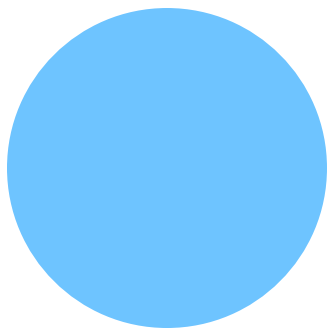
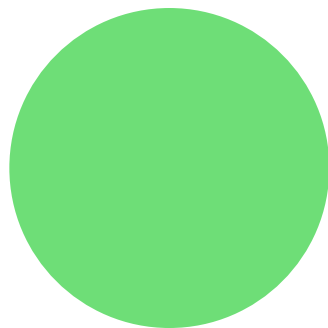




# T2でつながろう!

-つなぐつながるWebフレームワーク「T2」の紹介



T2プロジェクト  
米林 正明  
片山 暁雄





# 自己紹介

---

- 名前

- 米林 正明

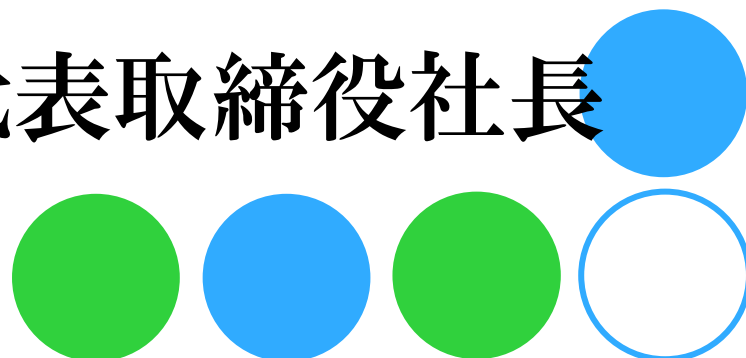
- ID

- id:yone098

- 所属

- 株式会社Abby

代表取締役社長





# 自己紹介

---

- 名前

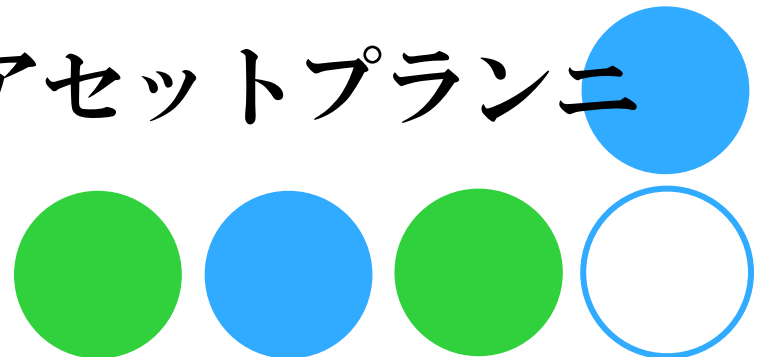
- 片山 暁雄

- ID

- id:c9katayama

- 所属

- 株式会社キャピタルアセットプランニ  
ング

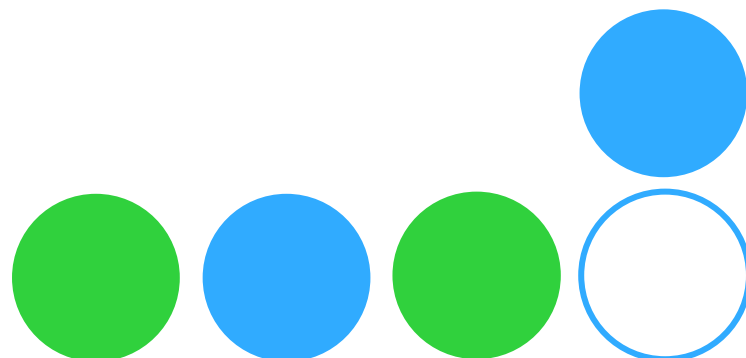




# Agenda

---

- T2の概要
- T2の基本姿勢
- T2の目指す所
- 機能紹介
- DIコンテナ非依存
- 今後の展開
- デモ

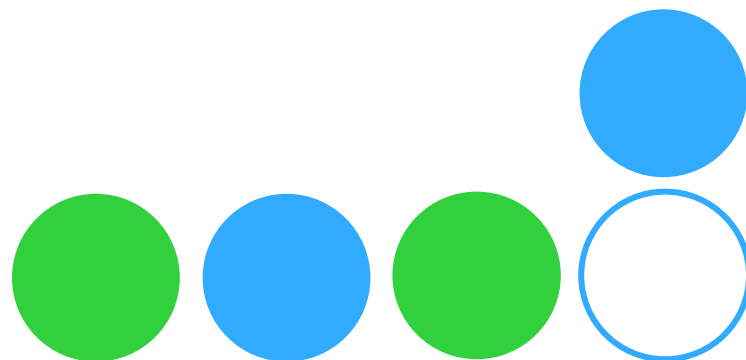




# T2 - The WEB Connector -

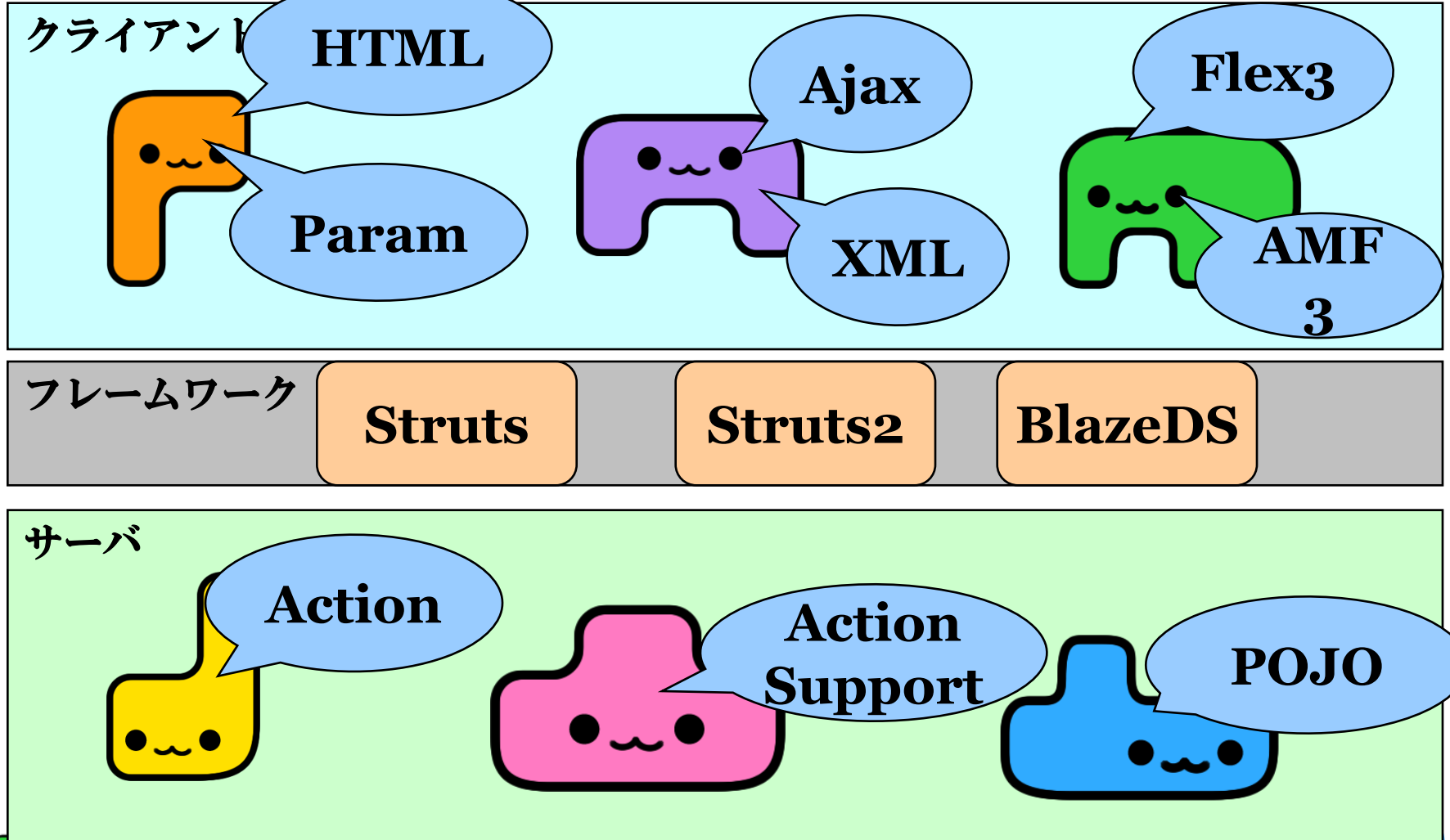
---

## T2の概要





# Webを取り巻く環境



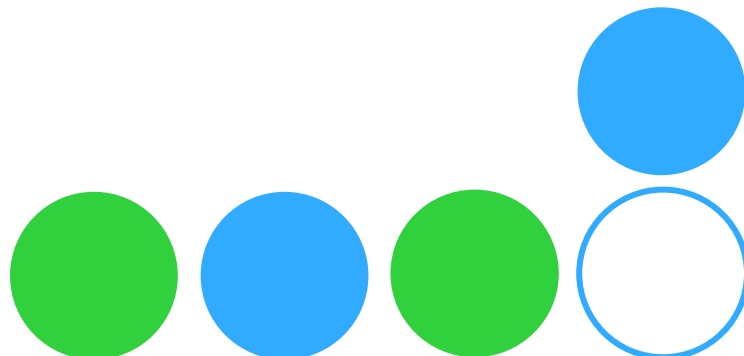


# Webを取り巻く環境

---



無理



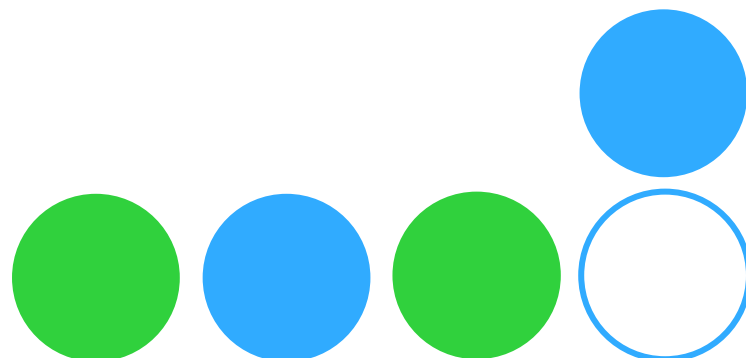


# T2とは

---

## ● T2とは

- Webフレームワーク
- テーマ「つなぐ・つながる」
- クライアントとユーザーコードの仲介役

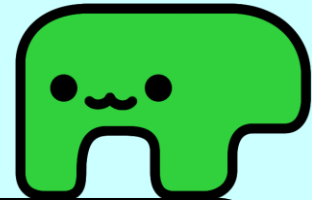
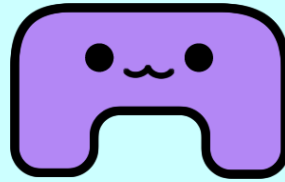






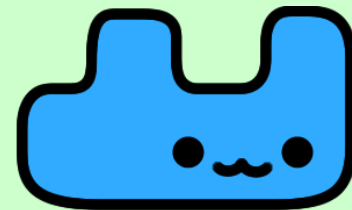
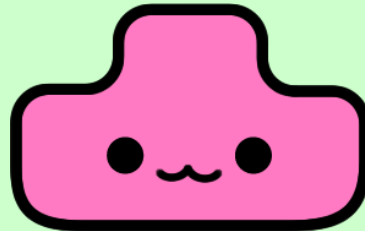
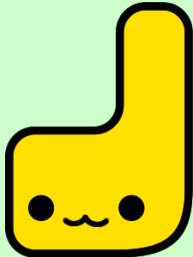
# T2でつなぐつながる

クライアント



T2!

