Introduction to Grails

Jeff Palmer

jeff.palmer@objectpartners.com

@jpalmer1026

Object Partners Inc

- Java, Groovy, JavaScript, Mobile, Open Source
- ~100 Senior Consultants
  - Minneapolis, Omaha
  - Chicago, Denver
  - Average Tenure Over 5 years
- Founded 1996

Grails

- Open source
- Full stack
  - Spring
  - Hibernate
  - SiteMesh
  - Tomcat
  - H2 database

- Convention over configuration
- Over 1000 plugins
- Large and active community

Companies using Grails

With Groovy and Grails we can create a new feature in a week, when before it could easily take a month or more

— Jon Mullen, Sky ScrumMaster

Our developers are much happier developing in Grails because they can accomplish tasks so much faster

— Paul Fisher, Tech Mgr Wired.com

Resources

Grails Homepage (https://grails.org)

User Guide (http://grails.org/doc/latest/)

Gr8Conf Videos (https://www.youtube.com/channel/UC7wUp2KIa1hoMNn0r7JUVEg)

Grails Diary (http://grydeske.net/news/show/55)

Stack Overflow

Books

Grails in Action

Programming Grails

The Definitive Guide to Grails 2

Grails 2: A Quick-Start Guide

Groovy

- Lightweight
  - optional return statements
  - optional semicolons
  - methods and classes public default
  - optional return statements
- Dynamic (can be extended at runtime)
  - responsibility for the flexibility of Grails
- Interoperable with Java

Java

```
public class Car {
    private int miles;
    private final int year;

    public Car(int theYear) { year = theYear; }
    public int getMiles() { return miles; }
    public void setMiles(int theMiles) { miles = theMiles; }
    public int getYear() { return year; }
}
```

Groovy

```
class Car {
    def miles = 0
    final year // generates getter but no setter

    Car(theYear) { year = theYear }
}
```

Lists

```
def myList = [1, 2, 3, 4, 5, 6, 7, 8] // java.util.ArrayList

myList.each { println it }

def doubled = []
myList.each { doubled << it * 2 } // [2, 4, 6, 8, 10, 12, 14, 16]

myList.collect { it * 3 } // [3, 6, 9, 12, 15, 18, 21, 24]

myList.findAll { it > 4 } // [5, 6, 7, 8]
```

Maps

```
def frameworks = [grails : 'groovy', play : 'scala', rails : 'ruby'] // java.util.LinkedHashMap

println frameworks['grails'] // groovy
println frameworks.play // scala

frameworks.each { element -> // MapEntry
    println "The $element.key framework uses language $element.value"
}

frameworks.each { framework, language ->
    println "The $framework framework uses language $language"
}
```

Demo