

How to create a RIA as a POJD

or how to use GWT, Grails and Cloud-Hosting to create the next killer web-app
as a Plain Old Java Developer

Matthias Lübken - <http://luebken.com> / <http://wishzettel.com>

With the Web 2.0 hype all kinds of languages and frameworks pop up. But as a Java developer it is still a challenge to pick the right tools and get a Rich Internet Application [1] up and running without spending too much time and money. In this experience report I would like to share my experiences on this task.

The example is a web application I've been thinking about for a while: <http://wishzettel.com> [2]. The basic idea is to enable collaborative granting of wishes. If a big celebration from one of my friends is coming up we usually get a big present. While preparing for the event there are basically two questions that have to be answered: First: What do we get? And second: Who wants to participate with what amount? Wishzettel helps answering these questions.

Where do I start?

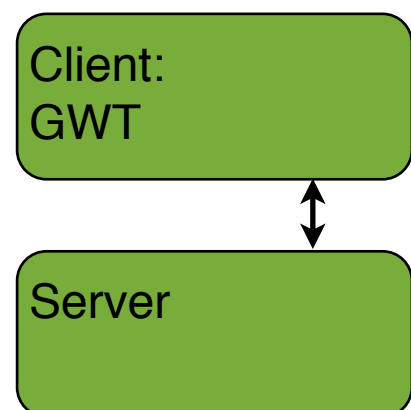
Obviously there are a 1000 ways to create web applications, but not that many for a Plain Old Java Developer (POJD). So with POJD I am actually referring myself. I am fluent in Java and although I wanted to use modern frameworks, I didn't want to learn a whole new language. Don't get me wrong here: I do think that you should learn a new language if it helps you with your problem (my current list is Objective-C and Scala ;-). But I had a minimum amount of time to work on this project and I wanted something up and running fast. And I wanted to see how far I can get with my Java knowledge. Said that, I kind of did learn a new language: Groovy. But I think the learning curve as a Java developer is pretty flat.

The term RIA is used in many ways so let me clarify: I wanted to write a "rich" web client. Just some Javascript / CSS tricks didn't do it for me. I wanted to build an application that actually had real logic on the client. And the application had to run on every browser without the need to install further plugins.

How did I start? (on the client)

Actually I was just fooling around with GWT and it was so easy for me to get started I just gave it a try. GWT [3] is a framework to write browser based applications. The most interesting part of GWT for me was, that you actually write your code in Java and then compile it to JavaScript. The way you program the user interface is actually pretty similar to programming with desktop toolkits like Swing or SWT. You create buttons and labels and place them on panels. You have different types of panels which use different layout strategies. On your webpage you can create areas where you can insert the panels. The reaction to user actions is again straight forward: just register an EventListener and implement the appropriate method like `onClick()`. GWT

doesn't want to hide the browser environment. It integrates seamlessly with JavaScript and



most of the UI tweaks are done in CSS. GWT also gives you a way to communicate to your server back-end through RPC. If you want to know more about GWT go to <http://code.google.com/gwt> and checkout the great documentation.

A typical GWT-Example:

```
public class AView extends Composite implements...
private RadioButton addWishesEveryoneButton =
    new RadioButton("addwishes", "Everyone can add wishes");
private RadioButton editWishesEveryoneButton =
    new RadioButton("editwishes", "Everyone can edit wishes");
private Button saveButton = new Button("Save changes");
private FlexTable table;

public AView(WishzettelModell modell, Controller controller) {
    modell.addModelChangeListener(this);
    wishzettelController.register... Button);
}

private Panel buildMainPanel() {
    DecoratorPanel decoratorPanel = new DecoratorPanel();
    table = new FlexTable();
    table.setWidget(0, 1, addWishesEveryoneButton);
    table.setWidget(2, 1, editWishesEveryoneButton);
}
```

Implement Patterns however you like.

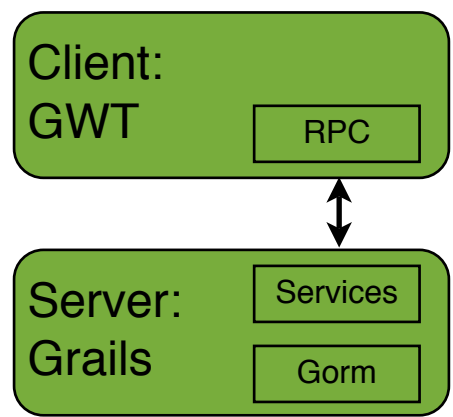
Just create Buttons and other Tables like in other UI - Frameworks

Register Listeners just like in Swing.

