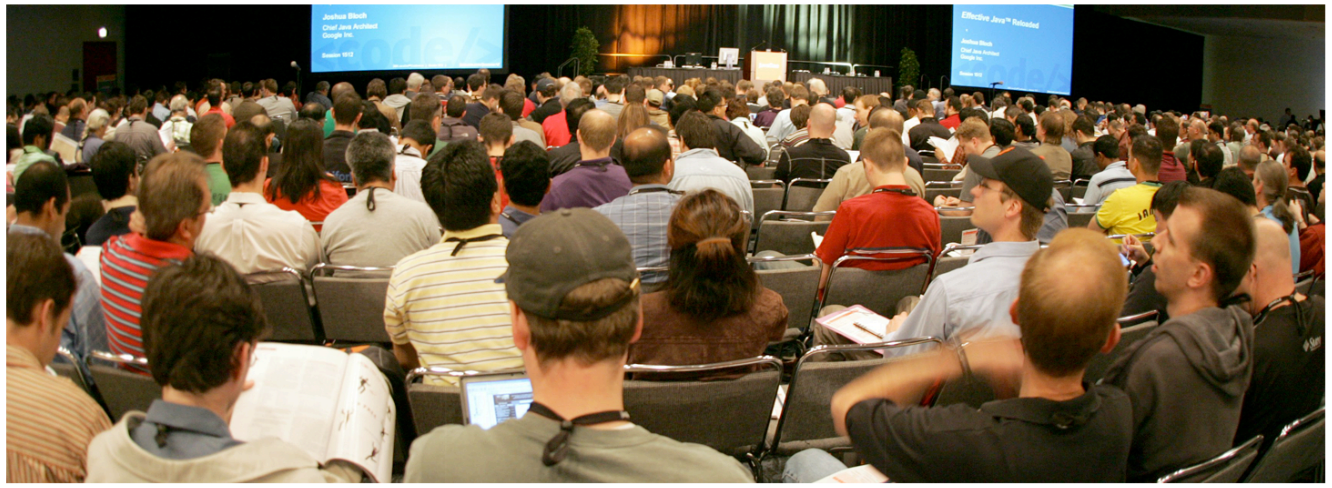


Grails, Trails, and Sails: Rails Through a Coffee Filter

Matt Hughes
David Esterkin

Chariot Solutions
http://chariotsolutions.com

BOF-9843

Agenda

Brief History of Web Development

Ruby On Rails

Sails

Trails

Grails

The Future of *ails

Agenda

Brief History of Web Development

Ruby On Rails

Sails

Trails

Grails

The Future of *ails

A Brief History of Web Application Development

In the beginning there was pain

. . .

then came Ruby on Rails

Agenda

Brief History of Web Development

Ruby On Rails

Sails

Trails

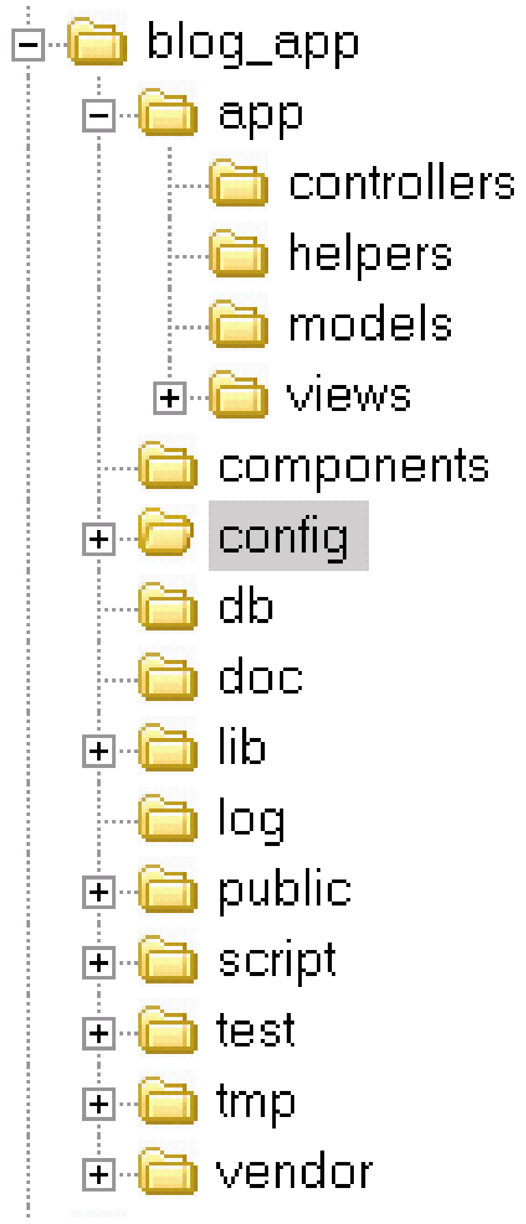
Grails

The Future of *ails

Rails Screencast

```
rails blog_app
```

Gives You...


```
blog_app
  app
    controllers
    helpers
    models
    views
  components
  config
  db
  doc
  lib
  log
  public
  script
  test
  tmp
  vendor
```

A functional CRUD app in 15 minutes

Ruby on Rails

Convention over Configuration

MVC

Opinionated Software

80/20 Rule

Don't Repeat Yourself

Test Driven Development

Agile

Get Real

Rails Dissected

ActiveRecord	Model
ERB	View
ActionController	Controller

Ruby

State of Java Web Development

- Coincides with
  - Disillusioned with EJB 2.x
  - Code, compile, deploy, restart server cycle
  - Popularity of dynamic languages on the JVM
  - Realization that Enterpriseyness != Self-Worth

RAILS

The Contenders

Trail	Domain Driven Design
Sail	Controller-centric
Grail	DDD / Full stack

Agenda

Brief History of Web Development

Ruby On Rails

Sails

Trails

Grails

The Future of *ails

- Started in 2005
- Brings the flavor of Rails development to Java
- Viento: custom template engine
- Rigging: custom dependency injection library

Similar to *ails

- Generates nice URLs
- Promotes easy testing
- Templates are closest to Rails of the 3 Java frameworks

Differs from *ails

- Does not provide utilities to generate scaffolding
- No functionality to facilitate Hibernate persistence layer

Components

- Model: Hibernate
  - Developers don't think ActiveRecord can be duplicated in Java
  - Already comfortable with Hibernate
- View: Viento
  - Custom template engine
  - Supports partials and caching
  - Mixins
- Controller: Rigging
  - Custom dependency injection library
  - Provides convention over configuration defaults

Convention over Configuration

- Controllers all go in a specific package
- Action URL contains the controller name, action name, and action parameters
  - ‘widget/list’ => WidgetController.list()
- Views all go under the /views webapp directory
- View names match the controller/action names
  - /views/widget/list.vto
- Template engine extensions follow similar pattern
  - View tools are in org.opensails.examples.tools
  - Mixins are in org.opensails.examples.mixins

Generate Sample Application

- Download zip file from opensails.org
- Create Eclipse project
  - Import Existing Projects into Workspace
  - Select Archive file (downloaded zip file)
- Configure Server
  - Run as Java Application
  - Main Class org.opensails.example.JettyBoot

Add New Controller

```
public class PostController
    extends BaseController {
public void list() {
  expose("posts", postService.getAllPosts());
}
public void view(int postId) {
  expose("post", postService.getPost(postId);
}
public void add() {
  exposeModel("post", new Post());
}
public void save(Post post) {
  // persist post
}
```

Maps to /post/* urls

Extends BaseController

Maps to /post/list
Exposes 'posts' to view

Maps to /post/view/#id
Exposes 'post' to view

Exposes the Post model for a form to use

Post is loaded from the form

List Posts View (list.vto)

```
<body>
  ...
  <table>
    <tr>
      <th>Date</th>
      <th>Title</th>
    </tr>
    $posts.each(cur_post) [[
      <tr>
        <td>$cur_post.dateString</td>
        <td>
        <a href="/app/post/view/$cur_post.id">$cur_post.title</a>
        </td>
      </tr>
    ]]
  </table>
  <a href="/app/post/add">New Post</a>
  ...
</body>
```

Ruby like each construct

Bean style attribute access

Add Post view (/post/add.vto)

```
<html>
  <head><title>Add Post</title></head>
  <body>
    $form.start
     $form.text('post.title').label("Title")<br />
     $form.textarea('post.body').label("Body")<br />
     $form.submit("Post Entry").action(save, [$post])
    $form.end
    </body>
</html>
```

Maps to
PostController.save(post)

Viento: Top Level Mixins

In Java:

```
public class Mixin {
  public boolean isEven (int i) {
    return (i % 2 == 0);
  }
}
...
binding.mixin(new Mixin());
```

In Viento:

```
$isEven($row_num)
```

JavaOne

Viento: Type Mixins

In Java:

```
public class EvenMixin {
  public boolean isEven (int i) {
    return (i % 2 == 0);
  }
}
...
binding.mixin(int.class, new EventMixin());
```

In Viento:

```
$row_num.isEven
```

Viento: Method Missing

In Java:

```
public class TagTool implements MethodMissing {
  public String methodMissing(String methodName,
                              Object[] args) {
    return “<” + methodName + “>”;
  }
}
...
binding.put(“tag”, new TagTool());
```

In Viento:

```
$tag.div
```

Viento: Custom Method Names

In Java:

```
public class Tool {
  @Name(“?”)
  public String question(String arg) {
    return “do something interesting”;
  }
```

In Viento:

```
$tool.?(“my string”)
```

Roadmap

- Project is dormant
- Development team is now using Rails!
- Lead developer was very helpful, and would like to see Sails continue

Agenda

Brief History of Web Development

Ruby On Rails

Sails

Trails

Grails

The Future of *ails

TRAILS

- Started in mid 2005
- Currently at 1.0-SNAPSHOT
- Influences
  - Ruby on Rails
  - Naked Objects pattern

Rails Influence

- Rapid web application development
- Scaffolding generation
- Convention over configuration

Naked Objects Influence

- http://nakedobjects.org
- Domain Driven Design
- Domain objects are behaviorally complete
- Domain objects have single point of definition

