

Autumnなのに Spring

日本Springユーザ会
長谷川裕一



1

はじめに

-
- おおまかなアジェンダ
 - 国内のSpring事情などの紹介
 - Spring2.xの新機能やS2APなどの最新情報



2

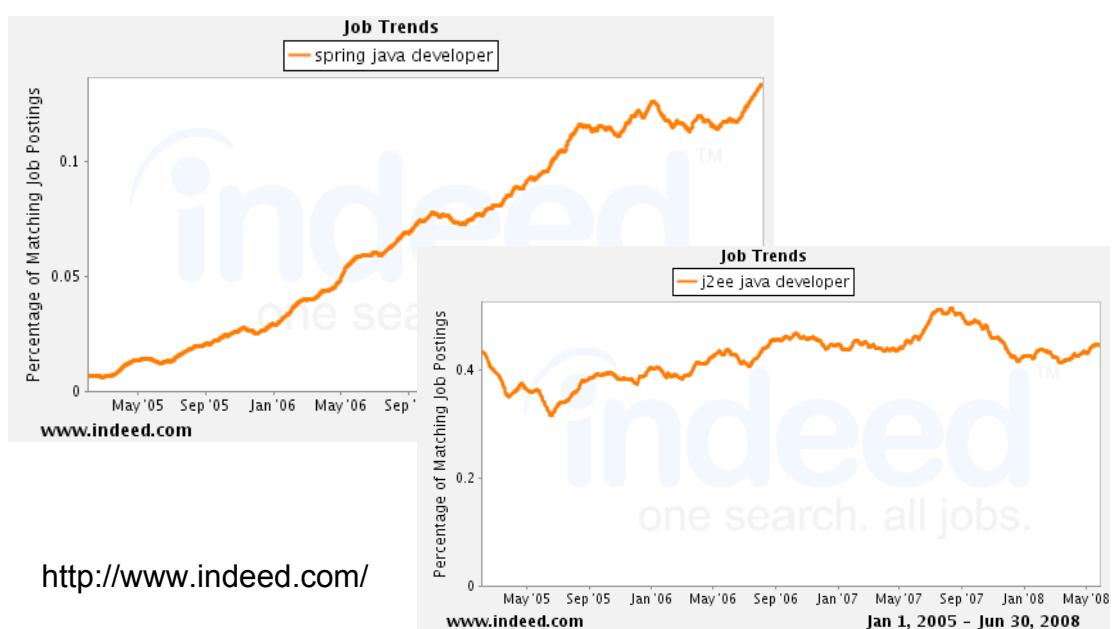
SpringとJSUG

- Spring Framework
 - SpringSource(旧Interface21)のロッド・ジョンソン氏が生みの親
 - DIxAOPコンテナの先駆け
- JSUG(Japan Springframework User Group)
 - 日本Springユーザ会
 - 国内のユーザグループ
 - 2ヶ月に1回程度の勉強会



