



Criando uma aplicação HTML5 com Java EE 7 e WebSockets

Bruno Borges

Principal Product Manager

Java Evangelist



Bruno Borges



- Java Evangelist
- Oracle Product Manager
- Entusiasta JavaFX e IoT
- Onde me encontrar
 - @brunoborges
 - plus.google.com/+BrunoBorges



Ascensão do Javascript

Cientes ricos HTML5

- O debate cliente ‘leve’ vs cliente ‘pesado’ é antigo...
- Frameworks web *server-side* mandaram por um tempo (Struts, Spring MVC, JSF)
- Ajax foi uma mudança suave para o *client-side* (GWT, Vaadin)
- Rich clients ~~estão voltando~~ voltaram, graças ao Javascript/HTML5
 - Motores Javascript melhoraram muito (V8, SpiderMonkey, Nashorn)
 - Melhores ferramentas (jQuery, MV* frameworks, Chrome, Firefox, JavaFX)
 - Melhores padrões (CSS3, HTML5, WebSockets, Web Components)

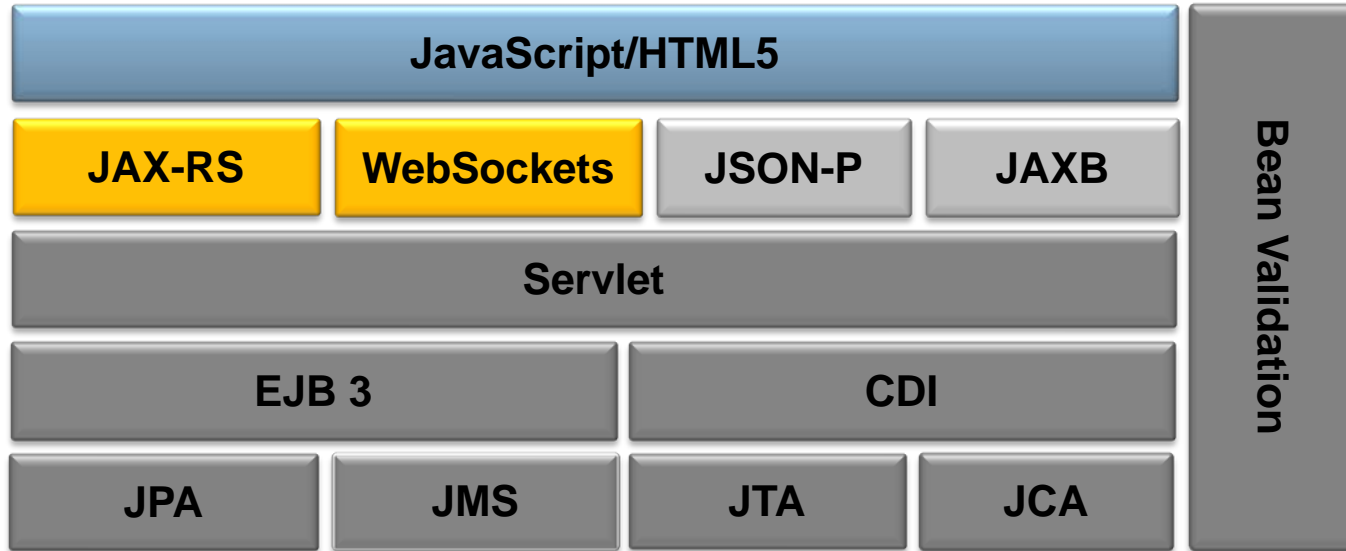


Arquitetura Rich Client

- Similar a arquiteturas cliente/servidor
 - Cliente responsável pela UI, input, validacao, lógica e estado
 - Server responsável pela lógica de negócio, modelo de domínio, persistência
 - Web/HTTP é a cola que conecta client e server
- Protocolos de comunicacao comuns
 - REST na maioria dos casos
 - WebSocket quando precisa de comunicacao full-duplex
 - Ferramentas Javascript suportam REST muito bem, mas ainda não WebSocket
- O formato comum (talvez ideal?) de troca de dados é JSON
- Java EE é uma ótima plataforma server-side para esta arquitetura

Java EE + Javascript

Plataforma server-side para suportar apps ricas HTML5/Javascript



JAX-RS

- API de desenvolvimento REST para Java
- Servidor e Client
- Orientada a Annotations, declarativa
 - @Path, @GET, @POST, @PUT, @DELETE, @PathParam, @QueryParam, @Produces, @Consumes
- Plugável e extensível
 - Providers, filters, interceptors