サーバ

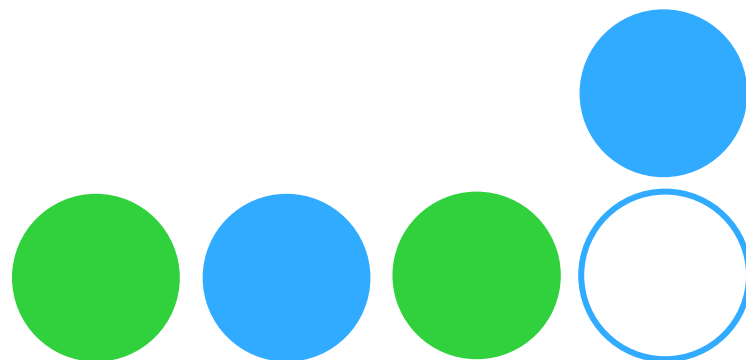




# T2 - The WEB Connector -

---

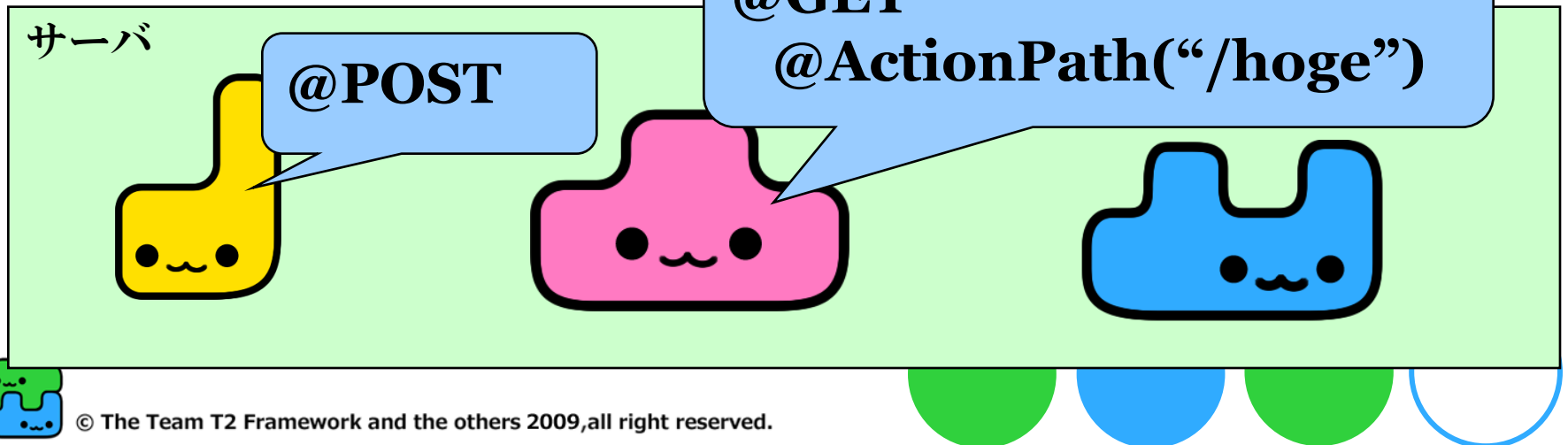
## T2のスタイル





# T2のスタイル

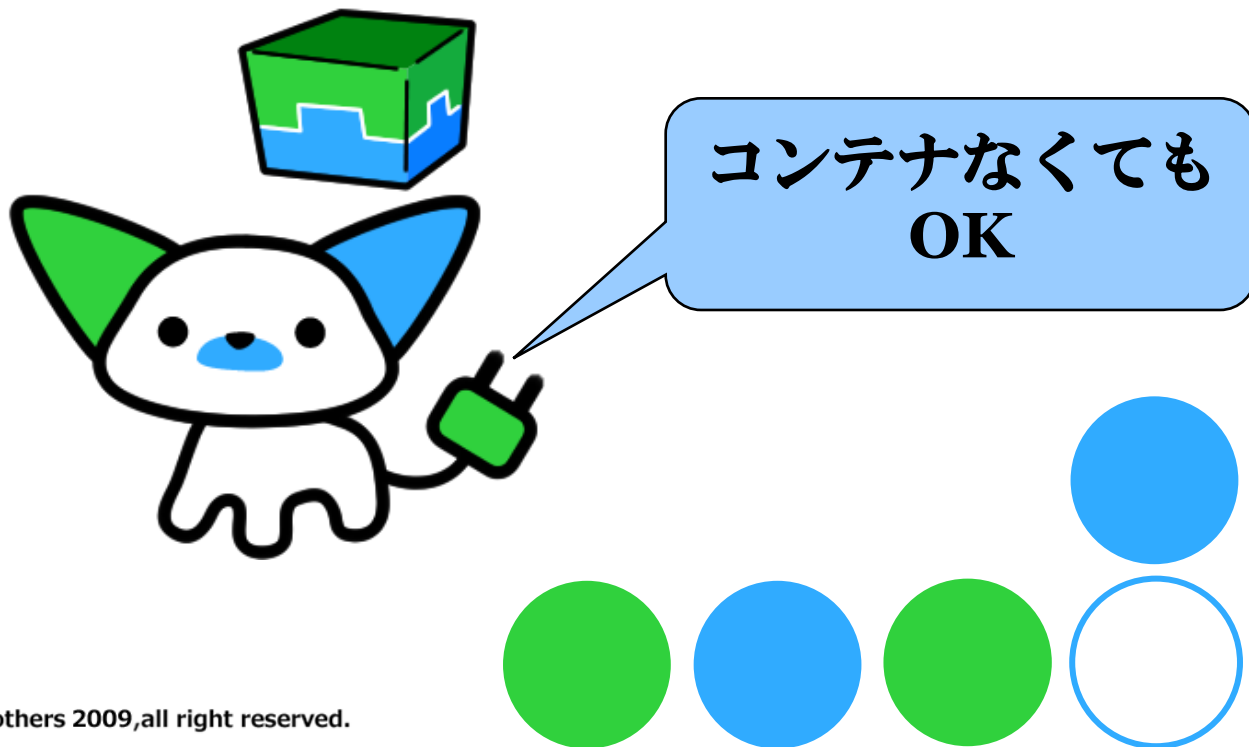
- アノテーションドリブン
  - エントリーポイントのPOJOにアノテーション付与
  - エントリーポイントの分かりやすさ重視
  - 規約とアノテーションを適切に利用





# T2のスタイル

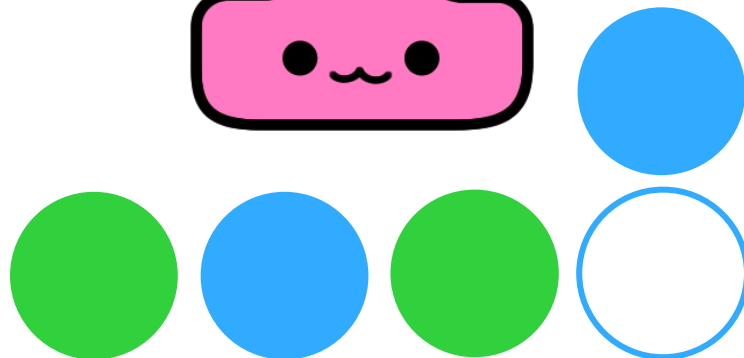
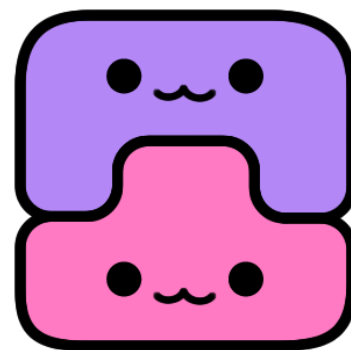
- 基本はステートレス
- 特定のコンテナへ依存しない





# T2のスタイル

- ユーザーに介入してほしい
  - コアはシンプルに extで拡張を提供
  - できればプロジェクトごとに必要な機能を作っ  
てほしい
  - extはあなたが作る！

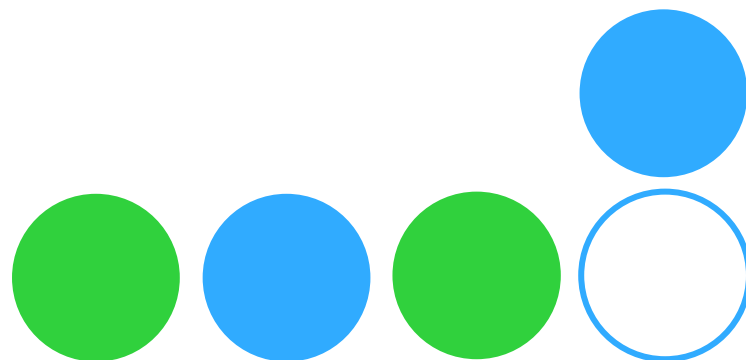




# T2 - The WEB Connector -

---

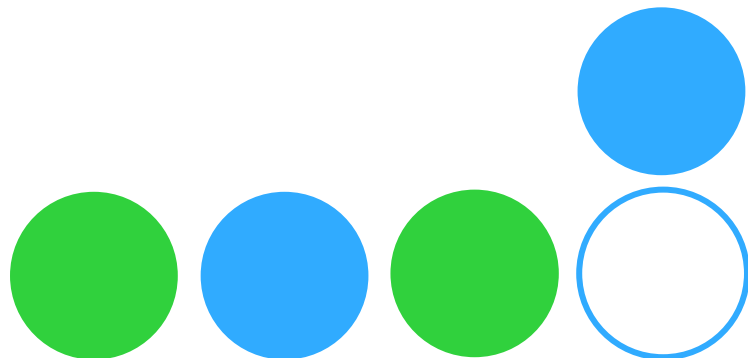
## T2の目指す所





## T2の目指す所

- マルチビューをさばけるフレームワーク
