the page with some arrow keys or click a specific section of the display, apart from utilizing the mouse to manage everything. Favor to make us of arrow keys if you are leaning forward. Lesser the movement you must make while reading the eBook better will be your reading experience. This will definitely help to make reading easier. By using each one of these powerful techniques, you can definitely boost your eBook reading experience to a great extent. This advice will help you not only to prevent particular risks that you may face while reading eBook often but also facilitate you to relish the reading experience with great relaxation. The download link provided above is randomly linked to our ebook promotions or third-party advertisements and not to download the ebook that we reviewed. We recommend to buy the ebook to support the author. Thank you for reading. Search a Book Search Recommended Books.

Chapter 4 : Effective Enterprise Java by Ted Neward

Just three weeks ago, I was asked to step in for Ted Neward to give a tutorial at Å–redev on Effective Enterprise calendrierdelascience.com I did not have time to get the tutorial materials printed, I present them here on the web for the participants and others.

As I did not have time to get the tutorial materials printed, I present them here on the web for the participants and others. Effect Enterprise Java architecture in Since the Effective Enterprise Java book was written, many of the topics regarding transactions, concurrency and shared state have been resolved. Here are the basic guidelines of an enterprise application in Java as of All processing is triggered by an event, such as an http-request, a timer or an incoming message Each processing event is handled in an isolated scope, never touching the data of another processing event. All coordination of data happens through the data layer. This means that objects are either stateful and short-lived or stateless and immortal. Each processing event is either completed or aborted totally. Very few applications will benefit from trying to automatically recover from most problems. Inconsistent updates are resolved when transactions are committed, usually through optimistic locking. Some things I told the attendants to consider: First, today most people consider EJBs to be more trouble than value with the exception of Entity beans 3. Second, all triggers can be forged. We return to the second issue when we discuss security. Web integration testing I showed a practical demo using WebDriver and Jetty to perform web integration tests as JUnit test. The remarkable things about this example is that it requires no installation of an app server Jetty is installed as a Maven dependency , it requires no separate starting of an application server Jetty can run embedded in the test and it is very fast Jetty starts up in about milliseconds. Hibernate integration testing I showed a practical example of how to test a DAO implemented with Hibernate. The remarkable things about this demonstration was that, again, no installation or startup is required I use H2 as an in-memory database. Hibernate is a power tool. I use the following analogy: If you want to remove a rock from you back yard, you may want to use dynamite. Hibernate is like that dynamite. You need knowledge and safety measures to deal with it correctly, but when you do, it can save you a lot of effort. Creating JUnit tests for your Hibernate code is one such safety measure. Security Almost all the threats an application developer should be concerned with are in the same class, namely that of Injection attacks. An injection attack is when a client tricks another process into treating data as instructions. For example by using SQL meta-characters: Never use tainted data in unsafe ways. When writing HTML pages, always escape tainted data. Another often overlooked exploit is request forgery, often used in combination with phishing attacks. To protect your users from request forgery, supply an authentication token as a hidden field with all forms. Make sure all operations have confirmation dialogs. Continuous Deployment Continuous Deployment is the practice of rolling out a deployment to a server after every successful build on your Continuous Integration server. I described two ways of doing Continuous Deployment during the tutorial, but I will restrict this discussion to the more modern one. Most teams doing continuous deployment use Maven or Ant to invoke the deployment tools of their respective application servers. Many application servers make this pretty hard, but the hardest part of the battle if finding out what command needs to be invoked. The Continuous Integration server can be configured to run this task. After doing deployment, it is a good idea to run some sort of system level integration tests. If you keep your logs clean, you can actually gain quite a bit of confidence just by looking at the logs after applying simulated load to your system. Some projects take this even further, by continuously deploying to production. Summary Effective Enterprise Java development has progressed a lot since Much of the emphasis now is on how to improve testing in enterprise Java applications. The way applications usually process data has stabilized as well, with most application preferring each event to be processed in an isolated, transactional context with very little automated recovery. In the end, Effective Enterprise Java is a lot simpler in than it was in

Chapter 5 : Effective Enterprise Java: Preface, State Management

*"With this book, Ted Neward helps you make the leap from being a good Java enterprise developer to a great developer!"--John Crupi, Sun Distinguished Engineer coauthor, Core J2EE Patterns*If you want to build better Java enterprise applications and work more efficiently, look no further.

Foreword Untitled Document Designing and implementing large-scale enterprise systems is hard. Building effective enterprise Java deployments is even harder. I see these difficulties on a daily basis. When consulting on enterprise projects, I see the real-world issues that developers are facing. I have also seen discussions, frustrations, and solutions to some of the issues on a daily basis on TheServerSide. It was the water cooler that allowed us to chat about solutions that worked for us, and it saw the growth of enterprise Java patterns. Developing for the enterprise is a very different beast when compared to building smaller, standalone applications. We have to consider issues that we can safely ignore in the other world. As soon as we have to share data among multiple users, we start down the enterprise path. Then we start facing questions: What is the best solution for allowing concurrency to this data? How coherent and correct does it have to be? How can we scale up from 2 to 50 to 1, clients? Well, simply answering the questions may not be the correct focus. We need to be taught about the various issues involved and shown techniques that can help with the various problems. No book has attacked these problems quite like Effective Enterprise Java does. The most important part of this book is that it teaches you two things really well. You will understand the general issues of enterprise computing. These enterprise problems are far from new. Ted has been around the block, and he understands the core issues at work. A non-Java developer would get a lot out of this book for this very reason. What you learn here will be with you for as long as you develop enterprise solutions. The language and APIs may change, but you will understand the issues in building a good architecture, the options you have for communication, the choices for where to store state, the various security concerns, and so much more. You will be able to attack the problems by using enterprise Java. Although the book offers genuine insight into the general enterprise problems, it also gives you tools to solve them with enterprise Java today. You will understand more about where the various enterprise Java technologies fit together. When would you use Web Services? What can messaging do for you? What is EJB good for? This book provides answers to these questions. It is great to have some answers to these common questions. The style of the book, in which you are given a set of "effective items," gets right to the point. Get stuck in, and enjoy the ride!

Chapter 6 : Effective enterprise Java - PDF Free Download

*Find out how the Java EE Security API supports enterprise security in cloud and microservices platforms, while introducing modern capabilities such as context and dependency injection. FREE DOWNLOAD** EFFECTIVE ENTERPRISE JAVA PDF.*

Chapter 7 : Rockford Lhotka - Effective Enterprise Java

Read Effective Enterprise Java Introduction or Effective Enterprise Java Chapter 5 or Download the chapters About the Author Ted Neward - Ted Neward is a Java calendrierdelascience.com author, instructor and speaker who lives in the Sacramento, CA area.

Chapter 8 : Pearson Education - Effective Enterprise Java

Stay ahead with the world's most comprehensive technology and business learning platform. With Safari, you learn the way you learn best. Get unlimited access to videos, live online training, learning paths, books, tutorials, and more.

Chapter 9 : Effective Enterprise Java at Ã–redev Â» Thinking Inside a Bigger Box

DOWNLOAD PDF EFFECTIVE ENTERPRISE JAVA

"Effective Enterprise Java" is similar in format and quality to "Effective Java." It is written at a higher level as it applies to J2EE systems, rather than the Java language. The 75 items include architecture/design (batch SQL to avoid server round trips) and coding level (serialVersionUID) tips.