Components

- Tapestry
- Spring
- Hibernate
- Maven

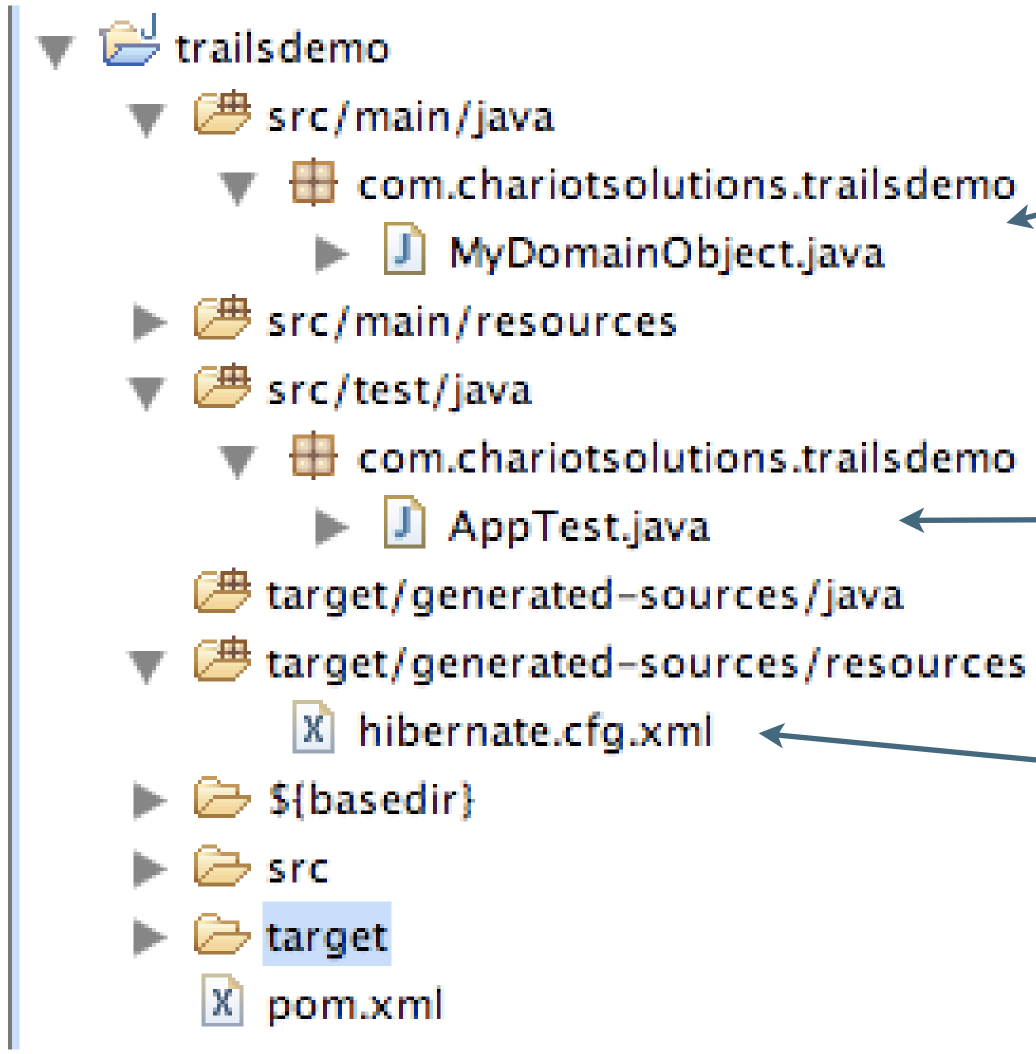
Getting Started

- Requirements
  - Java 1.5
  - Maven 2
- trails-archetype
  - 1.0-SNAPSHOT: build locally
  - Release will be in maven repository

Creating the Application

```
mvn -U archetype:create \
 -DarchetypeGroupId=org.trailsframework \
 -DarchetypeArtifactId=trails-archetype \
 -DremoteRepositories= \
http://snapshots.repository.codehaus.org/ \
 -DarchetypeVersion=1.0-SNAPSHOT \
 -DgroupId=com.chariotsolutions.trailsdemo \
 -DartifactId=trailsdemo
```

What this generates



```
▼ trailsdemo
    ▼ src/main/java
        ▼ com.chariotsolutions.trailsdemo
            ▶ MyDomainObject.java
    ▶ src/main/resources
    ▼ src/test/java
        ▼ com.chariotsolutions.trailsdemo
            ▶ AppTest.java
      target/generated-sources/java
    ▼ target/generated-sources/resources
        hibernate.cfg.xml
    ▶ ${basedir}
    ▶ src
    ▶ target
      pom.xml
```

Source structure created, and includes base domain object

JUnit application test

Hibernate configuration (set for HSQL)

Running the Application

`mvn tomcat:run` or `mvn jetty:run`

- Create process generates a base domain object
- Initially uses an in-memory HSQL database

IDE Support

- Because Trails is built on popular Java libraries, there is already pretty good support in the popular IDEs
  - mvn eclipse
  - mvn idea
  - Netbeans mevenide?

Create Company domain class

```
@Entity
@ValidateUniqueness(property="name")
public class Company {
    private int id;
    private String name;
    private String website;

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    public int getId() ...

    @PropertyDescriptor(index=0)
    public getName() ...

    @PropertyDescriptor(index=1)
    public String getWebsite() ...

      // omitted setters
}
```

Define as an entity

Force name to be unique

Define Primary Key and generation method

Set screen display order

JavaOne

Create Speaker domain class

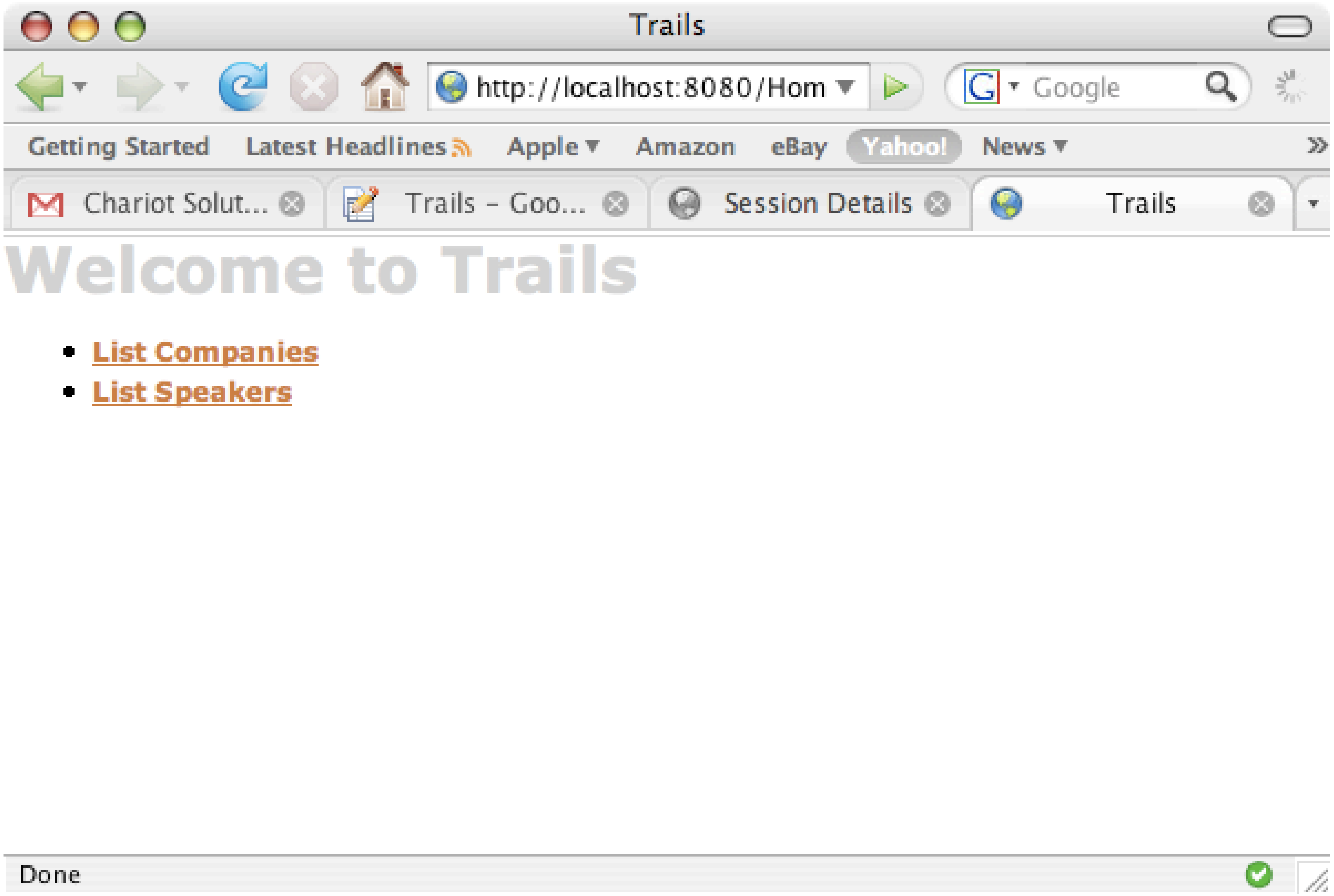
```
@Entity
public class Speaker {
    private int id;
    private String name;
    private Date presentationDate;
    private Company employer;

    @ManyToOne
    @JoinColumn(name="company_id")
    @PropertyDescriptor(index=3)
    public Company getEmployer() ...
}
```

Define many to one relationship between speaker and company

Ready to Go!

