## JET(JakartaEE Toolkit)ver0.90の使い方

## 1. 展開場所とフォルダ名の制限

解凍したフォルダは、どこかの<u>ドライブの直下</u>に置かなくてはいけません。また、フォルダ名を変更する と動作しないので、注意してください。

例えば、W ドライブに置く場合は、次のようになります。



#### 2. NetBeans の起動

<ドライブ>/jakarta/netbeans/bin フォルダにある、netbeans64.exe(32 ビットシステムの場合は netbeans.exe)をダブルクリックすると起動します。ファイル名を右ボタンでクリックして、ショートカッ トを作り、デスクトップに置いておくと、すばやく起動できます。

## 3. Payara サーバーとの連携を設定する

①起動した NetBeans で、[サービス] タブをクリックし、[サーバー] を右クリックして、[サーバーの追加] を選びます。



②ダイアログで、[Payara Server]を選んで [次へ]を押します

🗘 サーバー・インスタンスの追加	×
ステップ	サーバーを選択
1. サーバーを選択 2	サーバー(S): Apache Tomcat or TomEE GlassFish Server Payara Server
	名前① Payara Server
< 戻る(B) 次 >	終了(F) 取消 ヘルブ0-0

③ [Browse...] ボタンを押して、 <ドライブ>/jakarta/payara5 を選択します。 次のような画面になるので、[次へ] を押します。

🗘 サーバー・インスタンスの追加	×
ステップ	Server Location
1. サーパーを選択 2. Secure Location	Installation Location:
3. Domain Name/Location	W: ¥ jakarta ¥ payara5 Browse
	Local Domain     Remote Domain
	Choose server to download:
	Payara Server 5.194 🗸
	Download Now
	1 Detected a Payara Server 5.201 install. Olick Next to register remote or oustom local domains.
	< <b>戻る(B) 次&gt;</b> 終了(F) 取消 ヘルプ0-0

## ④[終了]を押します

🗘 サーバー・インスタンスの追加					×
ステップ	Domain Locat	ion			
1. サーバーを選択 2. Server Location 3. Domain Name/Location	Domain:	domain1			~
	Host:	localhost			🗸 🗹 Loopback
	DAS Port:	4848	HTTP Port:	8080	🗸 Default
	Tanget:				
	User Name:				
	Password:				
	🚺 Register e:	visting embedded domain: dor	main1		
		< 戻る(B) 次	:> \$	70月) 取消	<u><u></u> <u></u> <u></u> </u>

## 4.Payara サーバーを起動する

サーバーとして、Payara サーバーが表示されるので、起動します

①Payara Server を右ボタンでクリックして [Start] を選ぶ



②下段にサーバーログが開いて、起動する様子が表示されます

出力	- Payara	Sei	rvei	r )	×													
D.	0180.	<b>L</b> I	44	00	-00	91	91	40	20	10	UΠ	50	10	00		00	33	
	01B0:	33	F7	39	10	С5	28	2Å	21	7A	1 B	8A	C4	24	F9	ЗF	15	3.9(*!z\$.?.
	0100:	С8	9A	15	20	F5	55	62	96	ED	6D	93	50	BC	Ε4	AA	78	Ubm.Px
NN	01D0:	AD	D9	СВ	0A	65	87	Å6	66	C 1	C4	81	A3	77	ЗÅ	58	1E	efw:X.
w	01E0:	0B	EE	83	8B	9D	1E	D2	52	A4	СС	1 D	6F	Β0	98	6D	94	Rom.
	01F0:	31	Β5	F8	71	0A	DC	В9	FC	7D	32	60	Ε6	ЕΒ	AF	8 <b>A</b>	01	192`
2	]  #]																	

#### 5.新規プロジェクトを作成して、実行する

builder プロジェクトを使って JSF を使う新規プロジェクトを作成します。 ※archetype を使う方法も使えますが、実はこちらの方が簡単です

①builder プロジェクトを開く

新規プロジェクトのひな型を自動作成する builder プロジェクトを開きます。

builder プロジェクトは、<ドライブ>/jakarta/projects フォルダにあるので、[ファイル]⇒[プロジェ クトを開く]と選択して開いてください。



②builder プロジェクトを一度クリックしてから、実行ボタン( D- )を押して、実行します。

コンソールに**作成したいプロジェクト名**を入力するプロンプトが表示されるので、ここでは、Sample1 と 入力して Enter キーを押します。

この操作で Sample1 プロジェクトが projects フォルダに自動作成されます。

出力	1 ×
$\square$	Payara Server × 実行(builder) ×
$\square$	maven-compiler-plugin:3.1:compile (default-compile) @ builder
⇒.	Changes detected - recompiling the module!
Q.	<ul> <li>Compiling 3 source files to W:¥jakarta¥projects¥builder¥target¥classes</li> </ul>
	🛓 exec-maven-plugin:1.5.0:exec (default-cli) @ builder
<mark>୦</mark> ୦ ବିକ୍	・プロジェクト名>sample1
	BUILD SUCCESS
	Total time: 35.498 s
	Finished at: 2020-05-03T16:42:50+09:00
	Final Memory: 11M/28M