3

ますます伸びるSpring

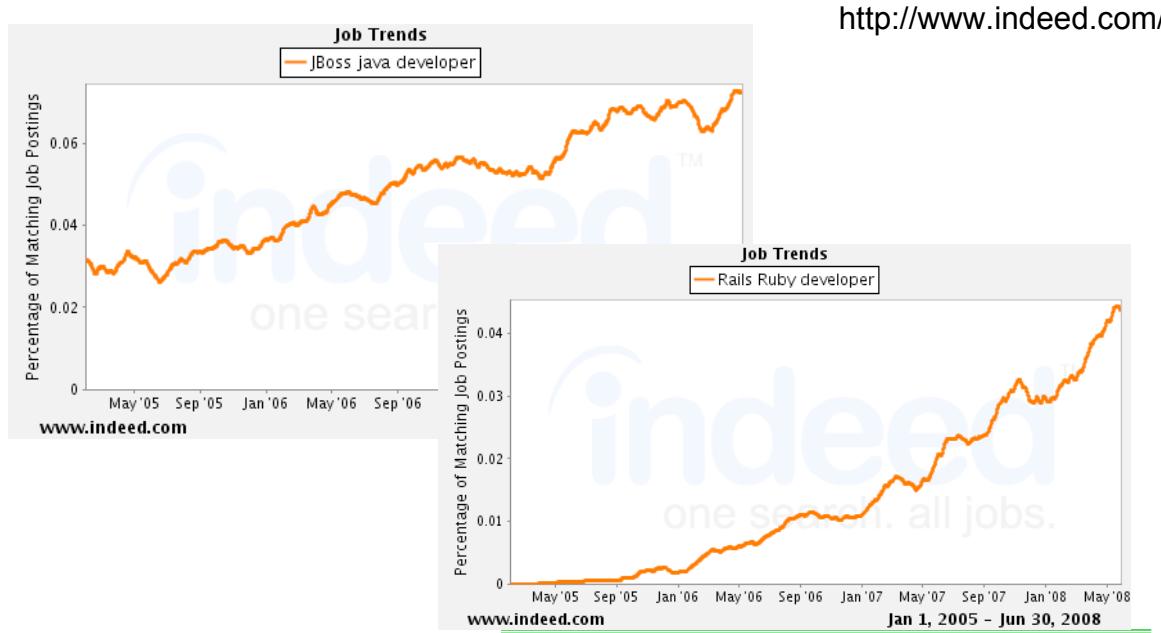


<http://www.indeed.com/>



4

ついでに比較



5

Springの利用事例

- 過去事例
 - 2006/4時点の集計
 - 19件+α
 - Webアプリ(イントラ)や社内フレームワークなど
 - <http://springframework.jp/>で公開
- 現在
 - 収集してません(色々うるさいので...)
- 最近の傾向
 - 大規模案件が増えてきている
 - リッチクライアント(.NETやAIR/Flexなど)+Spring
 - Spring+レガシーシステム



6

ますます伸びるJSUG

- 多分、今年になってようやくmixiのコミュニティをメンバ数で越えました！
 - 日本Springユーザ会
 - メンバー数: 346 名
 - mixiコミュニティ Spring Framework
 - メンバー数: 326 名

(2008/08/08現在)



7

ここからのアジェンダ

- Spring2.xの新機能
- S2AP
- Spring Dynamic Module
- Spring Batch
- Slim3



8

Spring2.xの新機能

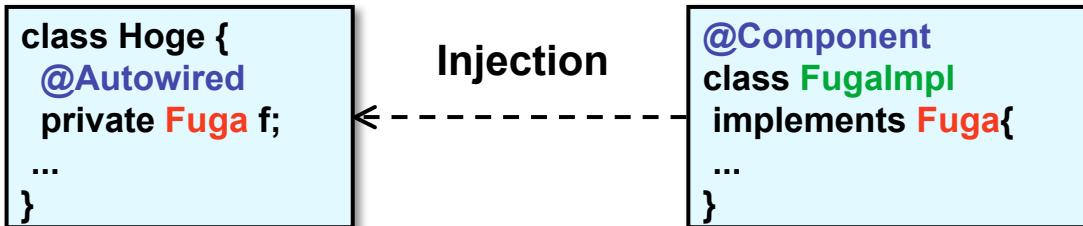
- 2007/11/19 リリース
 - 現在は2.5.5
- 2.xの特徴
 - アノテーションの充実
 - ロード・タイム・ウェービング
 - パフォーマンスの向上
 - etc...



9

DIが簡単になった

- インジェクションの設定(自動検出)
 - @Autowired
- Beanの設定
 - @Component



10

ソースコード

```
public class Hoge {  
    @Autowired  
    private Fuga f;  
    @Autowired  
    private FugaFuga ff;  
    ...  
    @Autowired  
    public Hoge(Fuga f, FugaFuga ff) {...}  
    @Autowired  
    public m(Fuga f, FugaFuga ff) {...}
```

```
@Component  
public class Fuga {  
    ...  
}
```

```
@Component  
public class FugaFuga {  
    ...  
}
```

11



Bean定義ファイル

```
<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns=http://www.springframework.org/schema/beans  
       xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance  
       xmlns:context="http://www.springframework.org/schema/context"  
       xsi:schemaLocation="  
http://www.springframework.org/schema/beans  
http://www.springframework.org/schema/beans/spring-beans-2.5.xsd  
http://www.springframework.org/schema/context  
http://www.springframework.org/schema/context/spring-context-2.5.xsd">  
    <context:annotation-config />  
    <context:component-scan base-package="sample.di.annotation, xxxx"/>  
</beans>
```

12



インジェクションの設定

- `@Autowired`
 - `byType`
 - `@Qualifier`
 - `@Autowired`と組み合わせて`byName`
 - `@Qualifier("HogelImpl");`
 - `@Resource`
 - 後述
 - `@Required`
 - Beanを必ずインジェクションさせたい時に、`@Resource`と組み合わせて利用
-



