

## **Interview with JavaFX student developer Joan Barrull**

[Student Views and Reviews](#) recently spoke with Joan Barrull, a law student with a background in software engineering. Joan has won several programming-related awards including the Mobius prize for a multimedia game in addition to a prize in the [JavaFX Blogging Contest](#). Joan currently works as a software engineer at [Baratz](#) and is working on developing a new generation of “documental” software. A shorter version can be found [here](#).

**Student Views and Reviews:** How long have you been using JavaFX?

**Joan Barrull:** I've been using JavaFX for some months. I started learning JavaFX 1.1 and then updated to the latest version of JavaFX 1.2. My company decided to write a little application in order to report employee expenses. We managed to write the application very fast, so the management staff decided to go on with JavaFX. Right now, we are developing several projects with JavaFX. Maybe in the future my company will use JavaFX more extensively so if you are a programmer willing to work with this technology don't forget to [visit our Web site](#) for job offers.

**SVR:** What resources did you use to learn JavaFX?

**JB:** In order to learn JavaFX, SUN provides a lot resources. To name some of them:

- <http://javafx.com/learn/> (plenty of samples and tutorials)
- SUN JavaFX forums (useful for technical questions )
- The online course: <http://www.javapassion.com/javafx/> This is a 15-week online course. In one word: Excellent.
- And don't forget the online blogs where JavaFX developers write about their journey with JavaFX

**SVR:** What new features do you find the most helpful in the new JavaFX 1.2 release?

**JB:** Ease of use. To write a Swing app takes a lot of technical effort. But with JavaFX you can write an initial prototype in just minutes. This way you can focus on the users instead of dealing with obscure technical tricks.

Another great feature is that the JavaFX architecture fosters the work flow of a graphical app:

1. Designers and programmers working in parallel.
2. Integration of media art and code.
3. Testing

**SVR:** How do you think JavaFX has changed or will change the way developers create RIAs?

**JB:** Right now a developer has several options in order to write a RIA: Adobe Flash, Microsoft Silverlight and even open-source alternatives like Expression. JavaFX has some advantages over all these alternatives: it is cross-platform, so eventually every computer on the planet can run a JavaFX

app (just install Java 6). Also, remember that JavaFX can be used to write phone apps, where JavaFX can become a very good alternative to more closed environments (iPhone/Objective C).

**SVR:** Where do you use JavaFX in your work and/or daily life?

**JB:** In my company we are using JavaFX for several projects. With JavaFX it is very easy to write prototypes and proof-of-concepts, so we use JavaFX often.

**SVR:** Do you plan to develop JavaFX apps for the Java Store when it is opened to developers outside the U.S.?

**JB:** Sure. Right now I have several applications ready to apply to the JavaStore. I know that opening a JavaStore in Europe takes a lot of work in order to comply with the local law (data protection law, copyright law, and corporate law) but I am eager to visit JavaStore Europe.

**SVR:** What do you like most about JavaFX?

**JB:** The language is just awesome, it has some characteristics – like closures and binding – that other languages don't have. But overall, JavaFX is a great authoring platform for designing graphical apps very quickly. That results in apps that focus on providing a great user experience.

**SVR:** What do you see as the future of JavaFX?

**JB:** In the future I see JavaFX like a great phone apps authoring tool. JavaFX has a great advantage over its competition: it is not a closed platform.

**SVR:** How do you think JavaFX compares to other RIA development programs? In what ways is JavaFX better?

**JB:** Following is a table when you can compare JavaFX against their competitors:

	<b>Adobe Flash</b>	<b>Microsoft SilverLight</b>	<b>Expression</b>	<b>JavaFX</b>
<b>Open platform</b>	No	No	Yes	Yes
<b>Multi-platform</b>	No	No	Yes	Yes
<b>Free</b>	No	No	Yes	Yes
<b>Desktop &amp; Phone authoring</b>	No	No	No	Yes

As you can see JavaFX can compete head to head against the other platforms.

**SVR:** If you could add one feature to JavaFX right now, what would that be?

**JB:** I am against adding features for the sake of it. An authoring platform is not a set of features or widgets. It is more complex than that. Right now, JavaFX has the minimum to prototype and do some basic stuff. It lacks a proper multimedia support (sounds, video). But I am sure this will be solved in the next versions. I prefer that the designers of JavaFX keep focused on architectural questions rather than spending time on widgets. I know a Cuban programmer that worked as a Gosling assistant when Java was born: he explained me that AWT (a graphical development kit) was designed in just two weeks! Obviously the first version of AWT had plenty of errors and now you have Swing plenty of deprecated methods. Moral: expending some time on basic stuff is well worth it.

**SVR:** Where can people follow you online?

**JB:** I write about my experiences on JavaFX on [my blog](#).

As a law student I write about law-related stuff: media licenses, JavaStore, copyright laws and of course there is code and JavaFX apps.