



ColdSpring

The Basics

Ted Steinmann

- Lead Developer
 - Hospital Dev Team
 - ImageTrend, Inc.
- CF User 6 years
- tedsteinmann@gmail.com
- <http://www.linkedin.com/in/tedsteinmann>

Questions

- Working on ColdSpring driven apps?
- Built/implemented ColdSpring?
- Not using ColdSpring?
- Using something else?
- Not Using components/cfcs?

What Does ColdSpring Do?

- Object/Bean (Component) Factory
 - Singleton
- Manage Dependencies
- Much, much more...

ColdFusion Component

- .cfc File containing data & functions
- Building blocks to CF in an OO manner
 - Encapsulation
 - Inheritance
 - Introspection

What's a Singleton?

- Design pattern
 - Single instance
 - Global access
 - One object
 - No State

```
application.cfc X
1 <cfcomponent output="false">
2     <!--- insert Ben Nadel Joke here --->
3     <cffunction name="onApplicationStart"
4         output="false"
5         hint="I fire once upon application startup">
6
7         <cfset application.userService =
8             createObject("component","com.services.UserService") />
9
10    </cffunction>
11
12 </cfcomponent>
```

What is a Transient Object?

- Container
 - Short lived
 - Not Persisted

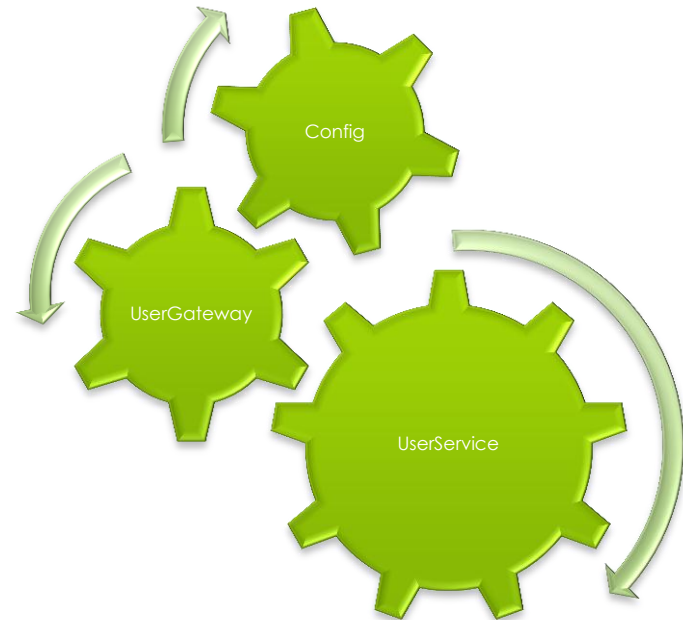
```
new-user.cfm X
1 <cfset user = CreateObject("component","com.user") />
2
3 <cfset user.setFirstName("Ted") />
4 <cfset user.setLastName("Steinmann") />
5
6 <cfoutput>
7     User Name:
8     #user.getFirstName()#
9     #user.getLastName()#
10 </cfoutput>
```

Example (user list)

First Name	Last Name	User Id	Password
Ted	Steinmann	tsteinmann	welcome
Emily	Christiansen	echristiansen	welcome
Tim	Meyer	tmeyer	welcome
Kurt	Wiersma	kwiersma	welcome

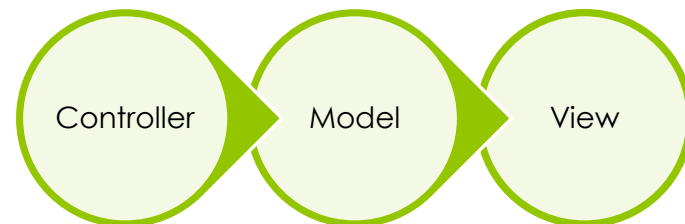
Dependency

- One piece of a program **depends** on another to get it's job done.
- a state in which one object uses a functionality of another object



Dependency

- MVC/SOA
 - Dependencies between layers
 - Dependencies between objects



What does that give us?

- Prevents duplication of code/config
- Separation of Concerns
- Modularity
- Encapsulation

Questions?

Configuration



BeanFactory

- Container or holder for components
 - Instantiate
 - Configure
 - Resolve Dependencies



Resolve Dependencies

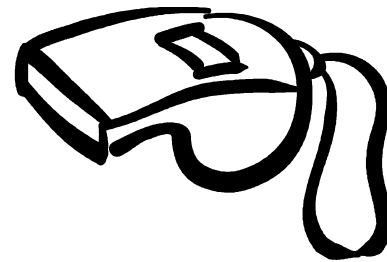
- Constructor Arguments
- Setter Injection

Before moving on...

- Questions on:
 - Dependencies?
 - How ColdSpring injects dependencies?

Additional functionality

- Import XML directive
- Dynamic Properties



Others

- Parent Beans
- Factory Beans
- AOP
- Remote proxies
- Mock Objects
- Factory-post-processor



Benefits of DI/IOC

- Separation of concerns
- Less coupling
- Configuration without changing code
- Components become easier to test
- Birds eye view of the dependencies
- Components are not tied to ColdSpring

Cost of DI/IOC

- Configuration required
- Longer startup times
- Other?

References

- ColdSpring
 - Quickstart
 - Reference Guide
 - What's new in 1.2
 - ColdSpring Lite
 - Annotation-based Dependency Injection using ColdSpring
- Other
 - Dependency Injection (Martin Fowler)
 - ColdFusion Components
 - Derby DB
 - Environment Config
 - Object Oriented Programming

Bonus



Constructor Args

```
UserService.cfc
cfml html js css
try # <!-- /% ' , ,, AB ab
1 component {
2
3     public any function init(required any userGateway) {
4         variables.userGateway = arguments.userGateway;
5         return this;
6     }
7
8 }
9
```

Setter Injection

```
UserService.cfc
cfml html js css
try # <!-- /% ' , ,, AB ab
1 component accessors="true" {
2
3     property userGateway;
4
5 }
```