



本文書の位置づけ

◎ cfneo-Projectからのご提案

- cfneo-Projectの目的
= ソフトウェア・テストを何とかする
為の、cfneo-Projectからのご提案です
- この文書では、
cfneoの使い方について記述しています。
- この文書は、設計ガイドの内容を含みません。

用語

- ◎ 本体
 - cfneoとその配下のパッケージを指します。
- ◎ 接触篇
 - 「cfneodocs-001 ユーザーガイド(接触篇)」を指します。
- ◎ 設計ガイド
 - 「cfneodocs-003 設計ガイド」を指します。

cfneo概要

- ◎ cfneoは、ColdFusionで開発されたWebフレームワークです。
- ◎ cfneo-Projectのゴール
 - 「テストを何とかすること」

cfneo成果物構成

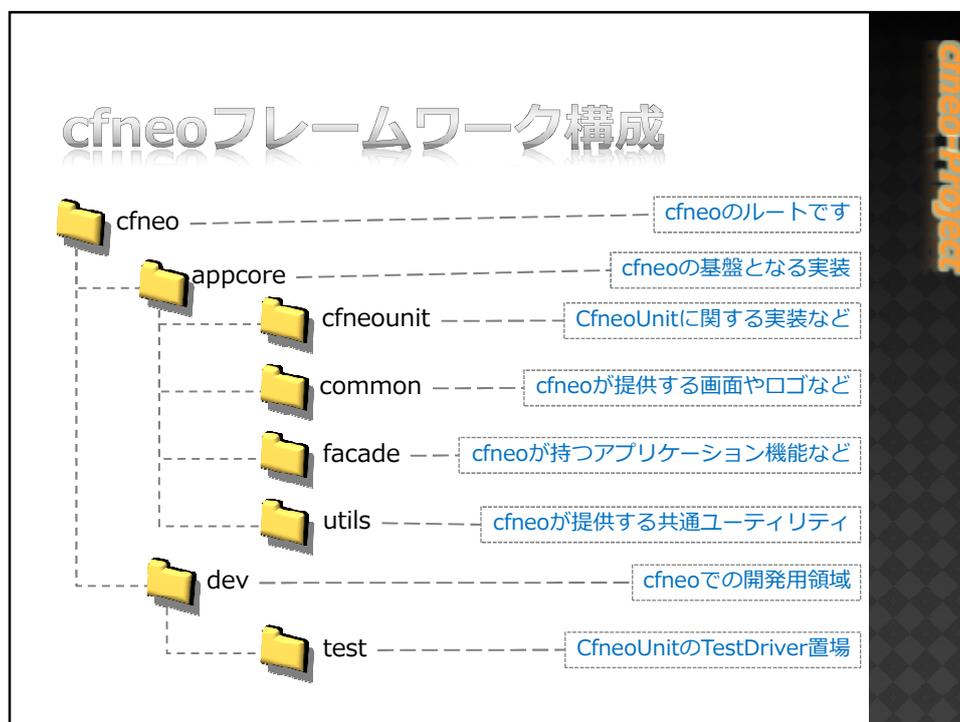
- ◎ 本体
 - cfneoとその配下のパッケージです。
- ◎ ユーザーガイド
 - 接触篇と発動篇があります。
- ◎ 設計ガイド
 - cfneoを使ったアプリケーション設計についてのガイドラインを記してあります。

cfneo Project

目次的なもの

- ◎ cfneoフレームワーク構成図
- ◎ Application Creator
- ◎ 簡易DI(ComponentManager)
- ◎ CfneoUnit

cfneo Project



Application Creator

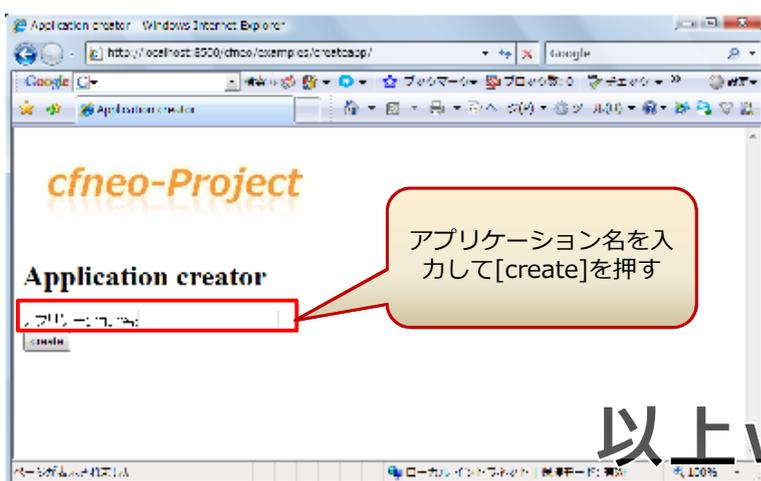
◎ Application Creatorとは…

- アプリケーション名 (APPLICATION.ApplicationName)が異なる cfneoクローンを生成する機能です。

■ cfneoクローンを生成する意味 ■

cfneoクローンを作る仕組みを作ったのは、ColdFusionの特性上、1つの開発マシンで開発を複数持てないためです。また、単純にコピーをただけだと継承元やアプリケーション名など、リテラルで定義しなくてはならない部分も、それぞれ直していきませんかといけません。これを一括して行うのが、Application Creatorです。

Application Creatorの使い方



以上w

Application Creator

- ◎ 指定されたアプリケーション名をルートに、以下のファイルをコピーします。

/appcore配下のすべて
/Application.cfc
/ApplicationProxy.cfc
/LICENSE.txt
/dev/test/Application.cfc
/dev/test/doCreateTestDriver.cfm
/dev/test/index.cfm
/dev/test/style.css