How did I continue? (on the server)

Although I wanted to do most of the application logic on the client, an RIA without an back-end isn't really fun. My requirements to the server side were basic: I had a couple of connected domain objects that had to be stored and loaded. And I needed some minor logic like generating keys and I needed to write E-mails. Fortunately I did some work with Grails [4] just a couple of months ago. So for me that was the naturally choice and I would suggest it to experienced Java developers facing similar problems. Grails was originally bringing the idea of Ruby on Rails to the Groovy / Java world. Today Grails brings some unique features to the table and is ready for prime time.

Grails is based on Groovy [5]. Groovy is a dynamically typed and very expressive scripting language running on the JVM. It is highly integrated into the JVM and makes it painless using existing Java libraries. Java developers will appreciate the familiar syntax. The web framework Grails embraces convention over configuration which makes your code much more concise. For example to make sure a domain object with a couple of fields is saved and retrieved to the database is a matter of creating a Groovy class in a predefined folder,



adding your fields to the class and calling the methods `save()` and `findBy()`. Grails uses standard technologies like Hibernate and Spring as its foundation and provides a plugin system that integrate additional features. For me the GWT-Plugin [6] was particularly interesting because it makes integrating GWT with Grails much easier. Especially the communication between GWT and Grails via RPC is supported.

A Grails Gorm example:

```
class Wish {
  Long id
  String name
  String suggestedBy
  String url
  String imageUrl
  int estimatedCost
}
```

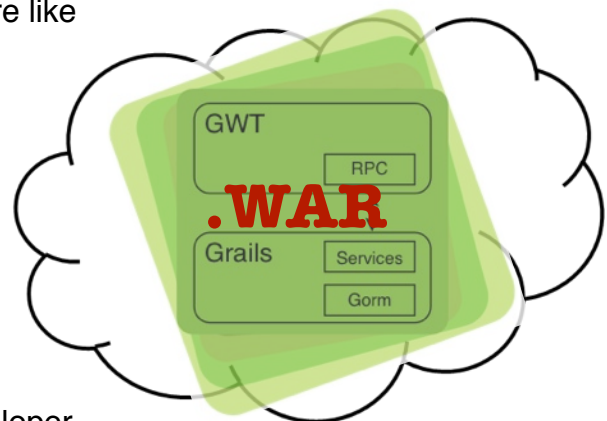
A typical Grails-Gorm model.

Just declare fields you need. All the database mapping is done for you.

How did I finish? (deploying my application)

This was holding me off for a while. Until about a year ago it was hard to get a Java web application up and running in the wild. I've never quite understood the reasons but it seems Java hosting is not the easiest thing. So if you wanted to deploy your Java application you had two choices: a) Go to a premium Java hoster and pay a fairly high price for your small application or b) get a dedicated or shared server and install and maintain your own web server. Both of these options didn't do it for me. I wanted to focus on my application.

Fortunately there are some new options out there like cloud based hosting services. The first cloud hosting service that I got aware of which is capable of hosting Java applications is Morph [7]. They provide hosting for different application environments and are based on the Amazon EC2 [8]. The great thing for an RIA developer is that you can start of fairly low and while your user base hopefully grows you can add new computing power. The other great thing about Morph for me was, that it was totally painless to deploy. After I've created a free developer account, all I had to do is to configure the database connection. For the deployment you have to use a special Jar which uploads your War. So for me releasing a new version is a matter of calling "grails war" and running a small deploy script.



Sum up



If you look around the RIA world you see many solutions. Most of these solutions are not intuitive for a Java developer but fortunately the landscape is changing. With GWT and Grails you can create fast and rich internet applications. And with new hosting solutions like Morph you can get your application into the wild fast and cheap.

I've did it with <http://wishzettel.com>. So where is your next killer web-app?

by Matthias Lübken - <http://luebken.com> / 11. December 2008

- [1]: http://en.wikipedia.org/wiki/Rich_Internet_application
- [2]: <http://wishzettel.com>: Right now we are on a beta test. Feedback appreciated.
- [3]: <http://code.google.com/webtoolkit/>
- [4]: <http://www.grails.org>
- [5]: <http://www.groovy.org>
- [6]: <http://grails.org/GWT+Plugin>
- [7]: <http://mor.ph>
- [8]: <http://aws.amazon.com/ec2>