

*Perfect World Programming presents*

---

# Spring in a Vert.x World

Mark Spritzler

---

---

# Who am I?

---

- Developed mod-spring-appcontext Vert.x module (really just a jar)
- One man of Perfect World Programming LLC - where I do training and consulting, except for the past year and a half
- Chief Software Architect for Z4 Poker, LLC where we are using Vert.x to build the fastest biggest online Poker application



---

# Topics

---

- Why use Spring within your Vert.x applications
- Why inject Vert.x into your Spring applications. (You got peanut butter in my chocolate)
- `mod-spring-appcontext`
- Demo



---

# Why use Spring within your Vert.x applications

---

- You already have code that you want to re-use in your Vert.x application and that code is already using Spring
- You want to use dependency injection in your Vert.x application
- You want your Vert.x application to be able to send things on the eventBus to your Spring application
- You want to take advantage of the many Spring projects in your Vert.x application



---

# Why inject Vert.x into your Spring applications

---

- You want your Spring application to be able to send things on the eventBus to your Vert.x application

---

# mod-spring-appcontext

---

- ❖ Vert.x module - sort of
  - ❖ Current version works best as a dependency jar
    - ❖ requires a little more work than I would have liked
- ❖ Future version will be a plain module
  - ❖ I wrote the module with earlier versions of Vert.x which had a different classloader design
  - ❖ Will allow us to configure it with config.json



---

# mod-spring-appcontext - how to use

---

- ❖ Use xml or Spring @Configuration class
- ❖ Add the mod-spring-appcontext jar file as a dependency to your project
- ❖ Now in code: Create a JsonObject to store some meta data
  - ❖ “configFiles” property takes a JsonArray of config files
  - ❖ “configType” property says what type of Spring config files are you using xml or @Configuration
- ❖ If you want Vert.x injected into your beans (optional) call  
`SpringApplicationContextHolder.setVertx(vertx);`
- ❖ Then create the ApplicationContext by calling  
`SpringApplicationContextHolder.createApplicationContext(configFiles);`
- ❖ Now get your beans.



# DEMO!!!

<https://github.com/bytor99999/SpringInAVertxWorld>



Thank you very much. Unless we still have time for a different topic? Then lets talk some more.



---

# Because we have extra time

---

- Discussion on Vert.x addressing granularity:
  - Coarse grained. Address per verticle type
  - Fine grained. Addressing via application data
    - address per userID
    - address per tableID
    - address per casino
  - Mid grained. Addressing per action type.
    - address on table sit down
    - address on player betting
    - address on avatar change



---

# Because we have extra time

---

- ❖ Possible Other Topics -
  - ❖ Different types of Verticles, Verticle versus Worker Vertical
  - ❖ Storing state in Verticles
  - ❖ Classloaders



# Evaluate This Session

- 1 Sign-in: [www.eclipsecon.org](http://www.eclipsecon.org)
- 2 Select session from schedule
- 3 Evaluate:   