JavaScript/HTML 5 Rich Clients Using Java EE 7

Reza Rahman
Java EE/GlassFish Evangelist
reza.rahman@oracle.com
@reza_rahman
Program Agenda

- JavaScript/HTML5 Rich Client Landscape
- Java EE 7
- Java EE + JavaScript
JavaScript/HTML5 Rich Clients Rising?

- The thin client vs. rich client debate is pretty old…
- Server-side web frameworks have ruled for a while (JSF, Struts, Spring MVC)
- AJAX was a mild shift to the client (PrimeFaces, GWT, Vaadin)
- Rich clients powered by JavaScript/HTML5 appear to be making a comeback…
  - Improving JavaScript engines (V8, *Monkey, Nashorn)
  - Better tools (jQuery, MV* frameworks, Chrome, FireFox, Avatar)
  - Standards advancement (CSS3, HTML5, WebSocket, HTML Components…)
Perhaps not a Slam Dunk?

- Richer clients clearly better at some things
  - Highly dynamic, interactive interfaces
  - Complex, feature-rich UIs
  - “Single page applications” (“Applets” 😊)

- But perhaps not a panacea
  - Heavily form/workflow driven applications
  - Server-side rendering still a better bet for performance/reliability?
  - JavaScript/HTML development is not without its pains…
  - Server-side frameworks are a strong incumbent

- Co-existence in the short and long term?
  - Islands of rich client functionality within server-centric UIs?
  - Different strokes for different folks?
My Big Fat Rich-Client Architecture

- Very similar to client/server architecture of lore
- Client responsible for UI rendering, basic input validation, logic and state
- Server responsible for business logic, domain model, persistence
- Web/HTTP is glue that connects client and server
- Typical communication protocols
  - REST for majority of cases
  - WebSocket when full-duplex communication is needed
  - JavaScript tools support REST well, but not WebSocket (yet)
- The typical (ideal?) data interchange format is JSON
- Java EE is a great server-side platform for this architecture...
Java EE + JavaScript

JavaScript/HTML5

JAX-RS  Java API for WebSocket  Java API for JSON  JAXB

Servlet

EJB 3  CDI

JPA  JMS  JTA  JCA

Bean Validation
JAX-RS

- REST development API for Java
- Server and client
- Annotation based, declarative
  - @Path, @GET, @POST, @PUT, @DELETE, @PathParam,
    @QueryParam, @Produces, @Consumes
- Pluggable and extensible
  - Providers, filters, interceptors
JAX-RS Example

@Path("/atm/{cardId}")
public class AtmService {

    @GET
    @Path("/balance")
    @Produces("text/plain")
    public String balance(
        @PathParam("cardId") String card,
        @QueryParam("pin") String pin) {
        return Double.toString(getBalance(card, pin));
    }

    ...

JAX-RS Example

...  

@POST  
@Path("/withdrawal")  
@Consumes("text/plain")  
@Produces("application/json")  
public Money withdraw(  
    @PathParam("card") String card,  
    @QueryParam("pin") String pin,  
    String amount) {  
    return getMoney(card, pin, amount);  
}  
}
Java API for WebSocket

- High level declarative API for WebSocket
- Both client and server-side
- Small, powerful API
  - @ServerEndpoint, @OnOpen, @OnClose, @OnMessage, @OnError, Session, Remote
- Pluggable and extensible
  - Encoders, decoders, sub-protocols
WebSocket Sample

```java
@ServerEndpoint("/chat")
public class ChatBean {

    Set<Session> peers = Collections.synchronizedSet(...);

    @OnOpen
    public void onOpen(Session peer) {
        peers.add(peer);
    }

    @OnClose
    public void onClose(Session peer) {
        peers.remove(peer);
    }

    ...
```
WebSocket Sample (Continued)

... 

@OnMessage

public void message(String message, Session client) {
    for (Session peer : peers) {
        peer.getRemote().sendObject(message);
    }
}

}
JavaScript Movers and Shakers

AngularJS
Search term

Backbon...
Search term

KnockoutJS
Search term

Ember.js
Search term

+ Add term

Graph showing the trend of search terms over time.
Project Avatar

- End-to-end open source JavaScript framework from Oracle
  - JavaScript on the client and server side (can be just one or the other)
  - Uses Nashorn/JDK8
  - Utilizes some underlying Java EE capabilities
  - Runs on GlassFish, perhaps on WebLogic at some point…

- Integrated support for REST, WebSocket, Server-Sent Events (SSE)…

- PhoneGap support

- Still nascent, good time to get involved

http://avatar.java.net
Java EE + JavaScript Demo

https://github.com/m-reza-rahman/javaee-javascript
Summary

- JavaScript/HTML5 clients gaining traction as opposed to server-side web frameworks
- Communication between the client and server happens via JSON over REST or WebSocket
- Java EE well positioned as a JavaScript rich client backend, especially with JAX-RS, the Java API for WebSocket and JSON-P
- JavaScript framework from Oracle - Avatar
- You can use the demo code as a starting point to exploring the emerging space
- Most importantly, have fun!
Try it Out!

http://download.java.net/glassfish/4.0/release/glassfish-4.0.zip
Resources

- Java EE Tutorials
  - http://docs.oracle.com/javaee/7/tutorial/doc/home.htm
- Digging Deeper
  - http://docs.oracle.com/javaee/7/firstcup/doc/home.htm
  - https://glassfish.java.net/hol/
  - https://java.net/projects/cargotracker/
- Java EE 7 Transparent Expert Groups
  - http://javaee-spec.java.net
- Java EE 7 Reference Implementation
  - http://glassfish.org
- The Aquarium
  - http://blogs.oracle.com/theaquarium