13

Beanの設定

- `@Component`
 - SpringのBeanとして扱う
 - `@Repository`
 - `@Component`の拡張、データアクセス層DAO用
 - 例外を全てDataAccessExceptionに変換
 - `@Service`
 - `@Component`の拡張、ビジネス層Service用
 - 今後、機能追加予定
 - `@Controller`
 - `@Component`の拡張、プレゼン層Controller用
-



14

JSR-250アノテーションサポート

- DataSourceなどのJNDI Lookupのサポート
 - @Resource
 - JNDIからローカルまで対応
 - デフォルトでフィールド名を使用

```
...  
@Resource  
private DataSource dataSource;  
...
```

```
...  
<context:annotation-config />  
<jee:jndi-lookup id="dataSource" jndi-name="java:comp/env.jdbc.hoge" />  
...
```

JSR-250アノテーションサポート

- ライフサイクル
 - @PostConstruct
 - @PreDestroy

```
public class Hoge {  
    ...  
    @PostConstruct  
    public void start() {...}  
    @PreDestroy  
    public void stop() {...}  
    ...
```

```
...  
<context:annotation-config />  
...
```

その他の代表的なアノテーション

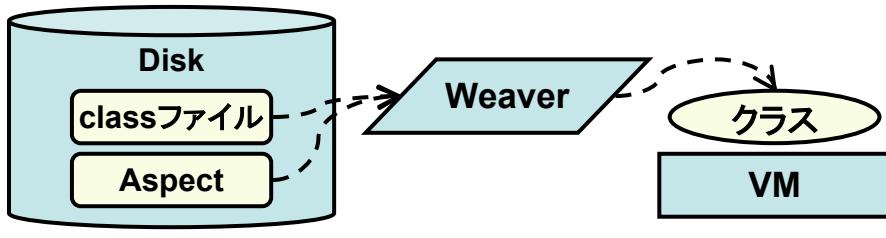
- スコープ
 - @Scope
 - singleton, prototype, session, request…
- アスペクト指向
 - @Aspect, @Pointcut, @Around…
 - 色々ある
- トランザクション
 - @Transactional
 - インタフェースにも使えるので便利



17

ロード・タイム・ウェービング

- Proxyタイプ織り込みだけではなく、ロード時の織り込みにも対応
- AspectJの利用
- <context:load-time-weaver/>
- META-INF/aop.xml



18

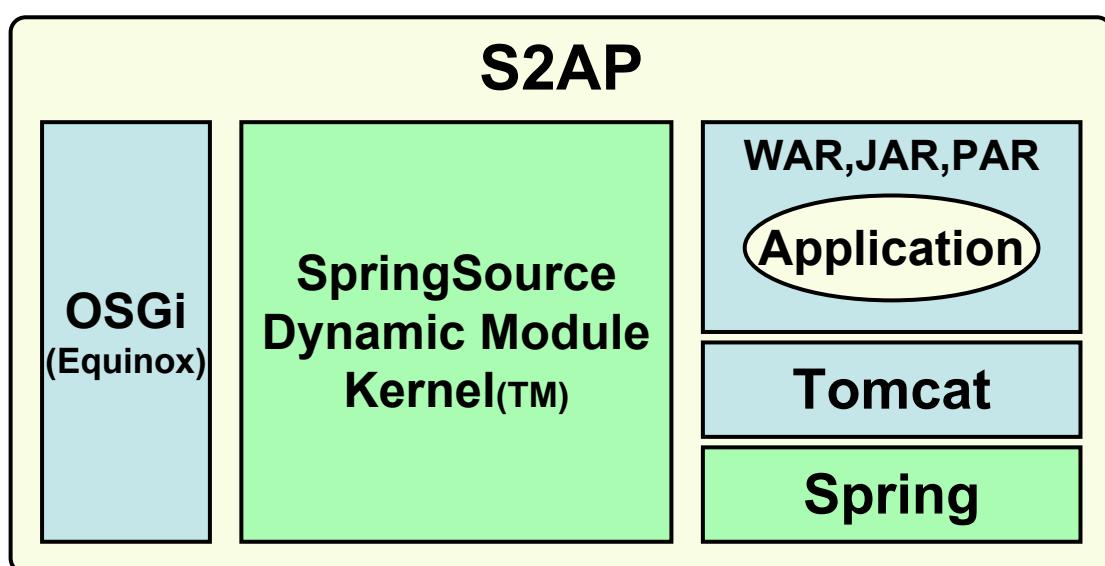
SpringSource App Platform (S2AP)