`mvn tomcat:run` or `mvn jetty:run`

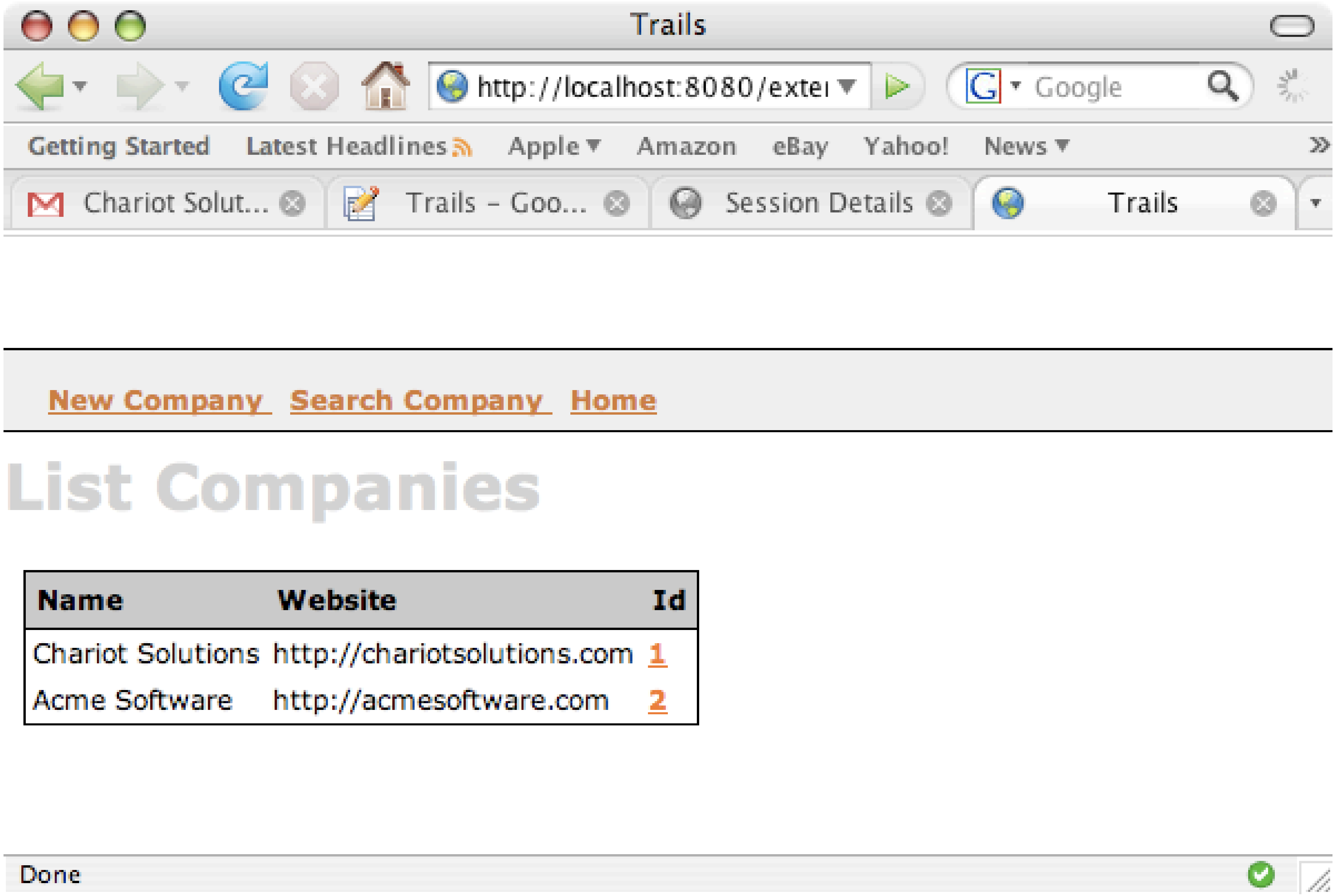
Home page



Welcome to Trails

- List Companies
- List Speakers

List Companies



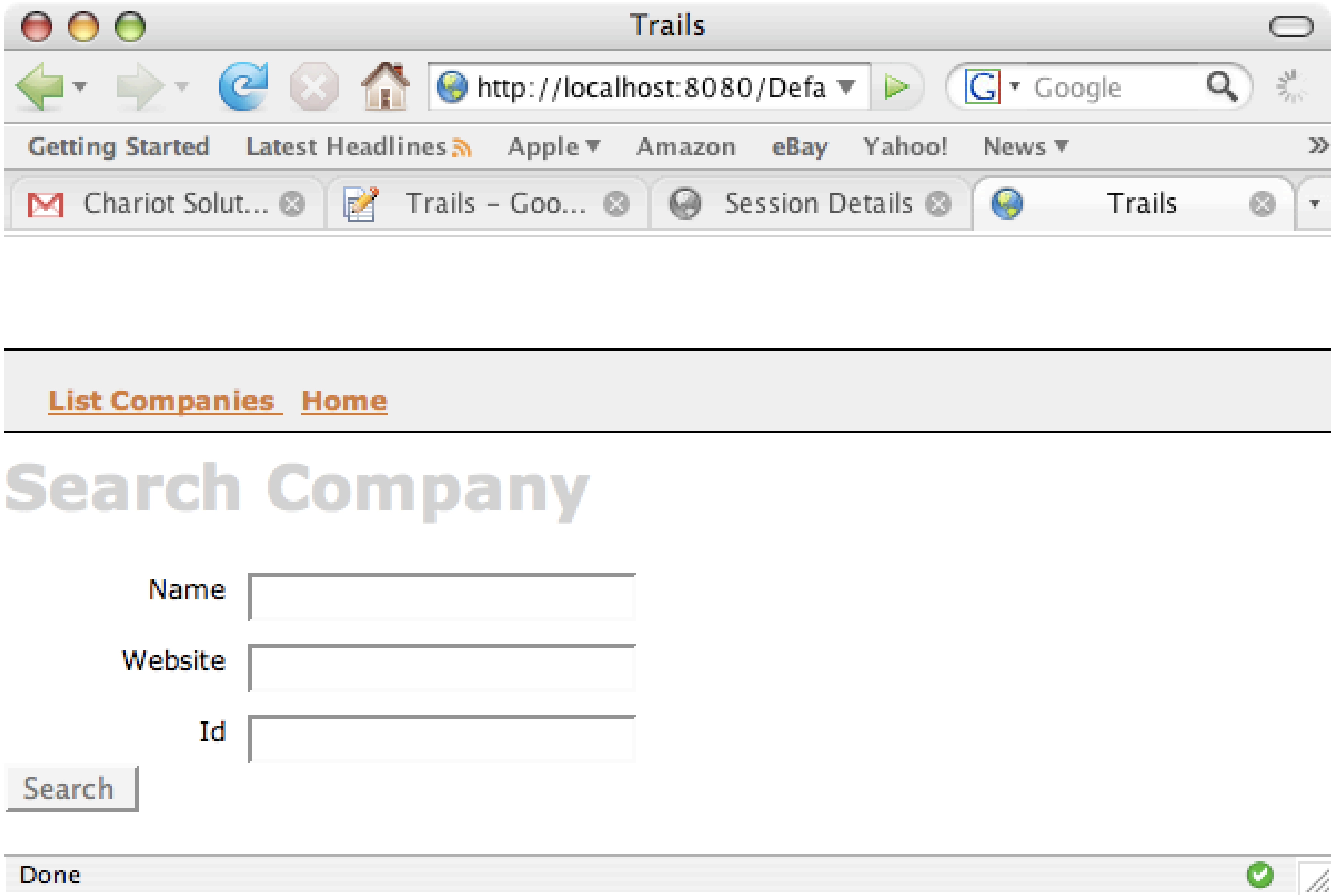
New Company Search Company Home

List Companies

Name	Website	Id
Chariot Solutions	http://chariotsolutions.com	1
Acme Software	http://acmesoftware.com	2