JAX-RS

Exemplo 1

```
@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

Exemplo 2

```
@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 WebSockets JSR 356



Java API for WebSockets

JSR 356

- Protocolo HTTP é half-duplex
- Gambiarras
 - Polling
 - Long polling
 - Streaming
- WebSocket resolve o problema de uma vez por todas
 - Full-duplex

Java API for WebSockets

- API declarativa de alto nivel para WebSockets
- Client e server-side
- Pequena e simples, mas completa
 - @ServerEndpoint, @OnOpen, @OnClose, @OnMessage, @OnError, Session, Remote
- Plugável e extensível
 - Encoders, decoders, sub-protocols

Java API for WebSockets

Exemplo 1

```
@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);
    }
}
```



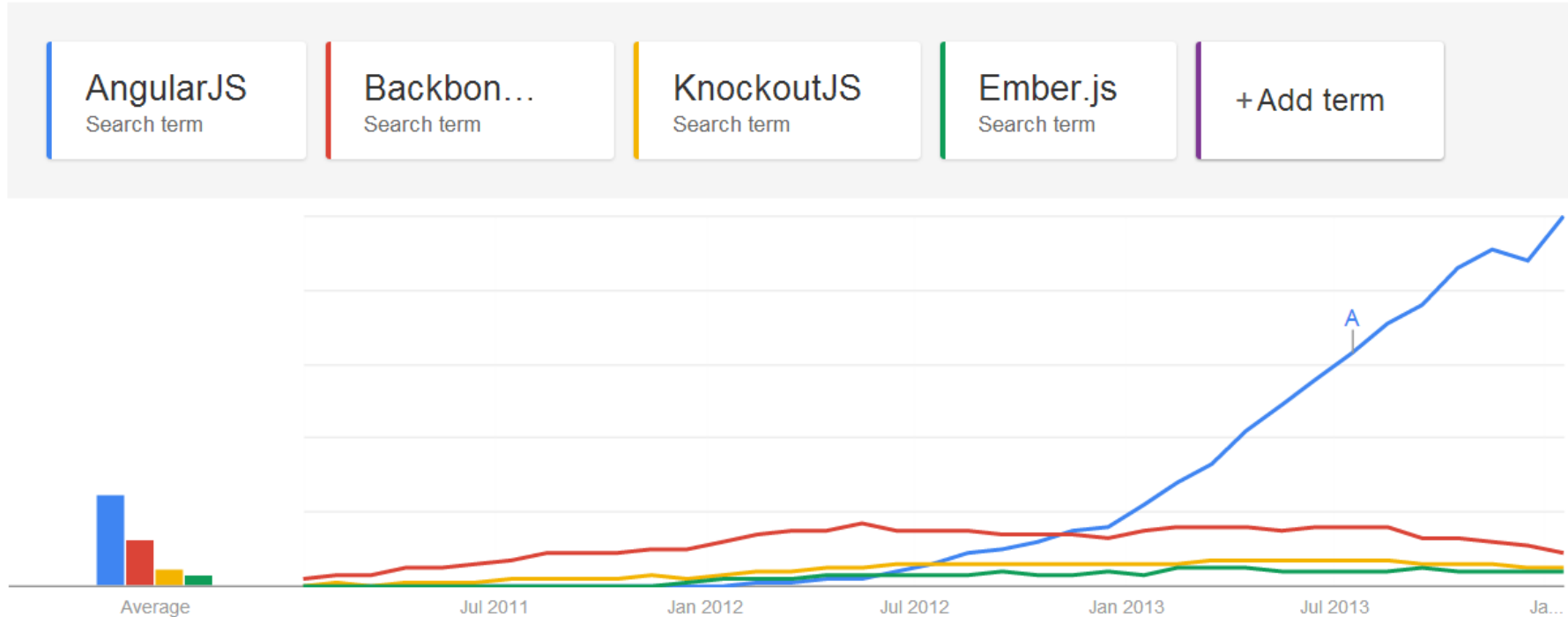
Java API for WebSockets

@OnMessage

```
public void message(String message, Session client) {  
    peers.forEach(  
        p -> p.getRemote().sendObject(message)  
    );  
}
```