③projects フォルダから、作成された Sample1 プロジェクトを開きます

🗘 プロジェクトを聞	開く			×
	ファイルの場所の)	projects	<ul> <li>         か         </li> <li>         プロジェカトタ(P)     </li> </ul>	
販辺便のた項目	🕕 · 🏭 builder		W¥jakarta¥projects¥sample1	
			必須プロジェクトを開く(R)	
デスクトップ				
<b>۲۲</b> ۲۲				
PC	ファイル名0ᢗ	W¥jakarta¥projects¥sample1		プロジェクトを開く
	<u>ファイルのタイプ(T)</u>	プロジェクト・フォルダ	~	取消

#### ④Sample1プロジェクトの内容を確認する



JSF フレームワークのための、設定ファイルは、WEB-INF フォルダ内に、すべて生成済です。また、リソ ースとして、CSS スタイルシート(style.css)も置いてあります。index.xhtml は次のようです。



ソースパッケージの jp.jakartaee.config には、JSF のバージョンを示すための Java クラスが入って います。JSF2.3 から CDI と連携するために必要になったクラスです。このまま置いておくだけで構いません。

persistence.xml は、jdbc/mydb というデータソースを参照します。汎用的な記述になっているので、通常は変更なしで使えます。



#### Pom ファイルは次のようです。

```
jakartaEEの8.0を使う指定です。JDKはLTSであるバージョン11を使ってコンパイルします。
```

<project <br="" xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"&gt; <modelversion>4.0.0</modelversion> <groupid>jp.jakartaee</groupid> <artifactid>sample1</artifactid> <version>1.0</version> <packaging>war</packaging> <name>sample1</name></project>	
<properties></properties>	
<project.build.sourceencoding>UTF-8</project.build.sourceencoding> <failonmissingwebxml>false</failonmissingwebxml> <jakartaee>8.0</jakartaee>	
<pre><dependencies></dependencies></pre>	
<dependency></dependency>	
<groupid>javax</groupid>	
<artifactid>javaee-api</artifactid>	
<version>\${jakartaee}</version>	
<scope>provided</scope>	
<dependency></dependency>	
<proupid>org.junit.jupiter</proupid>	

```
<artifactId>junit-jupiter</artifactId>
           <version>5.5.0</version>
           <scope>test</scope>
       </dependency>
       <dependency>
           <groupId>org.mockito</groupId>
           <artifactId>mockito-core</artifactId>
           <version>2.23.4</version>
           <scope>test</scope>
       </dependency>
   </dependencies>
    <build>
       <finalName>${project.artifactId}</finalName>
       <plugins>
           <plugin>
              <proupId>org.apache.maven.plugins</proupId>
               <artifactId>maven-compiler-plugin</artifactId>
               <version>3.8.1</version>
               <configuration>
                  <source>11</source>
                  <target>11</target>
               </configuration>
           </plugin>
           <plugin>
              <groupId>org.apache.maven.plugins</groupId>
               <artifactId>maven-war-plugin</artifactId>
               <version>3.2.3</version>
               <configuration>
                  <failOnMissingWebXml>false</failOnMissingWebXml>
               </configuration>
           </plugin>
       </plugins>
   </build>
</project>
```

⑤プロジェクトを実行する

バッキングビーンなどのクラスはありませんが、JSF プロジェクトとして実行できます。プロジェクトを 一度クリックしてから、ビルドボタン( デ)を押し、さらに、(Payara サーバーが起動していることを確 認して)実行ボタン( ▶・)を押してください。

次のようなウェブ画面が表示されます。



以上で、新規 JSF プロジェクトを作成して、実行できることがわかりました。後は、このファイルにクラ スやウェブページを追加すると、実際的な JSF プロジェクトを構築できます。

## 1. 付属している Payara Server での JDBC 設定は次の通りです。

JDBC 接続プール	DB 製品	DatabasName	ユーザー	パスワード	JDBC リソース名
		(スキーマ名)			
mydb_pool	MySQL	mydb	root	mysql8	jdbc/mydb
mydb_pg_pool PostgrSQL		mydb_pg	postgres	postgres	jdbc/mydb_pg
mydb_h2_pool	H2	mydb_h2	арр	арр	jdbc/mydb_h2