Done

Search Company



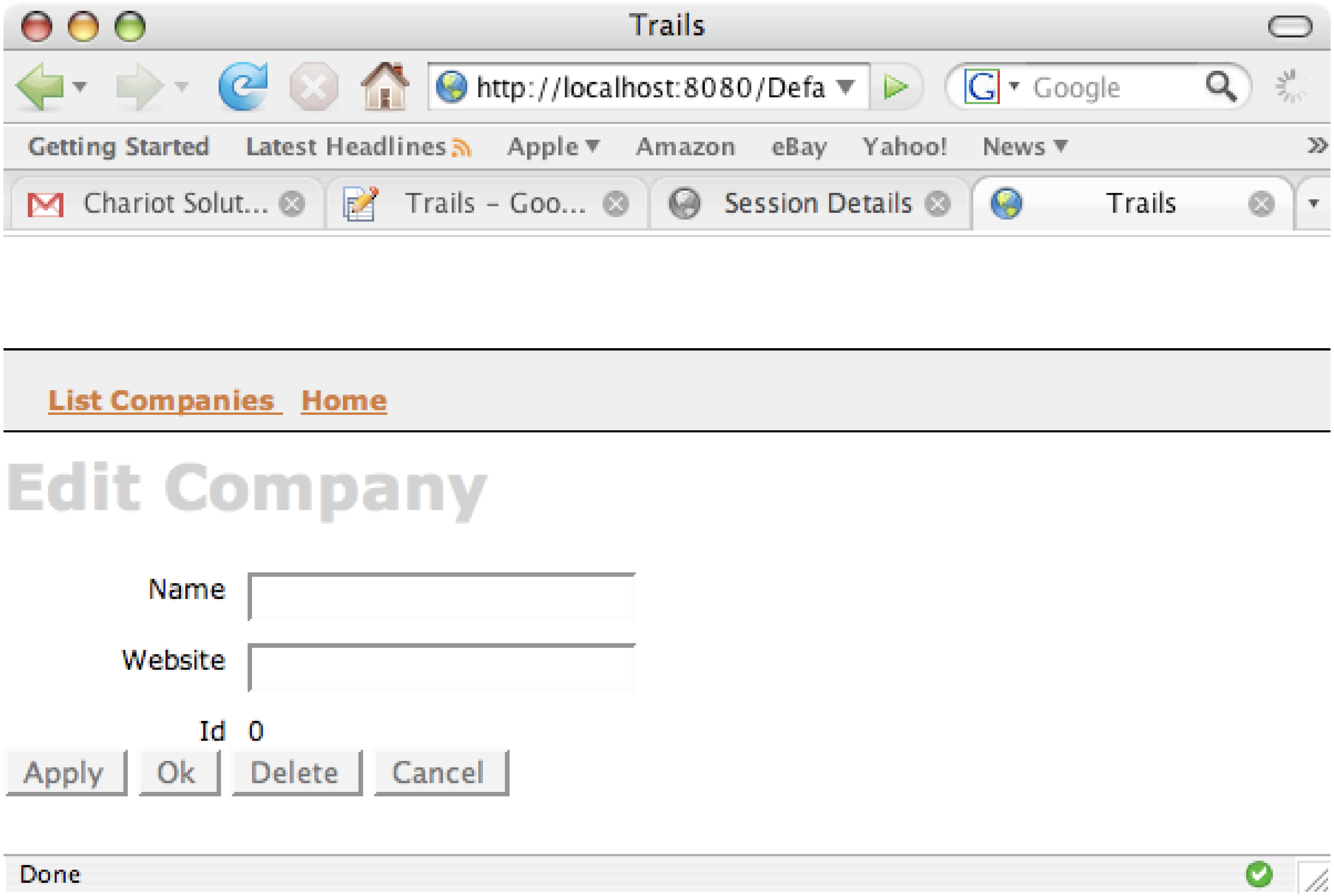
List Companies Home

Search Company

Name	
Website	
Id	

Search

Add/Edit/Delete Company

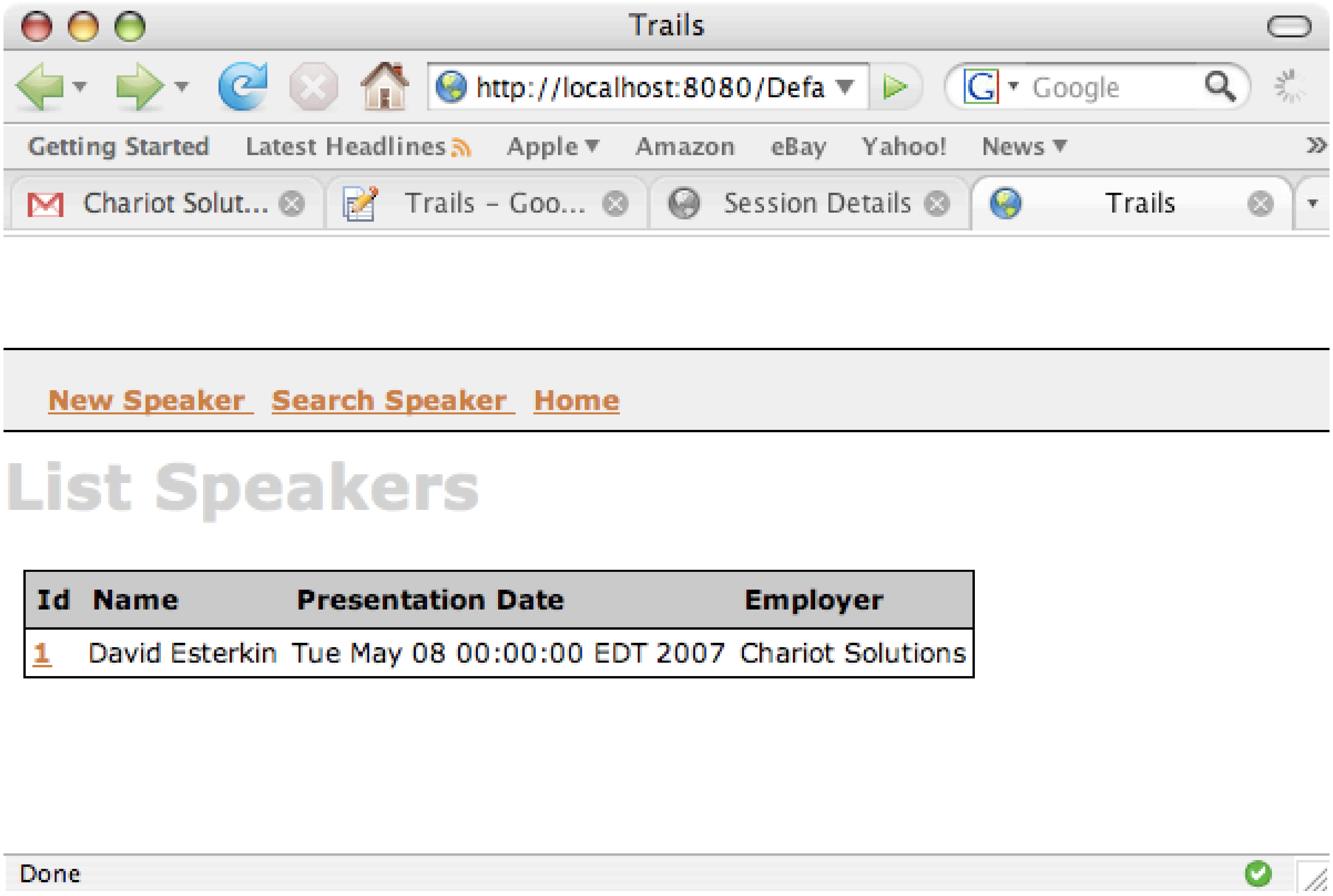


Edit Company

Name	
Website	
Id	0

JavaOne

List Speakers

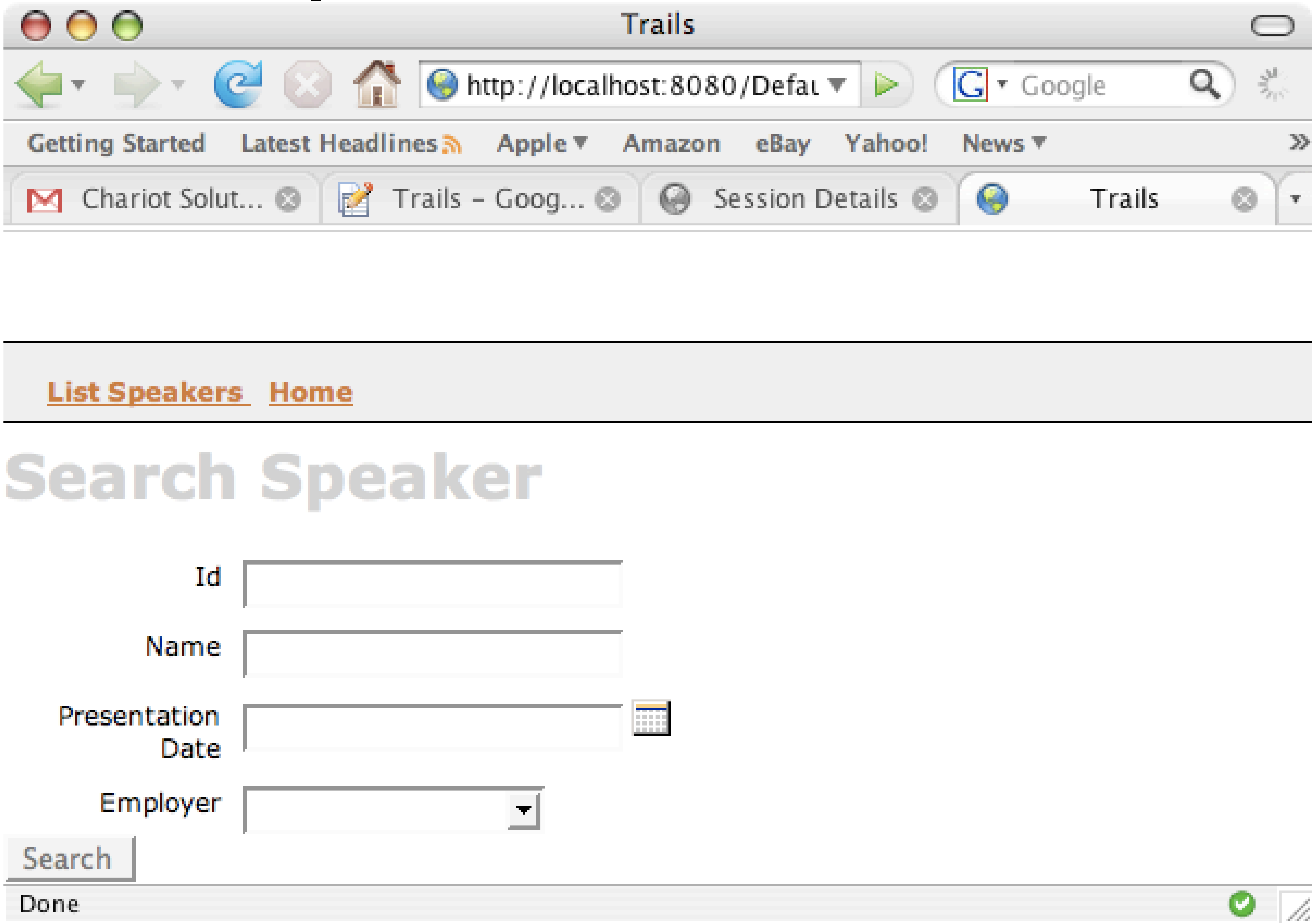


New Speaker Search Speaker Home

List Speakers

Id	Name	Presentation Date	Employer
1	David Esterkin	Tue May 08 00:00:00 EDT 2007	Chariot Solutions