- 名前のとおり、SpringSourceのもの
 - 現在はβ版, GPL-ver3
- ライトウェイトなアプリケーションサーバ
 - Tomcat + Spring + OSGi(Spring Dynamic Module)
 - JavaEE非標準
 - S2AP用のEclipse Plug-Inあり
- 適応プロジェクト
 - EJBはWith Outで良い
 - アプリケーションのリアルタイムな更新をおこないたい
 - 自動デプロイ
 - リソースの使用量を削減したい
 - 新しいWebアプリと古いWebアプリを同時に動かしたい



19

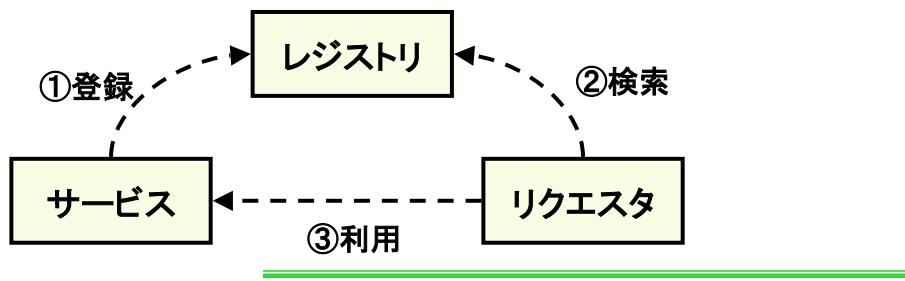
S2APの概要図



20

OSGi (Open Service Gateway initiative)

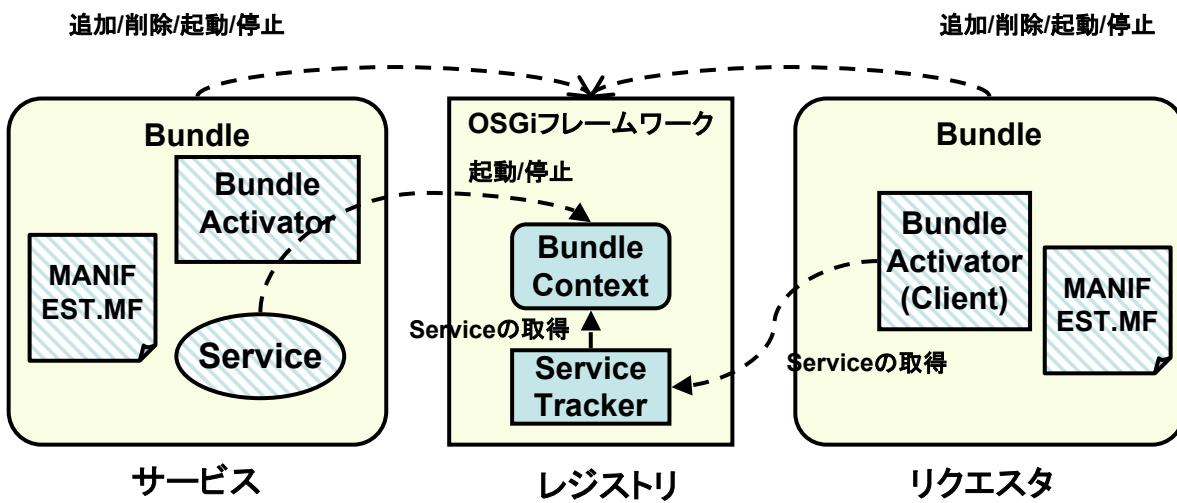
- ・ モジュラリティを上げる技術
- ・ VMを停止しないで、サービスの追加、起動、停止、削除が可能
- ・ ロードするライブラリのマルチバージョンサポート
- ・ Java7で標準に採用？