- 過渡期でも使えるフレームワーク
- 依存性の少ないフレームワーク
- ユーザーが支配出来るフレームワーク

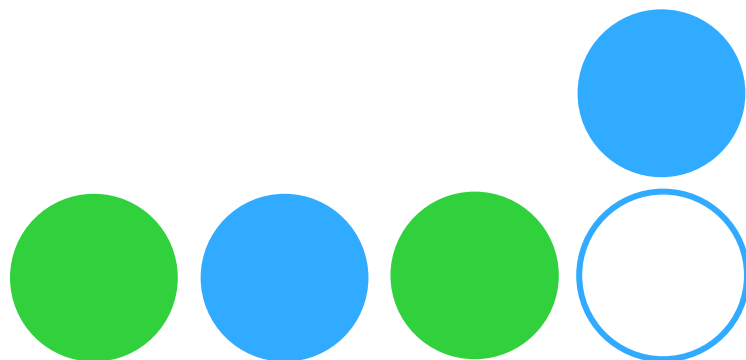




# T2 - The WEB Connector -

---

## 機能紹介



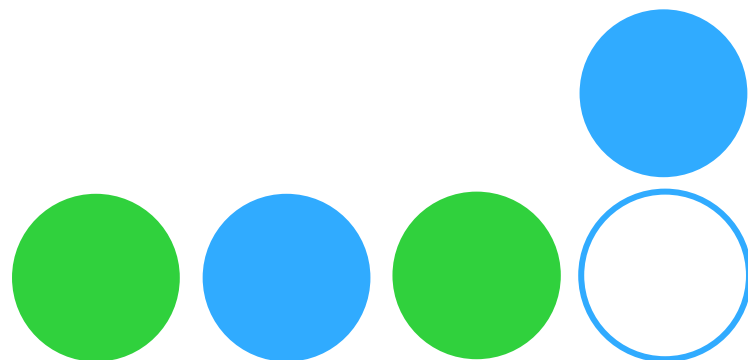




## T2 - The WEB Connector -

---

# リクエストとPOJOの マッピング



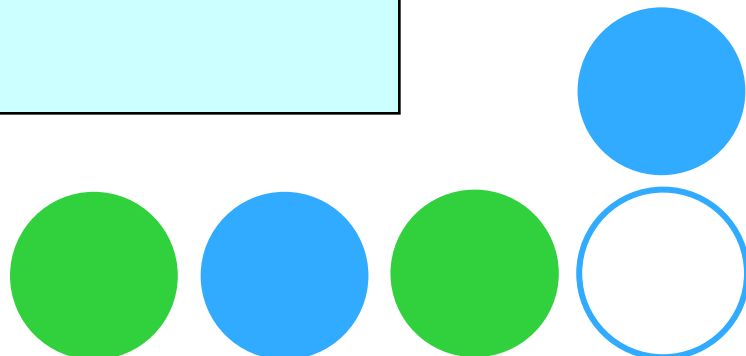


# 基本機能

## STEP1: @Pageでクラス特定

`http://yoursite.com/t2app/hello  
/world`

`@Page("/hello")  
public class MyHello {  
...  
}`





# 基本機能

## STEP 2:メソッドアノテーションでメソッド特定

<http://yoursite.com/t2app/hello/world>

```
@Page("/hello")
public class MyHello {

    @GET
    @ActionPath("/world")
    public Navigation helloworld(){

    ...
}
```