Search Speaker



List Speakers Home

Search Speaker

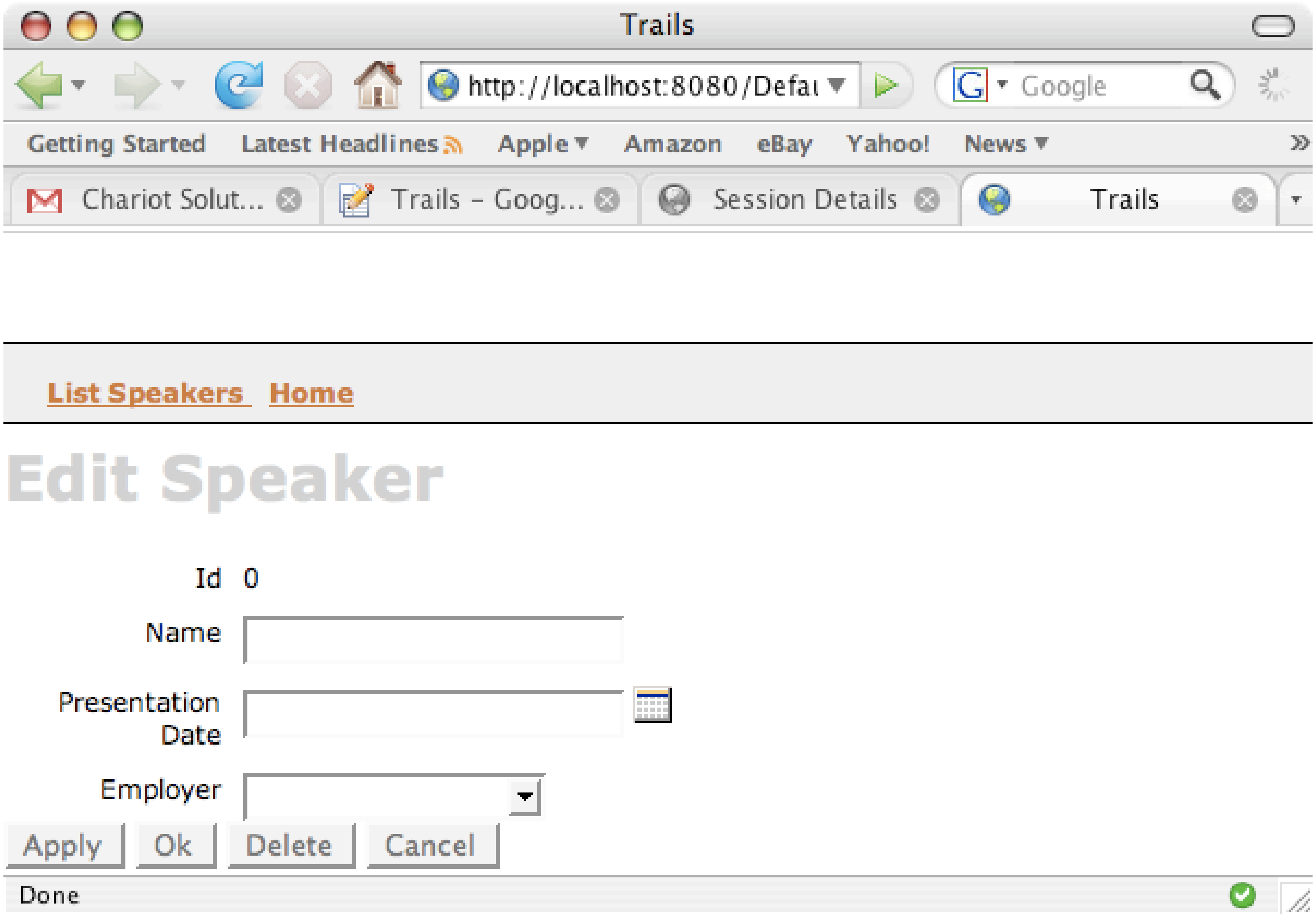
Id	
Name	
Presentation Date	
Employer	

Search

Done

JavaOne

Add/Edit/Delete Speaker



List Speakers Home

Edit Speaker

Id	0
Name	
Presentation Date	
Employer	

Apply Ok Delete Cancel

Done

Customizing View

- Copy the default view to view specific to the controller.
  - cp DefaultEdit SpeakerEdit
- Modify like any other Tapestry template

Roadmap

- Release version 1.0
- Search refactoring and Lucene integration
- equals() Aspect

Agenda

Brief History of Web Development

Ruby On Rails

Sails

Trails

Grails

The Future of *ails

- Open-source web framework started in early 2006
  - Most heavily influenced by Rails
- Built with top of Groovy
  - Dynamic language
  - Can compile down to Java bytecode
  - Interoperability with Java key goal
  - 1.0 released early 2007

First Cousin of Rails

- Takes the most inspiration from Rails
- Design really driven by language
  - Ruby drives Rails
  - Groovy drives Grails

...But not the Weird Cousin

- All the libraries you already know
  - Hibernate 3.2
  - Spring
  - SiteMesh
  - Quartz
- And access to anything else in the Java world
- Calls into Java natural

What’s the Same?

- Project quickstart / artifact generation
- MVC
- Convention over Configuration
- Dynamic finder methods
- Interactive console
- Support for development/production mode

What’s Different Philosophically?

- Domain Driven Development
  - No class to inherit from
  - Class properties drive DB, not the other way around
- Embrace Legacy
  - Support for more complex relationships with Hibernate
  - Middlegen support in the works
- Go Beyond Crud
  - Grails Services
- Half in Groovy, Half in Java

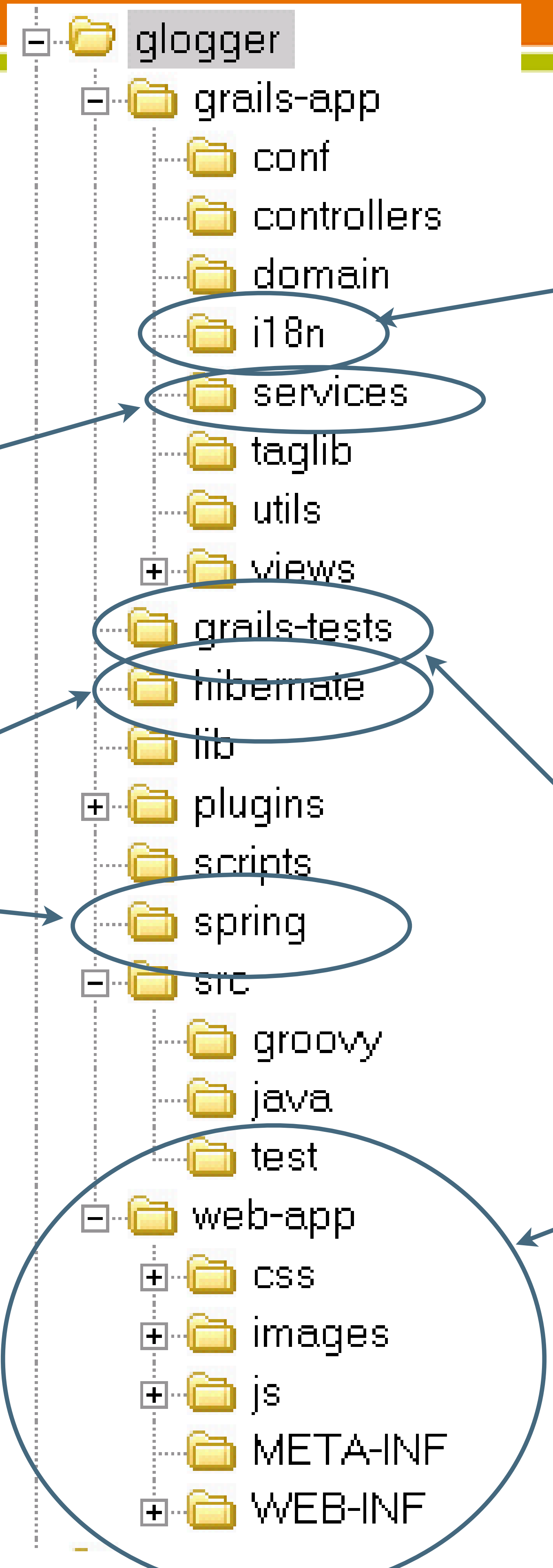
What’s Different Technically?

- Performance
  - Uses native threads
  - Runs on JVM
- Deployment
  - Deploys as a war, hence any servlet container including app servers
- These are arguably the motivations behind JRuby

Up and Running

```
grails create-app glogger
```

Gives you...



```
glogger
  grails-app
    conf
    controllers
    domain
    i18n
    services
    taglib
    utils
    views
  grails-tests
  hibernate
  lib
  plugins
  scripts
  spring
  src
    groovy
    java
    test
  web-app
    css
    images
    js
    META-INF
    WEB-INF
```

Built-in support for internationalization

Support for transactional services

Go under the covers when you need to

Promotes TDD

Your J2EE webapp

What Else Can It Do?

```
create-controller
create-domain-class
create-job
create-plugin
create-script
create-service
create-tag-lib
create-test-suite
```

```
create-webtest
generate-all
generate-controller
generate-views
generate-webtest
install-plugin
install-templates
run-app
run-webtest
shell
```

Dissecting the Domain

```
grails create-domain-class Post
```

$$\downarrow$$

```
grogger\grails-app\domain\Post.groovy
grogger\grails-tests\PostTests.groovy
```

Further Dissecting the Domain

```
class Post {
  String title
  String body
  String author
  String tags
  Date datePosted

  static hasMany = [comments:Comment]
  static constraints = {
    title(unique:true, length:0..150)
    body(blank:false, maxSize:5000)
    datePosted(nullable:false)
  }
}
```

No super class!

Simple properties automatically mapped

Easy definition of relationships

Powerful constraints

Generating the Rest

```
grails generate-all Post
```

$\downarrow$

```
grails-app\controllers\PostController.groovy
grails-app\views\post\list.gsp
......................show.gsp
......................edit.gsp
......................create.gsp
```

Groovy Views (GSP)

- Groovy Server Pages
- Creation of custom tags couldn’t be easier
  - No TLDs
  - Changes are seen instantly
- Discourages scripting
- Ships with large and growing tag library
  - Includes tags for AJAX

Controllers - Generated

```
class PostController {
  def index = {
    redirect(action:list,params:params)
  }

  def allowedMethods = [delete:'POST',
                        save:'POST',
                        update:'POST']
  def list = { ... }
  def show = { ... }
  def delete = { ... }
  ...
  ...
```

