

Czego się możemy nauczyć w bibliotece

Aleksander Nowiński,
Centrum Otwartej Nauki, ICM,
Uniwersytet Warszawski

Moja perspektywa

- Dużo luźno powiązanych rekordów (10-100M)
- Wyszukiwanie ponad wszystkim
- Dużo prezentacji
- Mało edycji
- Nie lubię komercyjnych (i w ogóle bardzo relacyjnych) baz danych
- Nosql
 - hadoop
 - mongodb
 - neo4j, RDF itd.

Doświadczenia z produkcji oprogramowania

- Izolacja jest fajna
- Warstwy - dane, logika (backend), prezentacja
- Logika "blisko" prezentacji
- Logika jest fajna
- Prezentacja najczęściej robi problemy
- Trzeba dobrze rozumieć framework którego się używa (jak to działa)

Gruby klient - doświadczenia

- Swing/java
- Komunikacja przez interfejs serwisowy
- Możliwość łatwego zbudowania skomplikowanej logiki interfejsu
 - łatwiej
 - mniej ludzi to umie :-)
- Wieczne problemy ze zgodnością z serwerem
- Zgodność platformy - "zamienił stryjek systemy na przeglądarki"

Największe rozczarowania

- **Java Server Faces:**
 - Premie - pozorne
 - Połowa czasu spędzona na obchodzeniu ograniczeń
 - Problemy dla użytkownika
- **Skomplikowane frameworki prezentacyjne**
 - zazwyczaj okazują się niestrawne na dłuższą metę

Prezentacja

- Sensowne zbudowanie jest trudne
- Zawsze kończyliśmy na prostych rozwiązaniach
 - Web flow
 - Spring MVC
- Nigdy nie dopuścić logiki do stron
 - template-y
- Proste komponenty edycyjne - wielokrokowe, zamiast skomplikowanych aplikacji

Spring - kluczowe kawałki

Pisali to ludzie, którzy myślą podobnie jak ja...

- Spring IOC
- Spring MVC
- Spring Remoting
- Spring Security
- Spring batch
- Spring integration
- Spring AOP

Jeśli ci brakuje czegoś - może spring już to ma?

Spring IOC container - zamiast j2ee

- biega niezależnie od serwera
 - od wielkich aplikacji
 - do lekkich klientów
- prosty koncepcyjnie
- bardzo dojrzały
- elastyczny
- bywa kosztowny przy testach (ale i tak się opłaca)

Spring MVC

- Oryginalne JSP mieszają prezentację z logiką
- MVC klei się gładko z większością rzeczy
- Usprawnia architekturę

Spring remoting

- Bardzo, bardzo elastyczne rozwiązanie dla wywołań zdalnych
- Różne nośniki:
 - RMI
 - Http Invoke (hessian) - serializacja po http
 - SOAP
 -
- Gładkie i lekkie rozwiązanie
- Relatywnie łatwa droga od prototypu (definiowanego w javie) do ustalonego protokołu

Spring Security

- Bardzo fajne, spójne rozwiązanie dla klejenia spraw uwierzytelniania i autoryzacji
- Zrobione dla integracji różnych źródeł informacji
- Oryginalny mechanizm zabezpieczeń webappa dość koszmarne

Spring batch

- szybkie i proste operowanie na zadaniach batchowych
- dobre radzenie sobie z błędami
- fajne narzędzie

Spring integration

- Narzędzie do budowania workflowów

Spring AOP

Naprawdę super narzędzie do transakcji...

...Ale poza tym - **samo zło**

Apache SOLR - ultimate search tool

- Działa out of the box
- Integruje się już chyba ze wszystkim (np. hibernate)
- Dobry
- Szybki
- Małe szanse że potrzebujesz czegoś więcej
- Dostarcza praktycznie wszystkich potrzebnych funkcjonalności
- Błyskawicznie się rozwija

Warsztat w CeON ICM

- Continuous Integration - Jenkins
- Sonar - marzenie wrednego szefa
- Test Driven Development - jak się uda ;-)
- Maven - chyba nie ma nic lepszego
- IDE & OS agnostic - nie ma problemu 'u mnie działa'

Na deser

- JVM jest fajny
- Bawmy się - zrobmy czasem moduł w scali, clojure
 - ale izolujmy
 - encapsulujmy

Dziękuję!

Komentarze, pytania