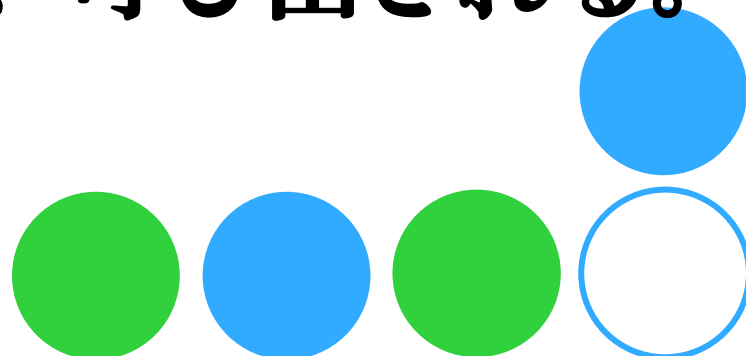


# メソッド特定方法

---

```
@GET GETのリクエストで かつ  
@ActionPath("/world") /worldのパスの場合  
public Navigation helloworld(){  
...  
}
```

→ 全マッチの場合、呼び出される。

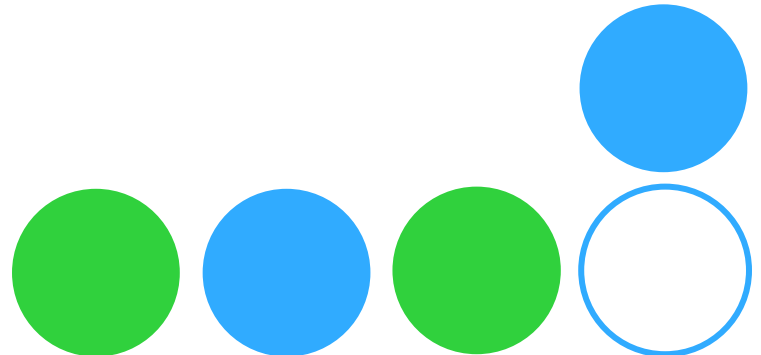




# メソッドアノテーション

---

- @GET, @POST
- @ActionPath
- @ActionParam
  - リクエストパラメータのkey
  - key-value指定も可能
- @Ajax
- @Amf (0.6から)
- @Default

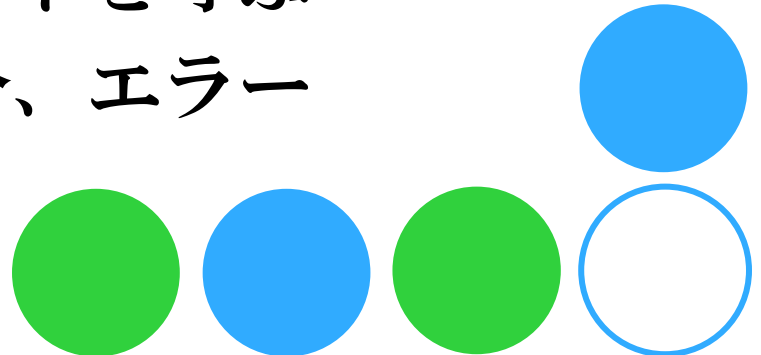




# アノテーションマッチ

---

- 複数のメソッドがマッチした場合
  - たくさんアノテーションがマッチした方を優先
- 同じ数のアノテーションがマッチした場合
  - 先に見つかったほうを使用 (バージョン0.5)
  - ただしここは検討中
- マッチするのがなかった場合
  - @Defaultがついたメソッドを呼ぶ
  - @Defaultつきがない場合、エラー

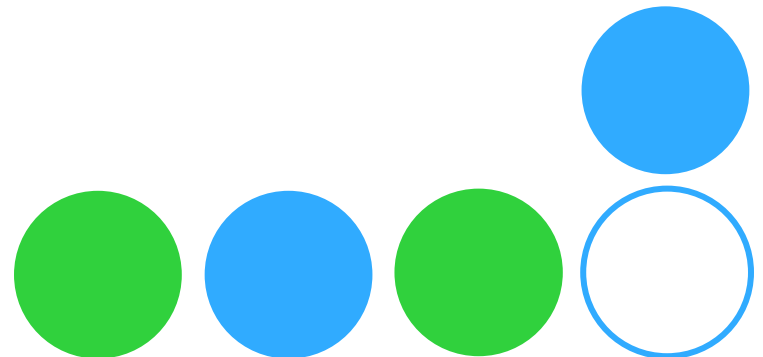




# メソッド引数特定方法

---

- 引数のアノテーションを見て、引数を解決
  - @RequestParam,
  - @RequestHeader
  - @SessionAttr
  - @Upload
  - @Form
  - @Index
  - @Var

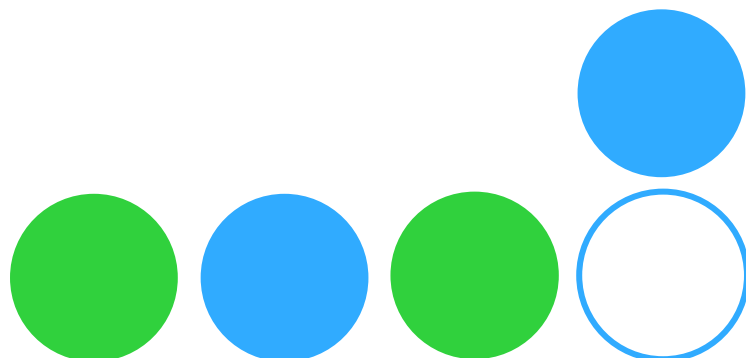




# メソッド引数特定方法

---

- 引数の型を見て、引数を解決
  - HttpServletRequest, HttpServletResponse
  - HttpSession
  - ServletContext
  - Cookie/Cookie[]
  - WebContext
  - Request, Response
  - UploadFile
  - ErrorInfo





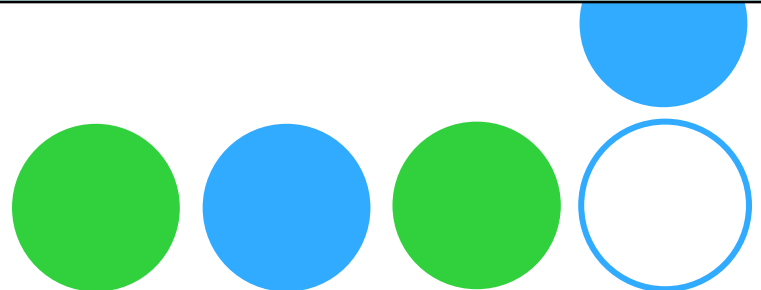


# メソッド引数特定方法

---

```
@ActionPath("/world/{id}")  
public Navigation helloworld(  
    HttpSession session,  
@Var("id") String id,  
@RequestParam("name") String name){
```

...