Controllers - Dynamic

```
class PostController {
  def scaffold = true
}
```

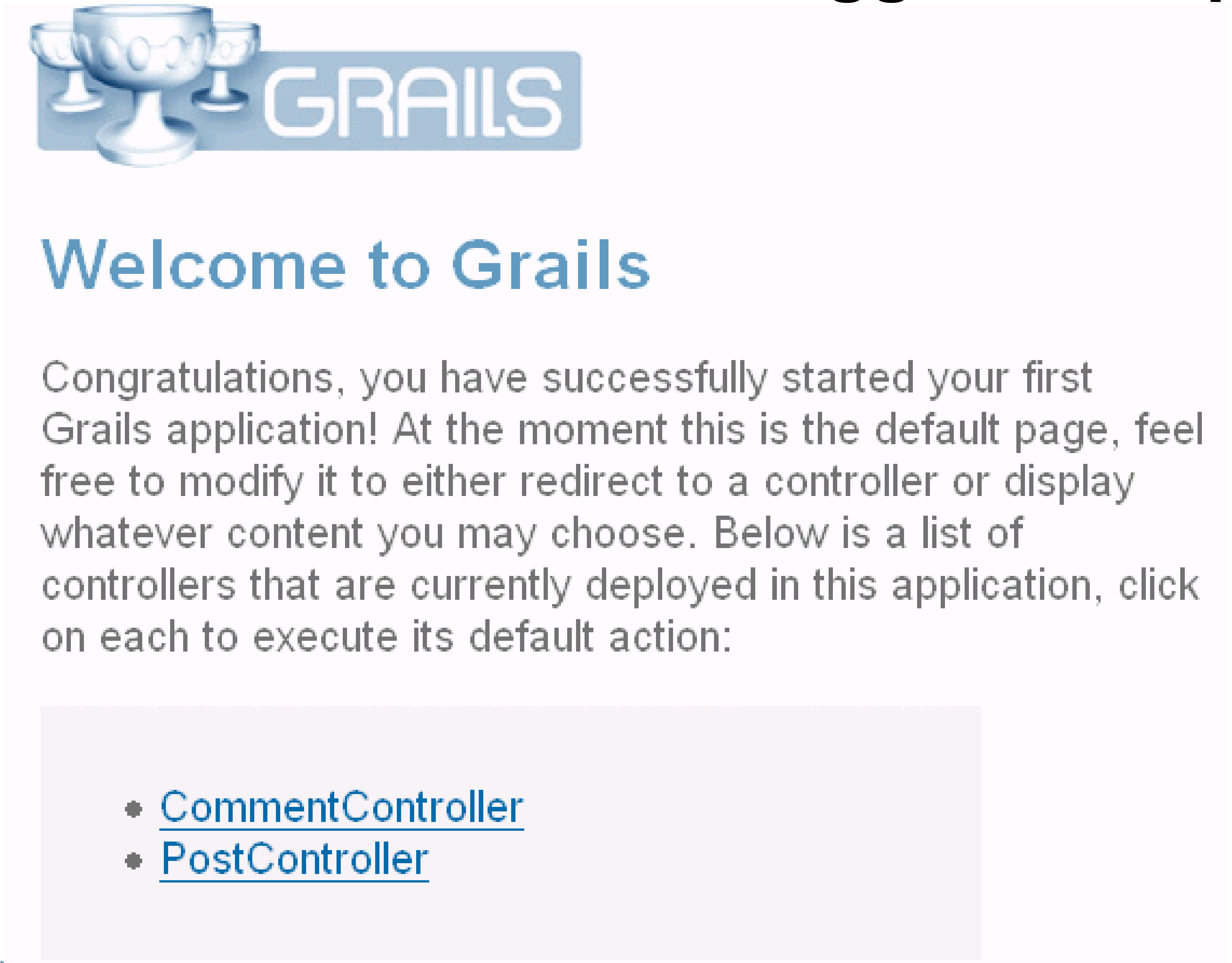
Controllers - Dynamic Override

```
class PostController {
  def scaffold = true

  def list = {
    if(!params.max)params.max = 10
    [ postList: Post.list( params ).reverse() ]
  }
}
```

Implement methods to override the default

Let’s See the App

Glogger Homepage

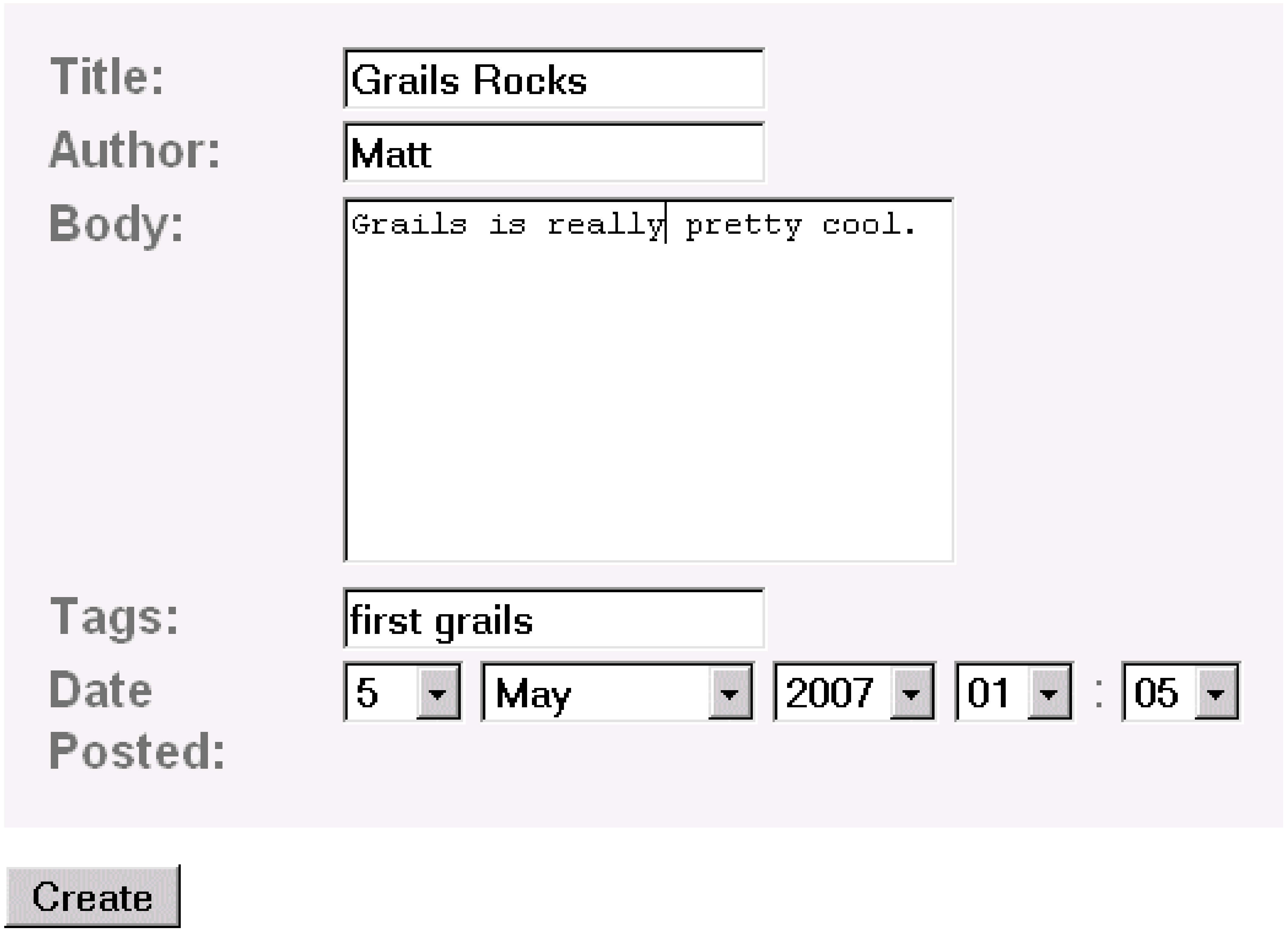


Welcome to Grails

Congratulations, you have successfully started your first Grails application! At the moment this is the default page, feel free to modify it to either redirect to a controller or display whatever content you may choose. Below is a list of controllers that are currently deployed in this application, click on each to execute its default action:

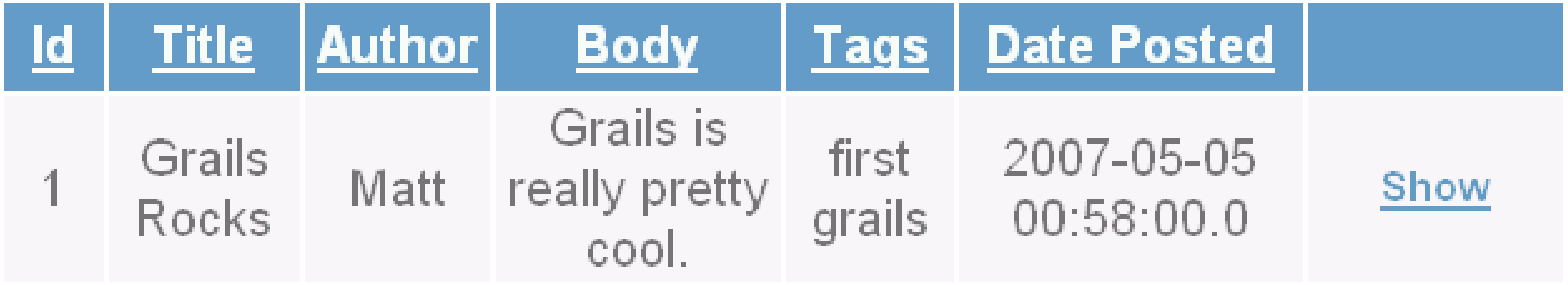
- CommentController
- PostController

Create Post



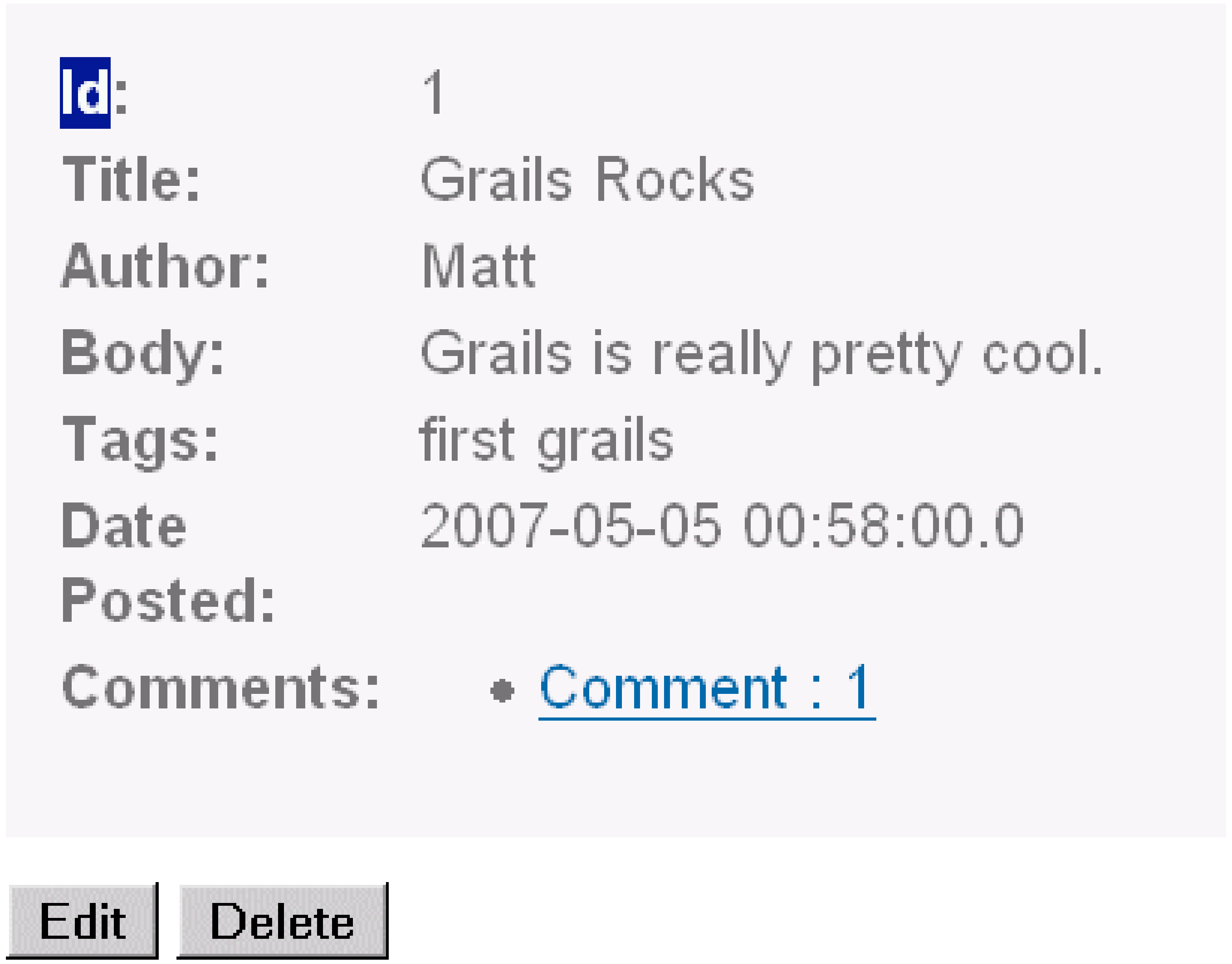
Title:	Grails Rocks
Author:	Matt
Body:	Grails is really pretty cool.
Tags:	first grails
Date Posted:	5 May 2007 01 : 05