21

OSGiの仕組み

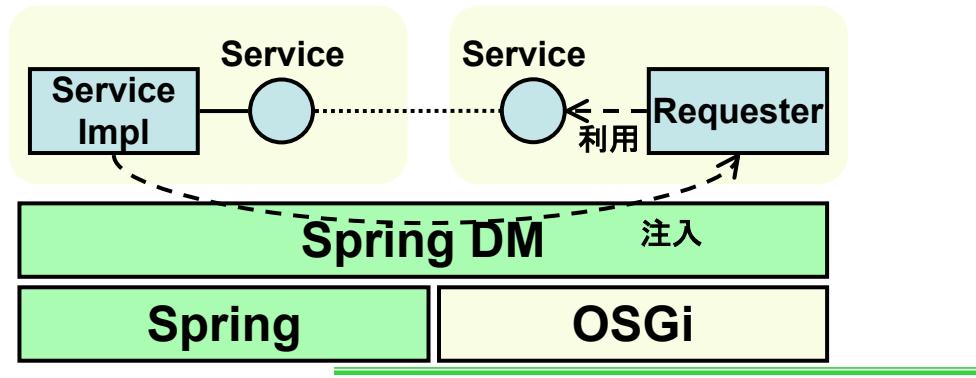
- ・ Bundle: モジュール



22

Spring Dynamic Module

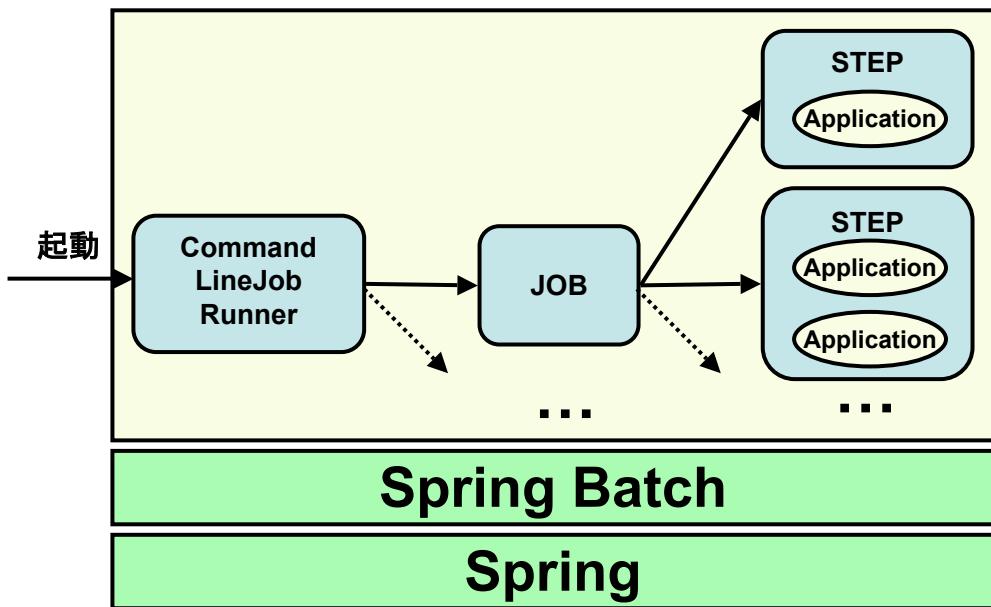
- 2008/07/29 1.1.1 リリース
- OSGi APIに非依存
- POJOでサービス/リクエスタが作成可能
- Bundleなど意識しなくて良い



Spring Batch

- 2008/7/31 1.1.1リリース
 - 現在、利用可能なのはスタンドアロンのJavaアプリケーションのみ
- Springを利用したバッチが開発可能
 - バッチの呼出し, バッチの構造化, レコード処理モデル
 - 実行監視や運用監視, 実行スクリプトなどは提供されない
 - ファイルの読み込み等をする便利なライブラリ付き
 - XMLを思う存分楽しめる(!?)

Spring Batchの概要図



25

Slim3

- Slim3 (Simplicity, Less Is More)
 - HOT deploy可能なフルスタックフレームワーク
 - Springをベースに、SAStruts, HOT deploy, S2JDBCを加えた感じ
 - 設定ファイルを最低限にし、CoCではなくアノテーションを利用



26

ありがとうございました

次回

Springだから Spring

でお会いしましょう



27

参考資料(JSUG)

- もっと詳しく知りたい方
 - 本発表資料(S2AP除く)の元となったドキュメントやサンプルなどは以下のURLにあります
 - <http://groups.google.co.jp/group/jsug/files>
 - Spring 2.5 アノテーション
 - Spring AOP 2.5
 - Spring Dynamic Module
 - Spring Batch
 - Slim3
 - etc



28

ライセンスについて

- ① JSUGマスコットアイコン(本スライド左下)が残されている場合に限り、本作品(またそれを元にした派生作品)の複製・頒布・表示・上演を認めます。
- ② 非商用目的に限り、本作品(またそれを元にした派生作品)の複製・頒布・表示・上演を認めます。
- ③ 本作品のライセンスを遵守する限り、派生作品を頒布することを許可します。