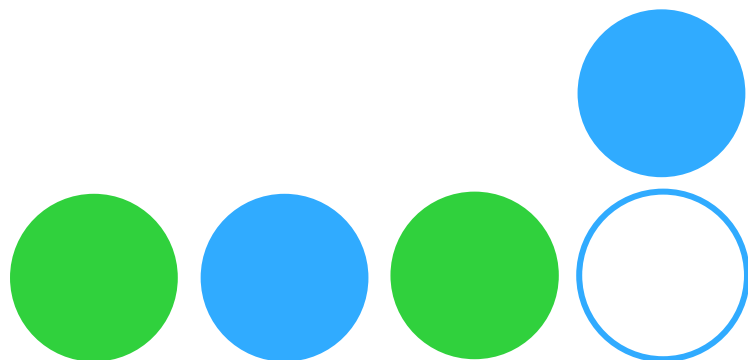




# T2 - The WEB Connector -

---

## レスポンス

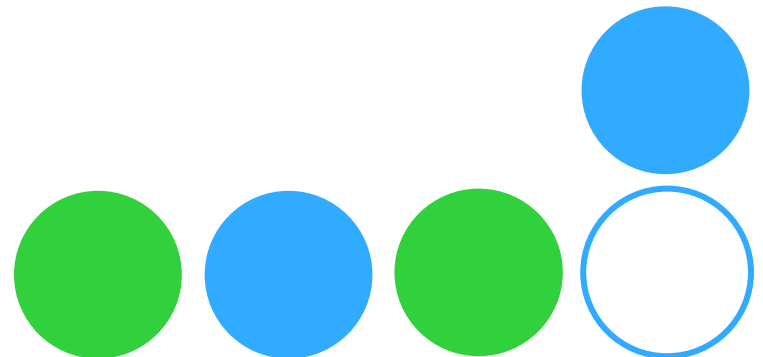




# レスポンス

---

- 戻り値として、Navigationインターフェースを実装したクラスのインスタンスを返す
  - Forward
  - Redirect
  - SimpleText
  - Direct
  - Json
  - NoOperation
  - PassThrough





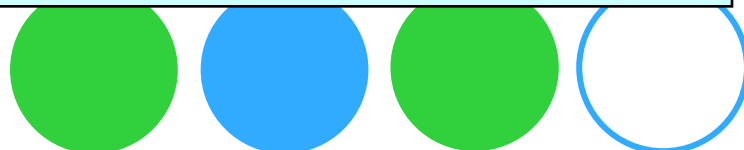
# レスポンス

**//JSPにForward**

```
@ActionPath("/world/{id}")  
public Navigation helloworld(){  
    return Forward.to("/WEB-INF/jsp/hoge.jsp");  
}
```

**//画像ファイルを返す**

```
@ActionPath("/worldImage")  
public Navigation helloworldImage(){  
    File image = new File("world.png");  
    return Direct.from(image);  
}
```

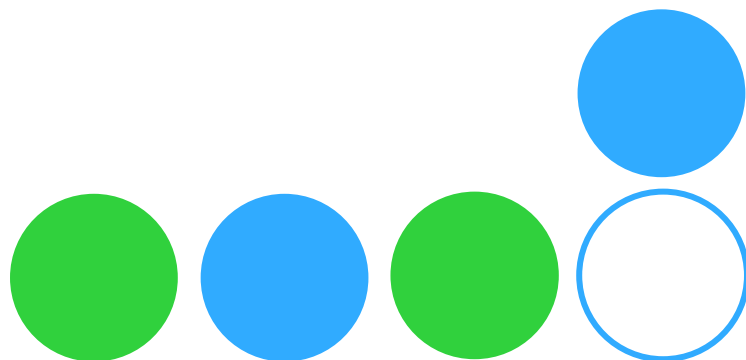




## T2 - The WEB Connector -

---

# DIコンテナ非依存





# DIコンテナ非依存

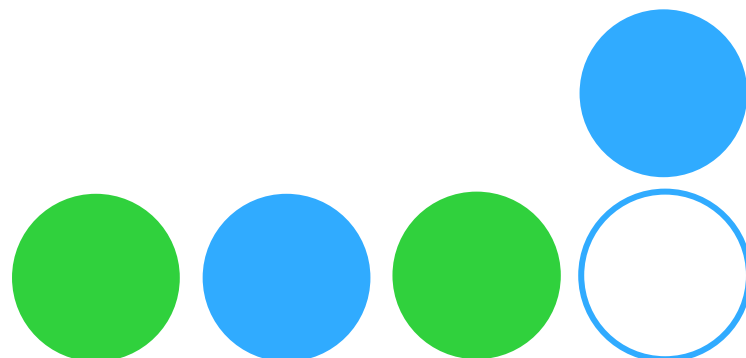
---

## ● T2が動作するDIコンテナ

○ Guice

○ Spring

○ Lucy





# DIコンテナ非依存

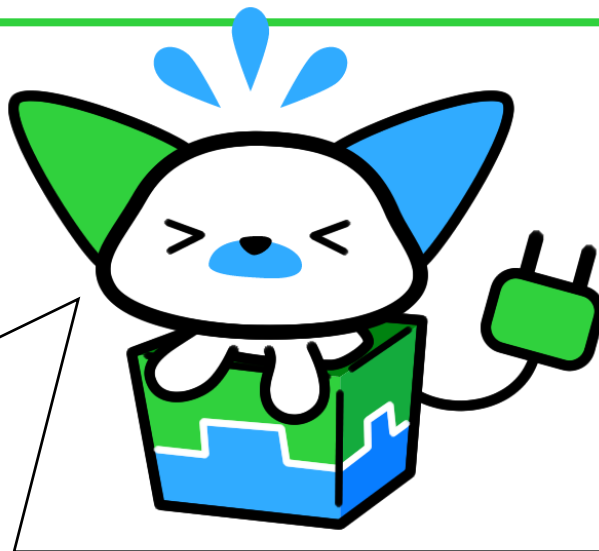
今日は  
Seasar

カンファレンス  
やないか！

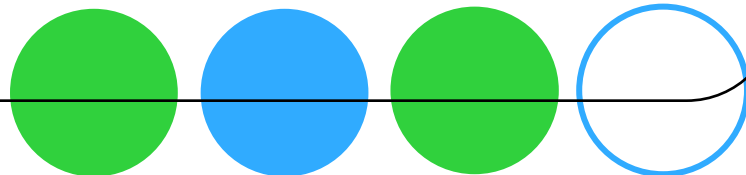




# DIコンテナ非依存



Seasar君、  
どこ行ってしもたんや...







# DIコンテナ非依存

## ● T2が動作するDIコンテナ

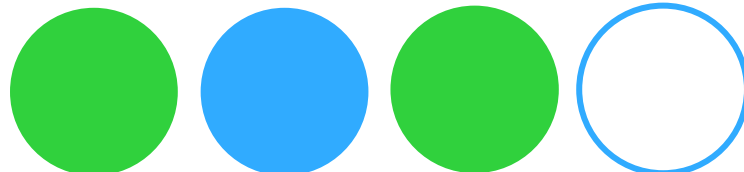
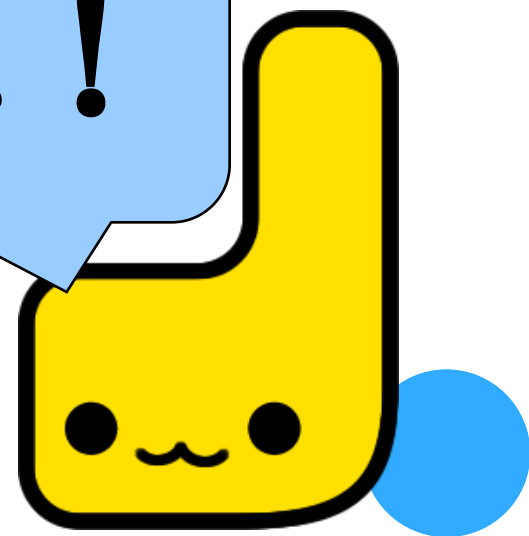
○ **Seasar2**