簡易DI

- ◎ 簡易DIとは、cfneoの管理下にあるCFコンポーネントの生成を行う為のコンポーネントComponentManagerを指します。

- ◎ コンポーネントComponentManager
 - ComponentManagerは、#getComponentのメソッドを提供します。
 - 引数：コンポーネント名
 - ex)/cfneo/appcore/ApplicationProxy.cfc
→"appcore.ApplicationProxy"と指定
 - 戻り値：生成したComponentObject

簡易DI

- ◎ cfmからの利用方法

```
<cfset VARIABLES.obj=getComponentManager().  
getComponent("ComponentName") />
```

ComponentManagerは、cfneoが提供するApplication.cfcのメソッド#getComponentManagerを用いて取得可能です。

簡易DI

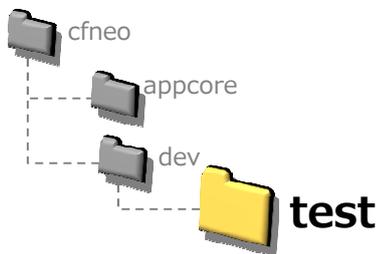
- ◎ cfcからの利用方法

```
<cfset VARIABLES.obj=THIS.comMgr.  
    getComponentManager().  
    getComponent("ComponentName") />
```

ComponentManagerを使用して生成されたCFコンポーネントの場合、comMgrプロパティにComponentManagerをそのままセットしています。

CfneoUnit

- ◎ CfneoUnitはcfneo上でxUnitを行う為のプラットフォームです。
 - /APPLICATION-ROOT/dev/test/の配下で利用可能



CfneoUnitの使い方

◎ Test Driver List

- TestDriverの作成状況を一覧で表示します
 - アプリケーション・ルートからcfcファイルを探し出し、それぞれにTestDriver(Hoge.cfcならHogeTest.cfm)が作られているかどうか、を確認できます。
 - TestDriverが存在した場合、対象となるcfcファイルのメソッド(publicのもののみ)がTestDriverの中でテストが書かれているかどうか、を確認できます。

※) TestDriverの中にメソッドのテストが書かれているかどうかは、以下の書式に沿ってテストが記述されているか、を元に判定します。

<fieldset>

<legend>メソッド名</legend>

</fieldset>

CfneoUnitの使い方

◎ Test Driver List

The screenshot shows a web browser window titled 'Test driver list - Windows Internet Explorer'. The address bar shows 'http://localhost:8900/cfneo/cew/test/'. The page content includes the 'cfneo-Project' logo, two buttons ('Show application Context' and 'Error Code List'), and a 'Test driver list' section. Below this, there is a 'Back' link and a table titled 'Target[37] / Test driver[30]'. The table has two columns: 'File name' and 'Link'. The rows show various application and temporary cfc files, each with a 'Create Test driver file' link. At the bottom of the table, there is a 'Go test driver' link.

File name	Link
cfneo\appsrc\application.cfc	Create Test driver file.
cfneo\appsrc\application\temporary.cfc	Create Test driver file.
cfneo\appsrc\dic\com\application.cfc	Create Test driver file.
cfneo\appsrc\dic\com\application\temporary.cfc	Create Test driver file.
	Go test driver

CfneoUnitの使い方

- ◎ 「Create Test drive file.」のリンクを押した場合
 - 空のTestDriverが生成されます。

■ TestDriverの命名規則の例 ■

テスト対象のCFコンポーネントが
[cfneo.appcore.ComponentManager]
(物理パス=ROOT/cfneo/appcore/ComponentManager.cfc)
の場合、そのTestDriverは、
[/cfneo/dev/test/appcore/ComponentManagerTest.cfm]
という名称で作成します。

CfneoUnitの使い方

- ◎ TestDriverの使い方
 - 生成直後のTestDriverはテンプレートのままです。
 - このテンプレートにテストを記述していきます。

```
<cfset in=12345 />  
<cfset exp=true />  
<cfset act=IsNumeric(in) />  
<cfset assert(1, "invoke", in, exp, act) />
```

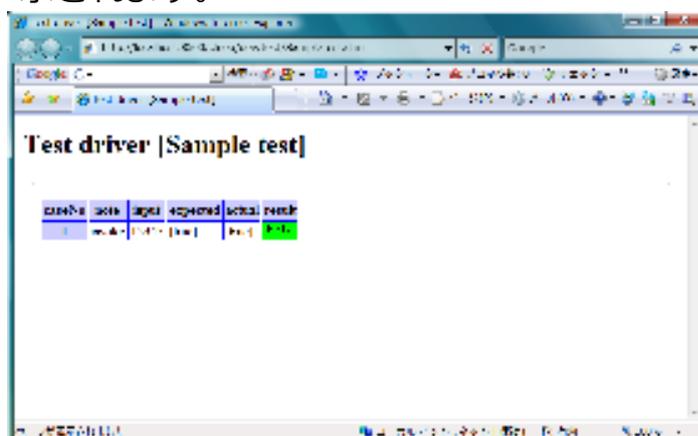
CfneoUnitの使い方

- ◎ テストを実装する。
 - TestDriver内では、VARIABLESスコープのプリフィクスは割愛しています。
 - 1つのファイルに対して、1つのTestDriverを実装します。

cfneo-Project

CfneoUnitの使い方

- ◎ TestDriverにアクセスすると以下のように表示されます。



The screenshot shows a web browser window with the title 'Test driver [Sample test]'. The browser's address bar shows 'http://localhost:8080/'. The main content area displays a table with the following structure:

index	code	target	expected	actual	result
1	test	100	100	100	pass

cfneo-Project