Developing cloud-native applications with Eclipse and the Spring Tool Suite

Martin Lippert - Pivotal
mlippert@pivotal.io
@martinlippert
back in time...
Applications of that time
I hold Microsoft Word responsible for inhibiting the flow of language, for imposing arbitrary standards upon the nature of text, for pushing technologies of editing and revision where they don’t belong.

protected void setUp() {
    fEmpty = new Vector();
    fFull = new Vector();
    fFull.addElement(new Integer(1));
    fFull.addElement(new Integer(2));
    fFull.addElement(new Integer(3));
}

if (fFull == Vector [id=23]) {
    capacity = 10;
    elementCount = 0;
    elementData = Object [id=38];
}
Welcome

- **Downloads** are now available.

- **Welcome to the Eclipse Open Source Project**
  We finally made it! Eclipse is now an open source project! Thanks to everyone in the Eclipse community who supported us and worked so hard to get us this far. If you are just now joining us, Eclipse is a kind of universal tool platform - an open extensible IDE for anything but nothing in particular. The real value comes from tool plug-ins that "teach" Eclipse how to work with things - Java files, web content, graphics, video - almost anything you can imagine. Eclipse allows you to independently develop tools that integrate with other people's tools so seamlessly you won't know where one tool ends and another starts. The very notion of a tool as we know it... disappears completely...

  Find out what Eclipse is all about - check out the white paper, read some technical articles, visit the newsgroup, take a look at the projects, and pick up the latest downloads. Eclipse is a work in progress, but people are building some pretty cool things with it right now!

- **Legal Stuff**
  Legal Information about the content and use of the Eclipse Project website.

News

- Nov 7 - **Eclipse becomes an open source project.**
- Nov 7 - **Latest 2.0 development stream builds** have been posted!

Featured Articles

- **Creating an Eclipse View** Dave Springgay 11/2/01
  In the Eclipse Platform a view is typically used to navigate a hierarchy of information, open an editor, or display properties for the active editor. This article the design and implementation of a view will be examined in detail. You’ll learn how to create a simple view based on SWT, and a more advanced view using the JFace viewer hierarchy. We’ll also look at ways to achieve good integration with many of the existing features in the workbench, such as the window menu and toolbar, view linking, workbench persistence and action extension.

- **Contributing Actions to the Eclipse Workbench** Simon Arsenault 10/18/01
  The Eclipse Platform is an open and extensible platform. This article explains in detail how the Workbench can be extended to add new actions and provides guidance to the plug-in developers on how they can design for extensibility.

- **Into the Deep End of SWT StyledText Widget** Lynne Kues and Knut Radoff 10/22/01
  This is the second of two articles on the SWT StyledText widget. This article dives into some of the more advanced concepts of StyledText and builds on the previous article "Getting Your Feet Wet With the SWT StyledText Widget".

- **Using Perspectives in the Eclipse UI** Dave Springgay 8/7/01
  In the Eclipse Platform a Perspective determines the visible actions and views within a window. Perspectives also go well beyond this by providing mechanisms for task oriented interaction with resources in the Eclipse Platform, multi-tasking and information filtering. In this article the concepts behind perspectives are examined. The process for perspective definition, creation and instantiation will also be covered in detail with coding examples and sample scenarios.

- **Eclipse Overview White Paper** The Eclipse Project Team 7/31/01
  The Eclipse Platform is designed for building integrated development environments (IDEs) that can be used to create applications as diverse as web sites, embedded Java™ programs, C++ programs, and Enterprise JavaBeans(TM). This paper is a general technical introduction to the Eclipse Platform. Part II is a case study of how the Eclipse Platform was used to build a full-featured Java development environment.

All Articles...

For problems with the eclipse.org site, please contact the webmaster.
Current Releases

Java™ 2 Platform, Standard Edition v 1.4 Beta 3

Java™ 2 SDK, Standard Edition, v 1.4 Beta 3 (SDK)
Java™ 2, Runtime Environment, Standard Edition, v 1.4 Beta 3 (JRE)
Java™ 2 Platform, Standard Edition, v 1.4 Beta 3 Documentation (Docs)

Java™ 2 Platform, Standard Edition v 1.3.1

Java™ 2 SDK, Standard Edition, v 1.3.1 (SDK)
Java™ 2 Runtime Environment, Standard Edition, v 1.3.1 (JRE)
Java™ 2 Platform, Standard Edition, v 1.3.1 Documentation (Docs)

Related Technologies

Java™ Plug-in
Java™ Web Start
Java HotSpot™ Server Virtual Machine

Previous Releases

Java 2 SDK, Standard Edition, v 1.2.2
Java 2 Runtime Environment, Standard Edition, v 1.2.2
Java Development Kit 1.1.x (JDK™ 1.1.x)
Java Runtime Environment 1.1.x (JRE 1.1.x)
Today?
Cloud-Native Apps
No contracts. No catch. No worries.

Join the internet TV revolution.

Start Your Free Month
How does that change application development?
the good old days

app/web server

big monolith WAR

Data Access

Business Logic

UI

the database
need to scale?

app/web server

big monolith WAR

Data Access

Business Logic

UI

the database
need to scale?
many small service apps
the world has changed...
changing a **running** system
working on a few
too large to run everything locally on your machine
We need to change the way we work with microservices in IDEs
Demo
Thanks

Q&A
(and don’t forget to evaluate)

Martin Lippert - Pivotal
mlippert@pivotal.io
@martinlippert

download the Spring Tool Suite via: http://spring.io/tools
Evaluate the Sessions

Sign in and vote at eclipsecon.org