○ Guice

○ Spring

○ Lucy

あるよ！





# DIコンテナ非依存

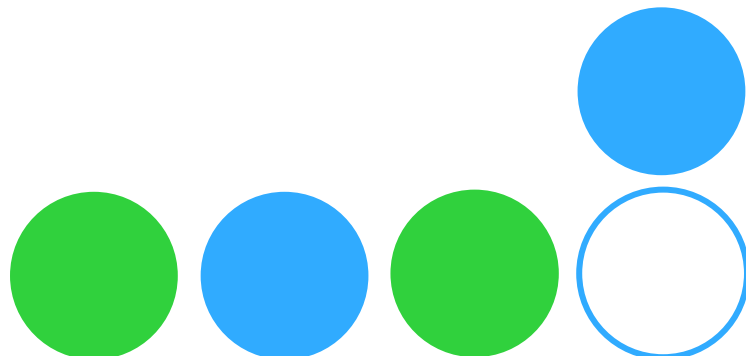
- 環境構築～Seasar2編～

- 参照

- t2-employee

- 構成

- S2.4 + S2Dao

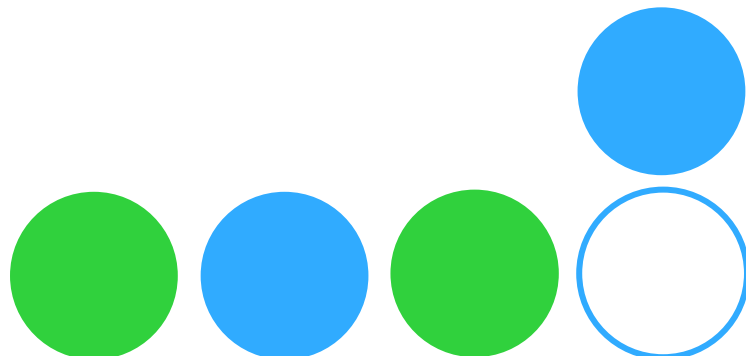




# DIコンテナ非依存

---

- 作成設定ファイル～Seasar2編～
  - aop.dicon
  - app.dicon
  - dao.dicon
  - jdbc.dicon





# DIコンテナ非依存

## ○web.xml～Seasar2編～

```
<filter>
  <filter-name>t2</filter-name>
  <filter-class>org.t2framework.filter.T2Filter
</filter-class>
  <init-param>
    <param-name>t2.rootpackage</param-name>
    <param-value>examples.employee.page</param-value>
  </init-param>
  <init-param>
    <param-name>t2.container.adapter</param-name>
    <param-value>org.t2framework.plugins.container.Seasar2Adapter</param-value>
  </init-param>
  <init-param>
    <param-name>t2.eagerload</param-name>
    <param-value>>true</param-value>
  </init-param>
</filter>
```



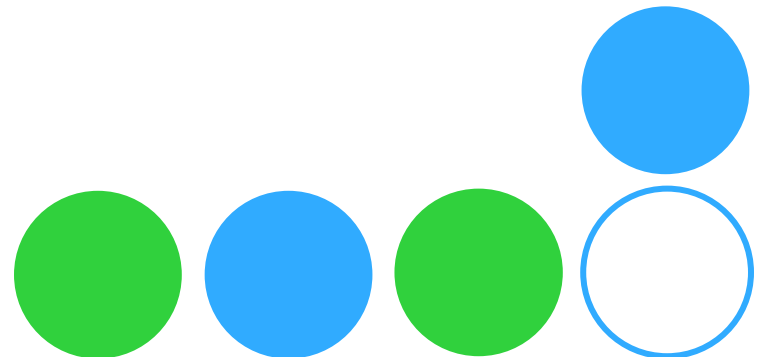


# DIコンテナ非依存

---

## ○Pageクラス～Seasar2編～

```
@RequestScope  
@Page("login")  
public class LoginPage
```





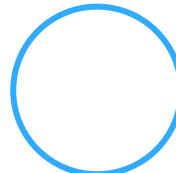
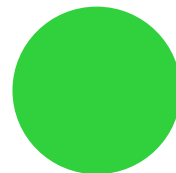
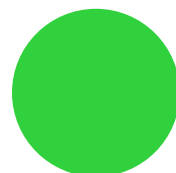
# DIコンテナ非依存

## ○PageクラスDI～Seasar2編～

```
protected FooService fooService;
```

```
@Binding(bindingType = BindingType.MUST)
```

```
public void setFooService(FooService fooService) {  
    this.fooService = fooService;  
}
```

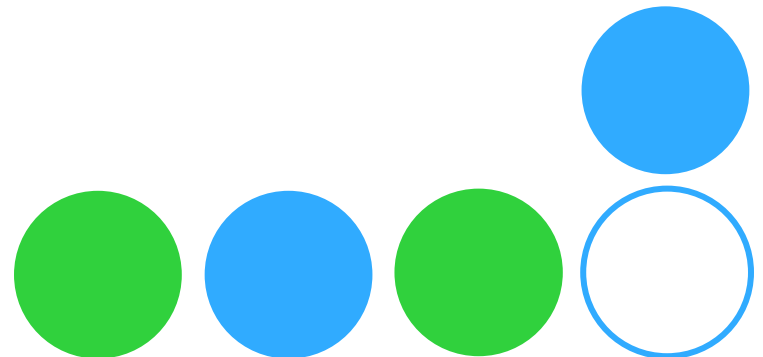




# DIコンテナ非依存

---

- 環境構築～Guice編～
  - 参照
    - t2-guice
  - 構成
    - Guice + JPA

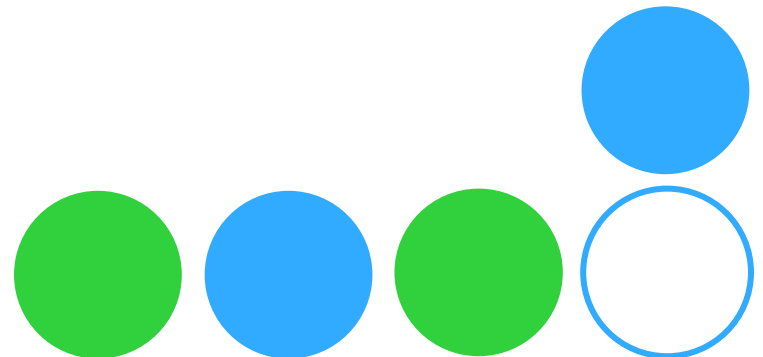




# DIコンテナ非依存

---

- 作成設定ファイル～Guice編～
  - persistence.xml
  - ※Guice用のモジュール（Javaで）







# DIコンテナ非依存

## ○web.xml～Guice編～

```
<filter>
  <filter-name>t2</filter-name>
  <filter-class>org.t2framework.t2.filter.T2Filter</filter-class>
  <init-param>
    <param-name>t2.rootpackage</param-name>
    <param-value>org.t2framework.samples.guice.page</param-value>
  </init-param>
  <init-param>
    <param-name>t2.container.adapter</param-name>
    <param-value>org.t2framework.t2.adapter.GuiceAdapter</param-value>
  </init-param>
  <init-param>
    <param-name>t2.exclude-resources</param-name>
    <param-value>css, js</param-value>
  </init-param>
</filter>
```



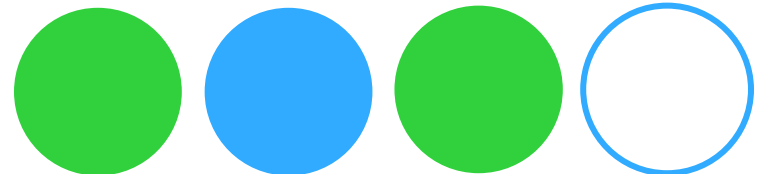


# DIコンテナ非依存

---

## ○Pageクラス～Guice編～

```
@RequestScoped  
@Page("add")  
public class AddPage
```





# DIコンテナ非依存

---

## ○PageクラスDI～Guice編～

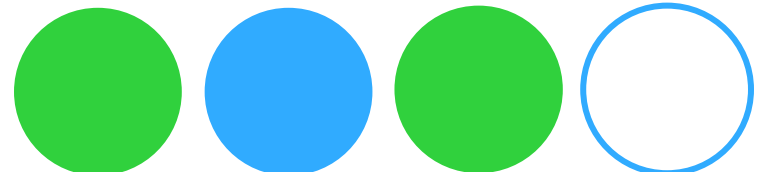
**@Inject**

```
private CalcService calcService;
```