List Posts



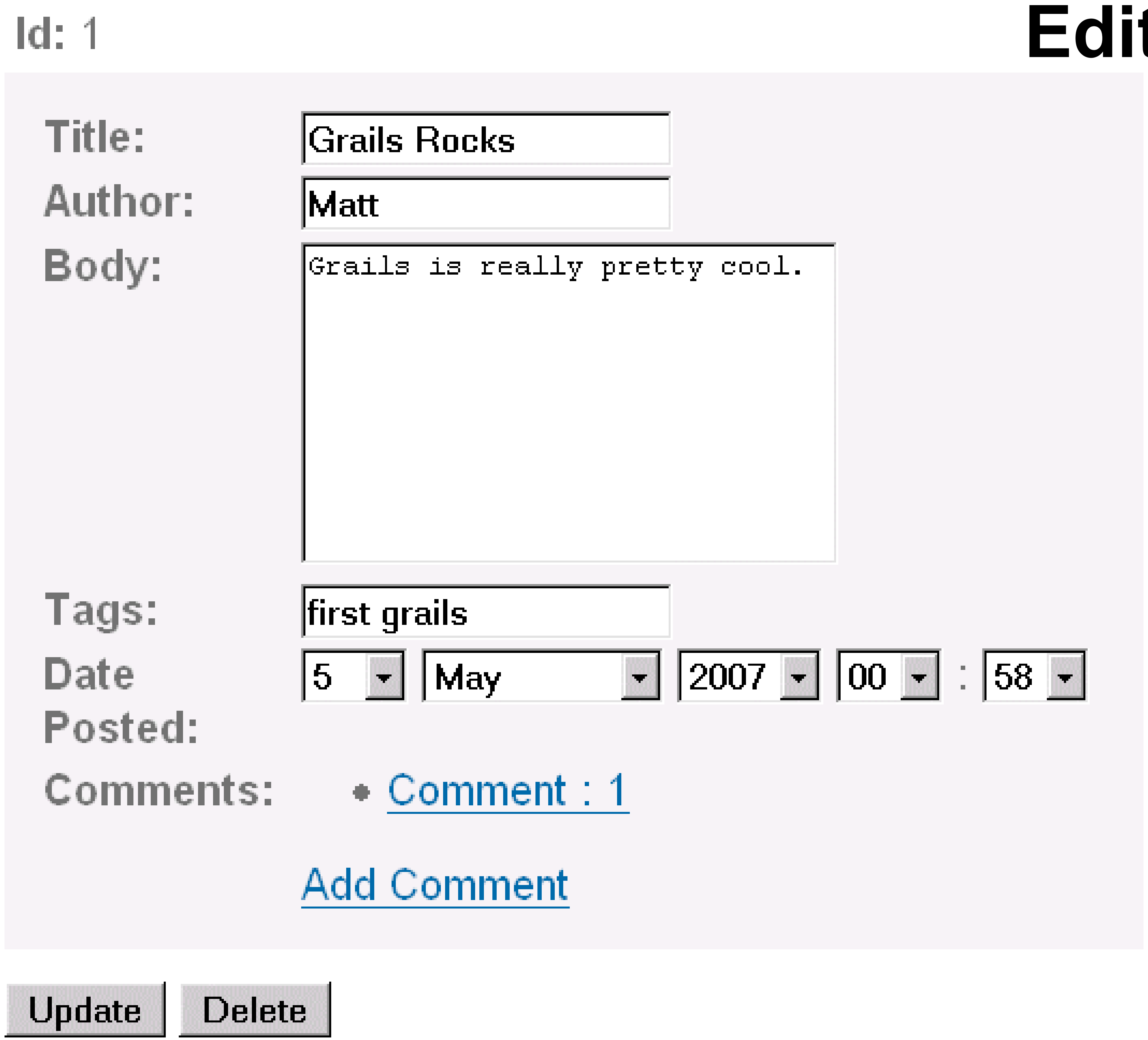
Id	Title	Author	Body	Tags	Date Posted	
1	Grails Rocks	Matt	Grails is really pretty cool.	first grails	2007-05-05 00:58:00.0	Show

View Post



Id:	1
Title:	Grails Rocks
Author:	Matt
Body:	Grails is really pretty cool.
Tags:	first grails
Date Posted:	2007-05-05 00:58:00.0
Comments:	• Comment : 1

Edit Post



Id: 1

Title:	Grails Rocks
Author:	Matt
Body:	Grails is really pretty cool.
Tags:	first grails
Date Posted:	5 May 2007 00 : 58
Comments:	• Comment : 1

Add Comment

Dynamic Methods and Properties

```
Post.findByAuthor("Matt")
Post.findByTitleAndAuthor("Grails", "Matt")
Post.findAll()
Post.listOrderTitle()
Post.hasErrors()
Post.save()
```

Services

- Keeping business logic in the right place

```
class PostService {
    boolean transactional = false
}
```

- Dependency Inject by Convention (Autowiring)

```
class PostService {
  CommentService commentService
}
```

Builders - Query Criteria

```
def c = Post.createCriteria()
def results = c {
  like("title", "%grails%")
  and {
    eq("author", "Matt")
  }
  maxResults(10)
  order("title", "desc")
}
```

Builders - Configuration

```
def bb = new grails.spring.BeanBuilder()
bb.beans {
  dataSource(BasicDataSource) {
    driverClassName = "org.hsqldb.jdbcDriver"
    url = "jdbc:hsqldb:mem:grailsDB"
    username = "sa"
    password = ""
  }
  sessionFactory(ConfigurableLocalSessionFactoryBean) {
    dataSource = dataSource
  }
}
```

Builders - XML Generation

```
<blog>
  <post title="Grails Rocks" author="Matt">
    <body>
      Grails has some real potential
    </body>
    <comment author="anonymous">
      Yeah right.
    </comment>
  </post>
</blog>
```

Grails Roadmap

- 1.0 now targeted for autumn 2007
- Performance and stability are key
- Middlegen support
- JPA support
- JavaScript templates