Javascript Movers and Shakers



Java™

ORACLE®

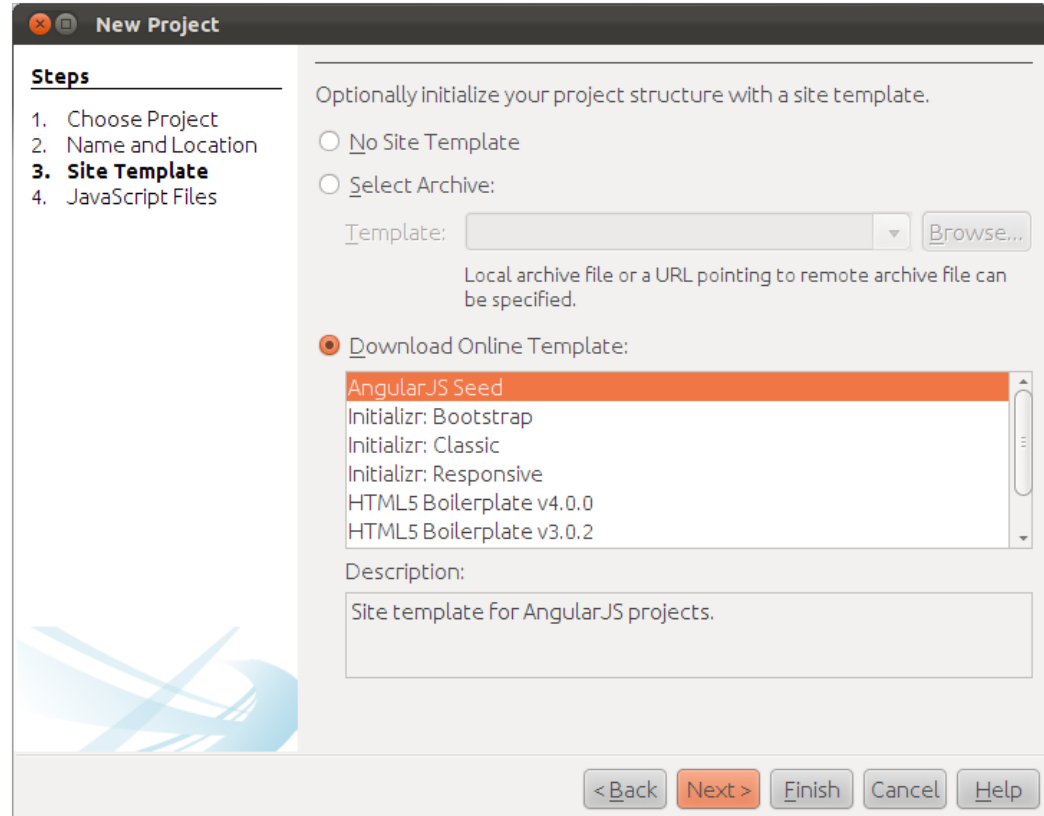
Projeto Avatar

avatar.java.net

- Javascript framework da Oracle
- Javascript no client e no server-side
- Nashorn/JDK8
- Utiliza algumas capacidades do Java EE
- Roda no GlassFish, e em breve no WebLogic também
- Suporte já integrado para REST, WebSocket, Server-Sent Events (SSE)
- Suporte a PhoneGap
- Versão 1.0 já disponível. Bom momento para participar!

Projetos no NetBeans

- Twitter Bootstrap
- HTML5 Boilerplate
- Initializr
- AngularJS
- Mobile Boilerplate



Demo

[.github.com/brunoborges/javaee-javascript](https://github.com/brunoborges/javaee-javascript)



Conclusão

- Cientes JavaScript/HTML5 conquistando desenvolvedores
- Comunicação entre cliente e servidor em JSON via REST ou WebSocket
- Java EE funciona muito bem como um backend para clientes ricos Javascript, especialmente com JAX-RS e a Java API para WebSockets, e JSON-P
- JavaScript framework da Oracle – Projeto Avatar
- Você pode usar o código de demonstração para começar a explorar estas ideias
- E o mais importante, divirta-se programando! :-)

GlassFish 4.0.1

GlassFish Nightly Builds

- glassfish.org
- Java EE 7
- Suporte a JDK 8
- Projeto Avatar



Para saber mais

- **Java EE Tutorials**
 - <http://docs.oracle.com/javaee/7/tutorial/doc/home.htm>
- **Outras páginas para iniciar os estudos**
 - <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>
- **The Aquarium (blog Java EE da Oracle)**
 - <http://blogs.oracle.com/theaquarium>

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

ORACLE®



CREATE THE FUTURE