**@Inject**

**@RequestParameters**

```
Map<String, String[]> params;
```





# DIコンテナ非依存

---

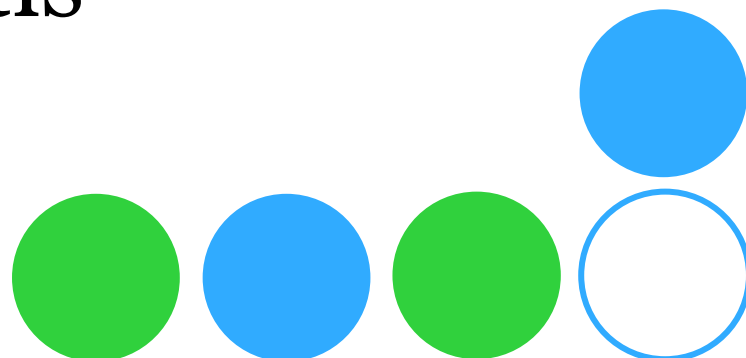
- 環境構築～Spring編～

- 参照

- t2-spring

- 構成

- Spring2.5 + iBatis

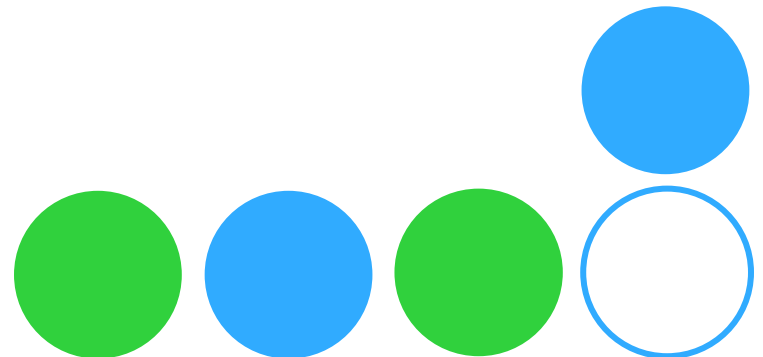




# DIコンテナ非依存

---

- 作成設定ファイル～Spring編～
  - applicationContext.xml
  - dao-ibatis.xml
  - sql-map-config.xml
  - XxxSqlMap.xml





# DIコンテナ非依存

## ○web.xml～Spring編～

```
<filter>
  <filter-name>t2</filter-name>
  <filter-class>org.t2framework.t2.filter.T2Filter</filter-class>
  <init-param>
    <param-name>t2.rootpackage</param-name>
    <param-value>org.t2framework.samples.spring.page</param-value>
  </init-param>
  <init-param>
    <param-name>t2.config</param-name>
    <param-value>applicationContext.xml</param-value>
  </init-param>
  <init-param>
    <param-name>t2.container.adapter</param-name>
    <param-value>org.t2framework.t2.adapter.SpringAdapter</param-value>
  </init-param>
  <init-param>
    <param-name>t2.eagerload</param-name>
    <param-value>>true</param-value>
  </init-param>
  <init-param>
    <param-name>t2.exclude-resources</param-name>
    <param-value>css, js</param-value>
  </init-param>
</filter>
```

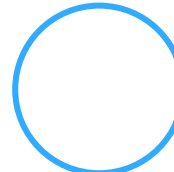
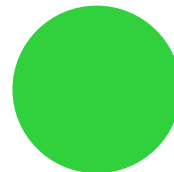
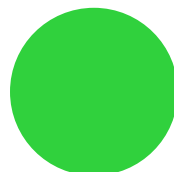


# DIコンテナ非依存

---

## ○Pageクラス～Spring編～

```
@Component  
@Page("add")  
public class AddPage
```





# DIコンテナ非依存

---

## ○PageクラスDI～Spring編～

```
private CalcService calcService;
```

```
@Autowired(required=true)
```

```
public void setCalcService(
```

```
CalcService calcService) {
```

```
    this.calcService = calcService;
```

```
}
```







# DIコンテナ非依存

---

- 動作確認アプリケーションサーバ

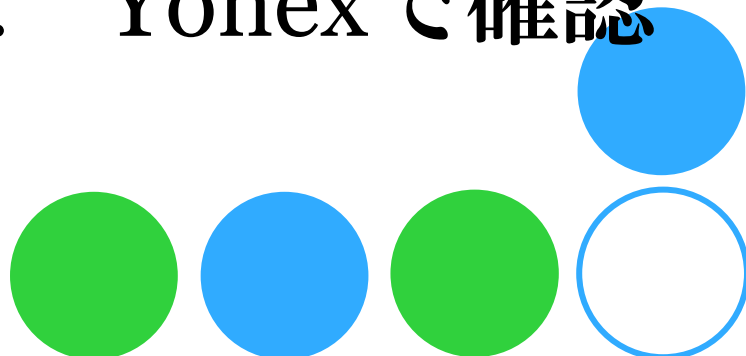
- Tomcat5x/6x

- JBoss4x/5x

- Jetty6x

- WebLogic10x

- その他いろいろ... Yonexで確認

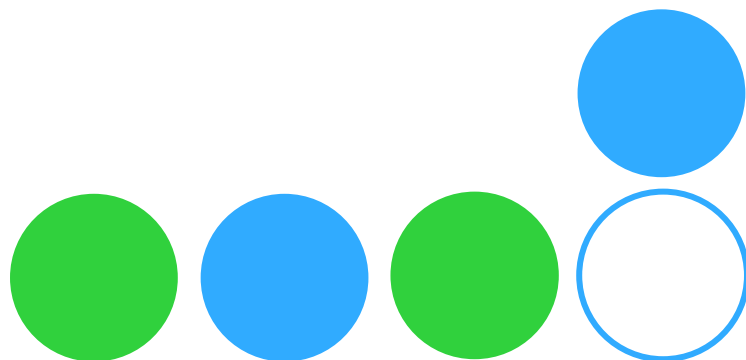




# T2 - The WEB Connector -

---

## 今後の展望

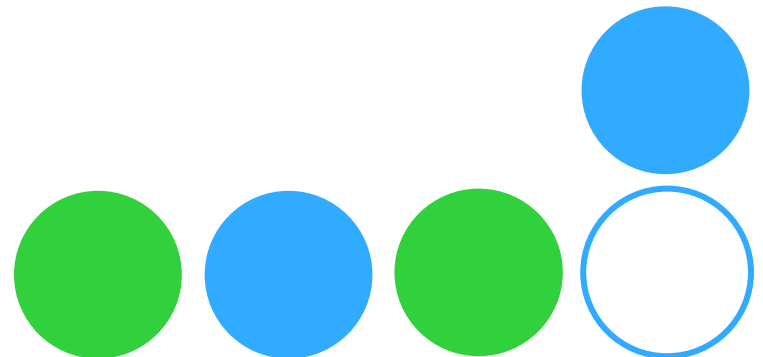




# 今後の展望

---

- バージョン0.6
  - Flex3対応 (AMF3)
  - BlazeDSのAMF3ライブラリ使用
  - もしくはS2Flex2のAMF3部分使用
- バージョン0.7
  - Atom, RSS

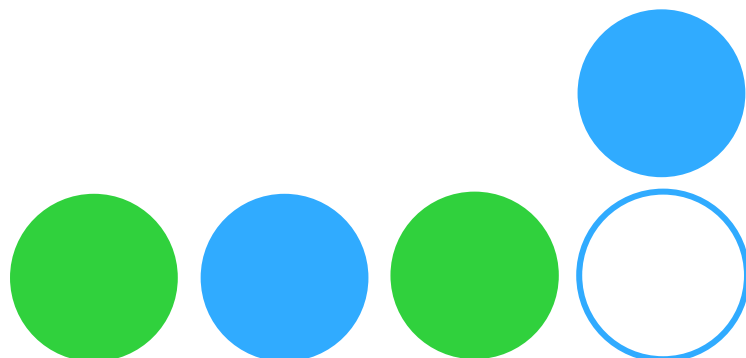




# その他プロダクト

---

- Commons
  - 共通ユーティリティ郡
- Lucy
  - DIコンテナ
  - アノテーションベース
- Daisy
  - 2 waySQLフレームワーク
  - OS(Object-SQL)マップ
  - 薄いJDBCラッパー