Agenda

Brief History of Web Development

Ruby On Rails

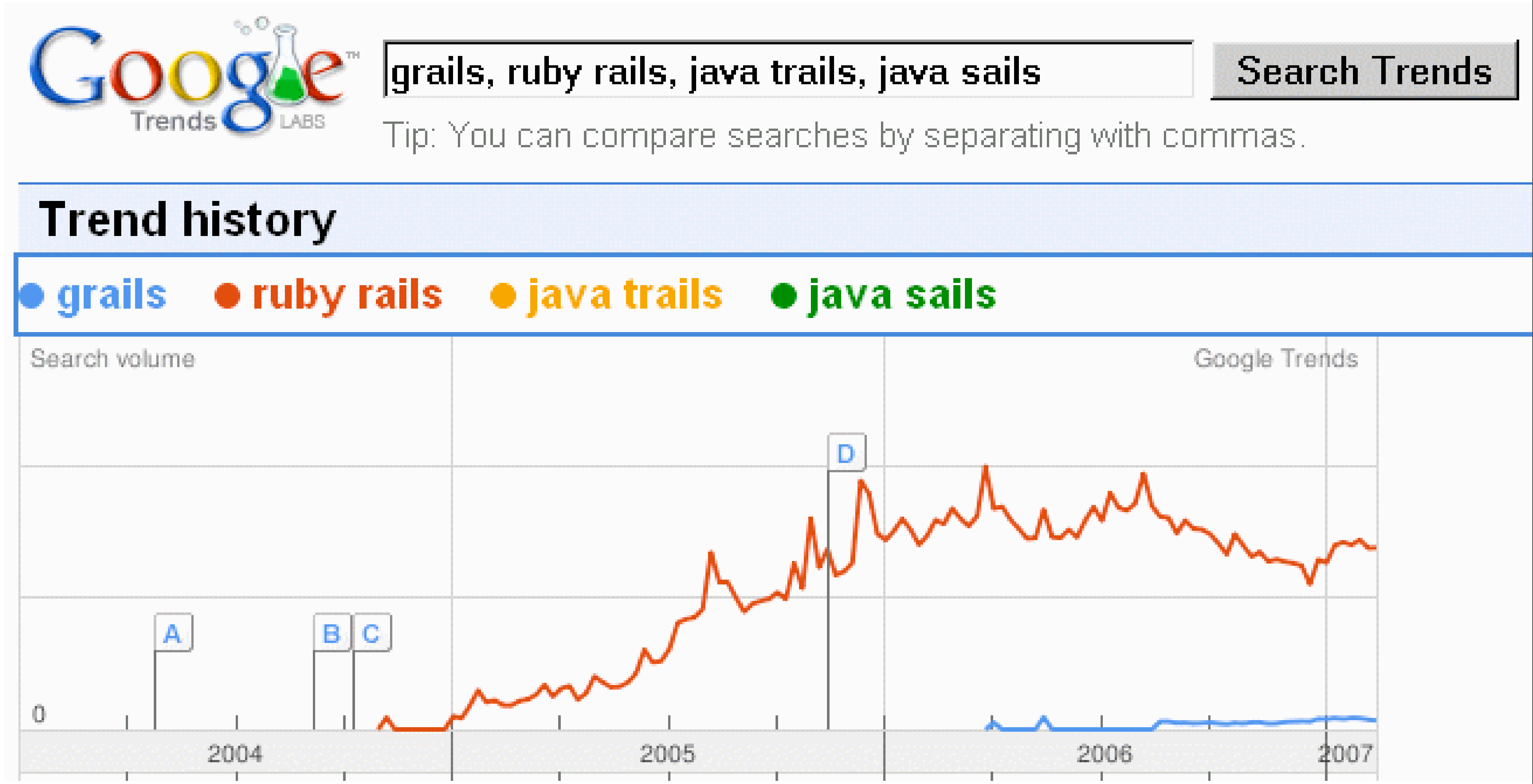
Sails

Trails

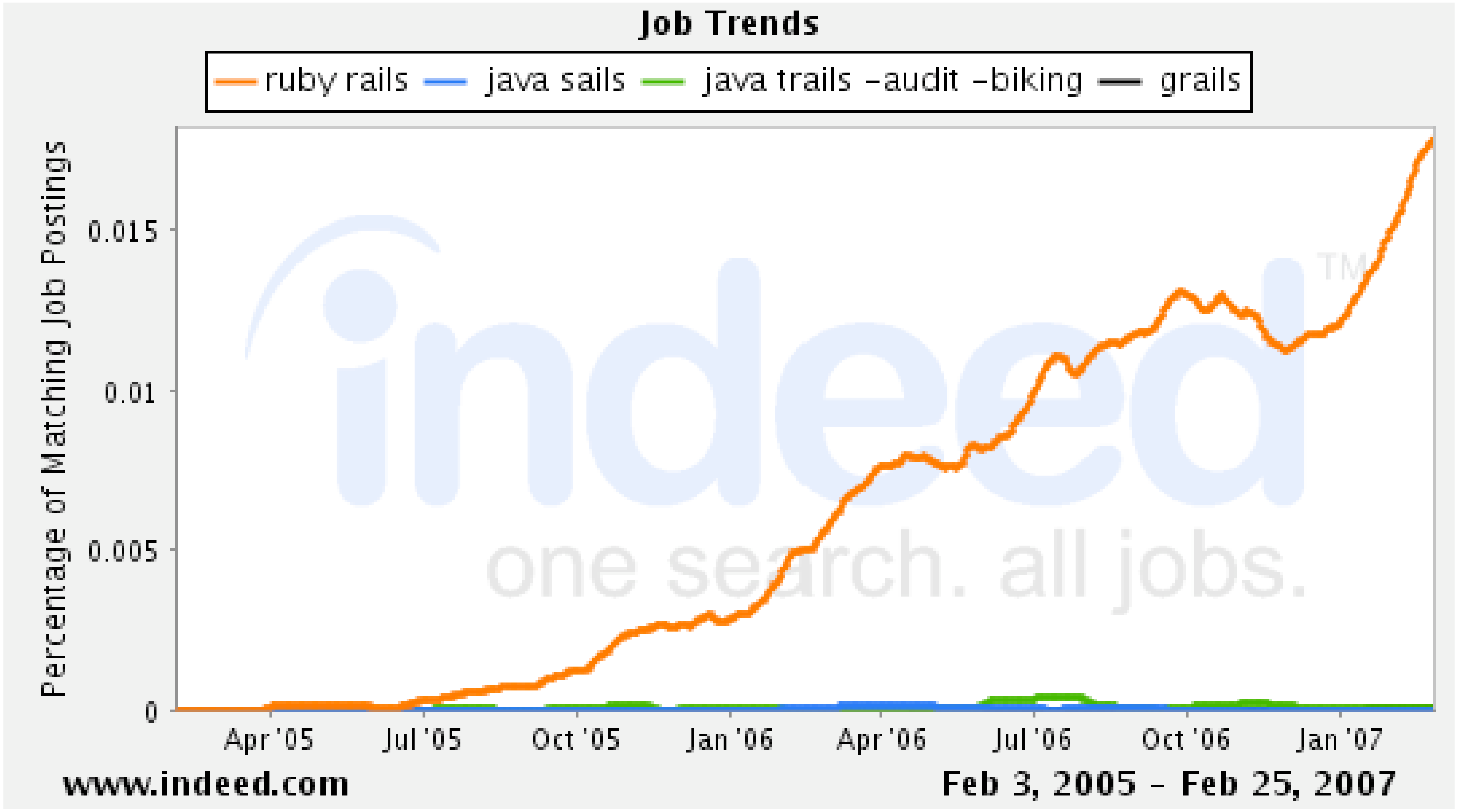
Grails

The Future of *ails

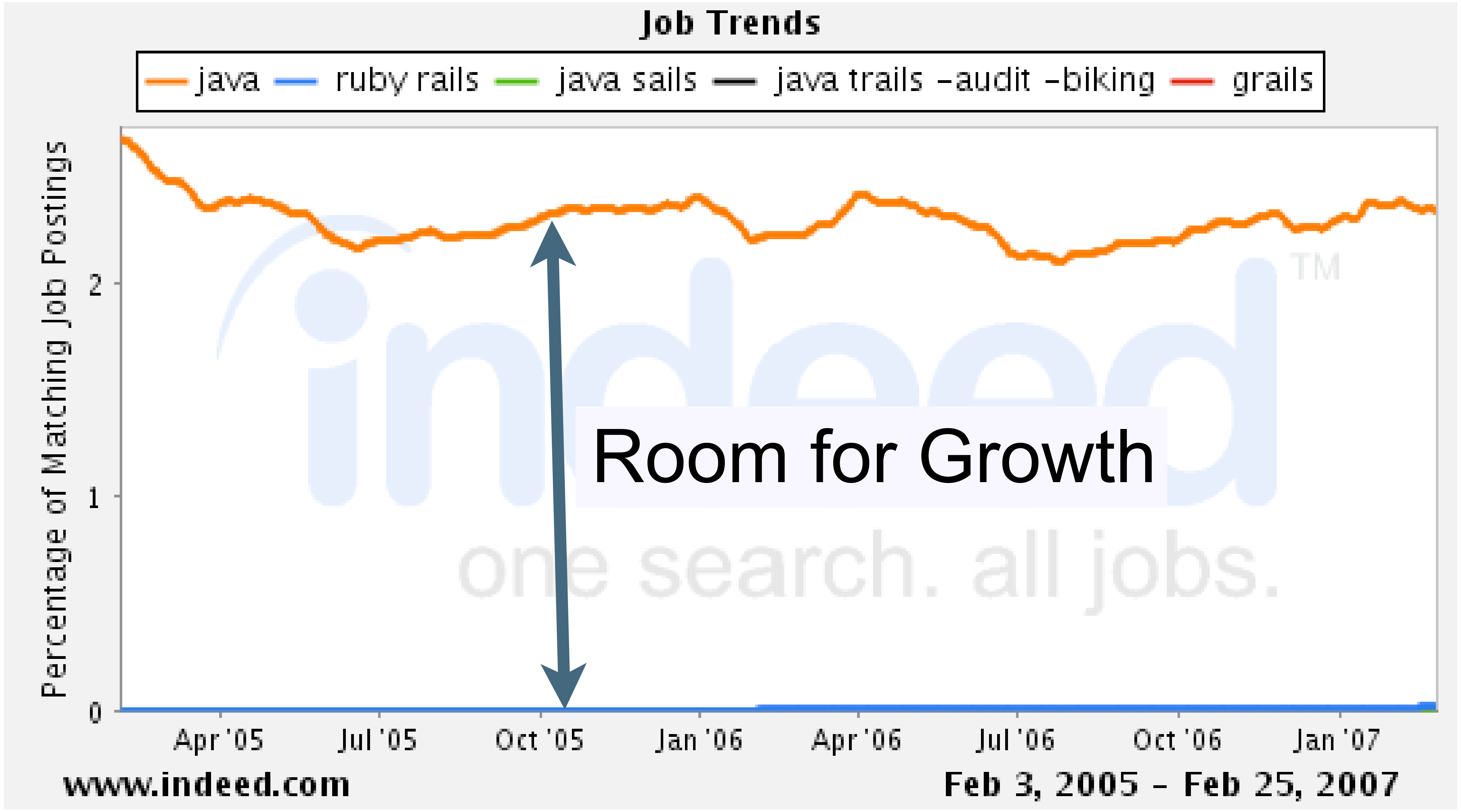
Popularity


Google Trends LABS
grails, ruby rails, java trails, java sails
Search Trends
Tip: You can compare searches by separating with commas.
Trend history
● grails ● ruby rails ● java trails ● java sails
Search volume
Google Trends
A
B
C
D
0
2004
2005
2006
2007

Jobs


Job Trends
ruby rails
java sails
java trails -audit -biking
grails
Percentage of Matching Job Postings
0.015
0.01
0.005
0
Apr '05
Jul '05
Oct '05
Jan '06
Apr '06
Jul '06
Oct '06
Jan '07
www.indeed.com
Feb 3, 2005 - Feb 25, 2007

But...


Job Trends
java
ruby rails
java sails
java trails -audit -biking
grails
Percentage of Matching Job Postings
0
1
2
Apr '05
Jul '05
Oct '05
Jan '06
Apr '06
Jul '06
Oct '06
Jan '07
www.indeed.com
Feb 3, 2005 - Feb 25, 2007
Room for Growth
indeed
one search. all jobs.

Why Aren’t *ails More Popular?

- Haven’t reached critical 1.0 milestone
- Do Trails/Sails solve enough pain points?
- JRuby
    - Are Java developers holding out for JRuby on Rails?
- Inertia?
    - Rails already has huge community, documentation, training, etc

Q&A