データベースを定義する際、Database 名(スキーマ名)、ユーザー名、パスワードを、これと違うもので 設定した場合は、Payara サーバーの管理コンソール(http://localhost:4848)の[JDBC 接続プール] 設 定で、それぞれの接続プールを選択し、[追加プロパティ] タブを開いて書き換えてください。

## 2.使用するデータベースは、Docker を使って準備します

具体的な方法は、Qiita ブログの「Jakarta EE+Payara server+NetBeans で開発しよう!(その2)」 を参照してください。

3.データベースを起動したら、テストとして、dbtest2 プロジェクトを開いて実行します 次のような入力を行うウェブが開きます。記入して、新規登録ボタンを押すと、EMPLOYEE というテーブル が作成されて、レコードが書き込まれます。

😡 dbtest2 - Apache NetBeans IDE 11	3
ファイル(F) 編集(E) 表示(V) ナビゲー	ト(N) ソース(S) リファクタリング(A) 実行(R) デパッグ(D) ブロファイル(P) チーム(M) ツール(T) ウ・
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	(デフォルト構成) ~ (② - 資 證) > - 45 100 + (③ - 100
プロジェクト × ファイル サービン	<b>λ</b> -
H W dhtest?	
	S dbtest
	$\leftarrow \rightarrow \mathbb{C} \ \bigcirc \ \bigcirc \ $ localhost:8080/dbte $\diamondsuit \ \blacksquare \ $
	データベースへの新規登録
Bbjava	番号
🗟 Employee java	氏名
💰 EmployeeDbjava 💰 LoggerProducerjava	メール
🛓 📰 ipjakartaee.config	新規登録
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□	
日本語その他のソース	
± 🛼 7⊡ジェクト・ファイル	
🗄 🕘 sample1	

結果は、NetBeans 上から確認できます。方法は、上記のブログに書いてあります。

(次ページの図を確認してください)



# Maven archetype "jakarta-ee-jsf" について

アーキタイプの作成と利用法については、別途、ブログ等でお知らせする予定です。ここでは、ある程度 の関連知識があることを前提に説明します。

コマンドプロンプトを起動し、**くドライブ>/jakarta/archetype/**jsf-maven-archetyp ディレクトリ にて、mvn clean install コマンドを実行すると、ローカルリポジトリに、jakarta-ee-jsf というアー キタイプが登録されます。

```
> w:
> cd jakarta¥archetype¥jsf-maven-archetyp
> mvn clean install
```

※実行するには、JDK と maven をインストールしておく必要があります

NetBeans から利用するには、[新規プロジェクト] ⇒ [新規プロジェクト] ⇒ [Maven] : [原型からのプロ ジェクト] を選択します。

🔾 新規プロジェクト		×
ステップ	プロジェクトを選択	
1. プロジェクトを選択 2	Q 7/109(T)	
	カテゴリ(C): Tロジェクト(P): Weich Constant	
	〕 説明(D)	
	Maven独自の原型プラヴインによって作成された、Mavenプロジェクトのテンプレート。	
	< 戻る(B) 次> 終了(F) 取消 ヘルプ(H	)

アーキタイプを選択する画面になるので、検索欄に、jakarta と入れると、jakarta-ee-jsf が先頭に表示されます。builde プロジェクトで生成されるものとほぼ同じプロジェクトが生成されます。

🕥 新規プロジェクト			×
ステップ	Maven原型		
1. プロジェクトを選択 2. Mayon 画型	検索(S) iskarta		🔲 より古いものを表示(0)
3. 名前と場所	既知の原型(T):		
	iskarta-ee-i5f         iskarta-ee-webapp         iskarta-ee-spplication-quickstart-archetype         iskartaee-application-quickstart-archetype         iskartaee-essentials-archetype         iskartaee-mp-archetype         iskartaee-mp-archetype         iskartaee-mp-archetype         iskartaee-mp-archetype         iskartaee-mp-archetype         iskartaee-mp-archetype         iskartaee-mp-archetype         iskartaee-mp-archetype         iskartaee-application-archetype         iskartaee8         iskartaee8         iskartaee7:cm2         j/u-7/D(G)       ip iskartaee         //-ジョン(/x)       1.0         ikBID(D)       intervention	アーティファウトIDC (): リポジトリ(R): local	
	jəkarta-ee-jsfərchetype		
	見>	<b>そる(B) 次 &gt;</b> 終了(F)	取)首 ヘルプ0-0

注意

ローカルリポジトリは、デフォルトでは、Cドライブの ユーザー/<ユーザー名>/.m2/repository にあ ります。mvn clean install を実行すると、この中に archetype-catalog.xmlが生成されて、ローカル リポジトリのアーキタイプとして認識されるようになります。