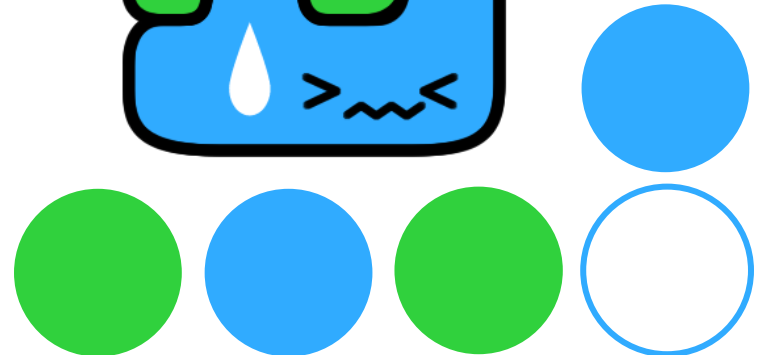
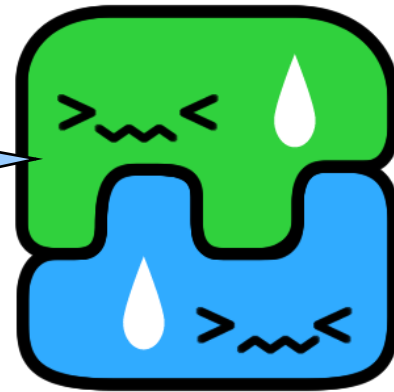




# その他プロダクト

- Yonex (仮)
- Hyper Integration Test Suite

レポート  
とかも  
提供したい!





# プロジェクトリソース

---

- サイト

- <http://code.google.com/p/t-2/>

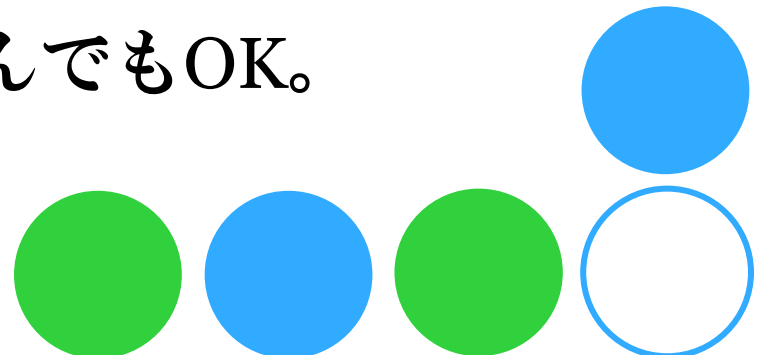
- リリースアーカイブ、ドキュメント、ソースコード

- メーリングリスト

- T2-users

- <http://groups.google.com/group/t2-users>

- 開発の話、仕様の話、なんでもOK。

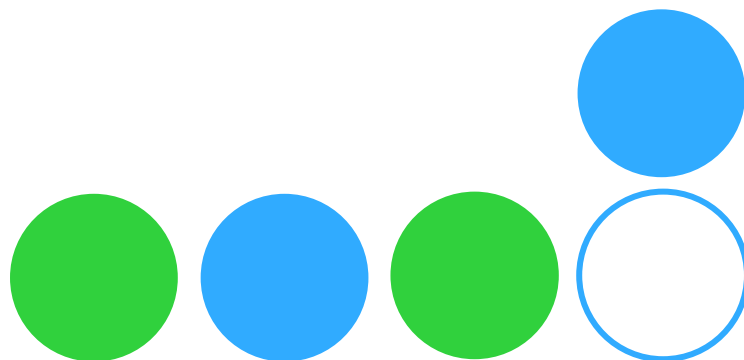




# T2 - The WEB Connector -

---

まとめ

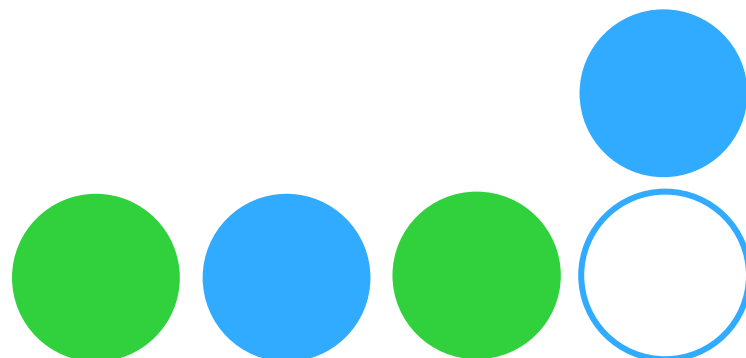




# まとめ

---

- T2はマルチビュー対応
- つなぐ機能は提供します！
- が
- 組み合わせはユーザーで！
- まだまだ発展途上。ご意見ください！



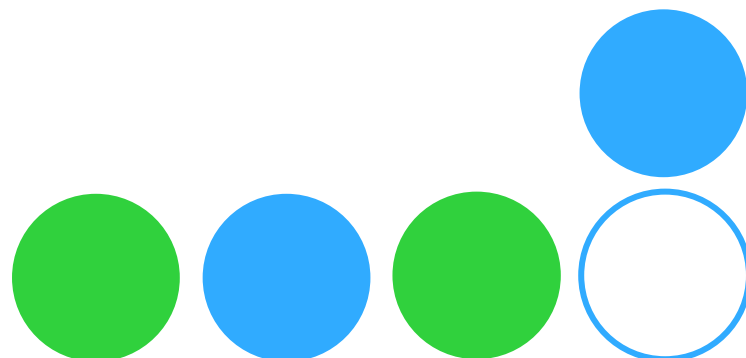




# T2 - The WEB Connector -

---

## エンドロール





# T2 - The WEB Connector -

出演





# T2 - The WEB Connector -

テツ





# T2 - The WEB Connector -





# T2 - The WEB Connector -





# T2 - The WEB Connector -

イーダ





# T2 - The WEB Connector -





# T2 - The WEB Connector -







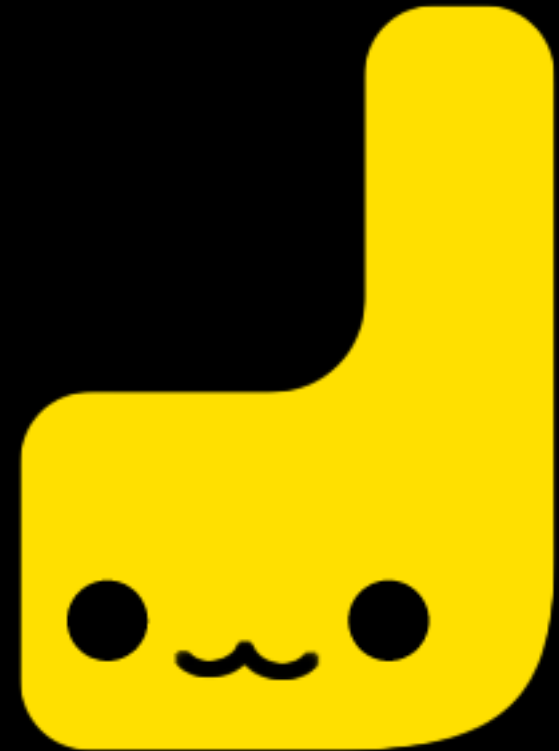
# T2 - The WEB Connector -

イーダ





# T2 - The WEB Connector -





# T2 - The WEB Connector -





# T2 - The WEB Connector -

イーダ





# T2 - The WEB Connector -





# T2 - The WEB Connector -





# T2 - The WEB Connector -





# T2 - The WEB Connector -



designed by  
カネウチカズコ







# T2 - The WEB Connector -

## 出演コミッタ





# T2 - The WEB Connector -

片山 暁雄





# T2 - The WEB Connector -





# T2 - The WEB Connector -

米林 正明





# T2 - The WEB Connector -





# T2 - The WEB Connector -

横田 健彦





# T2 - The WEB Connector -





# T2 - The WEB Connector -

大谷 晋平 (自宅)







# T2 - The WEB Connector -





## T2 - The WEB Connector -

---

御静聴ありがとうございました

