

Play Framework

A look inside the machine room

Play! 

Alexander Reelsen

alexander@reelsen.net

[@spinscale](#)

Inside the machine room

- Model:
 - Property Enhancement
 - Finders
- Controller:
 - Redirects
 - Rendering context

About me - Alexander Reelsen

- Studied information systems
- 10 years linux system engineering, converted to software engineering
- Web framework enthusiast, fed up with complex java environment for simple webapps
- Other interests: Scaling web architectures, Web 2.0 (nosql, search)
- Started with play 1.0 in 2009, started writing cookbook in dec 2010
- Streetball/Basketball

Code sample - Entity

```
@Entity public User extends Model {  
    public String name;  
    public String password;  
    public void setPassword(String pwd) {  
        password = Crypto.passwordHash(pwd);  
    }  
    public boolean hasPassword(String pwd) {  
        return Crypto.passwordHash(pwd).equals(password);  
    }  
}
```

Code sample - Controller

```
public static void setPassword(String nw, String old) {  
    User user = User.findById((Long)session.get("id"));  
    notFoundIfNull(user);  
    if (!user.hasPassword(old)) {  
        index();  
    }  
    user.password = nw;  
    user.save();  
    render(user);  
}
```

05

}

Property enhancement

- What happens in `user.password = nw;`?
- According to the documentation: The setter is called - but how?
- Solution:
`play.classloading.enhancers.PropertiesEnhancer`
- Performing bytecode enhancement via javassist library
- Enhancer loops over all fields, checks if getters/setters exist, creates them if necessary
- Next step: Intercept all field accesses and redirect to getters/setters
- Similar: [Project Lombok](#), Scala, many other languages

Example getter/setter creation with javassist

```
String code =  
    "public String getName() { return this.name; }";  
CtMethod getMethod = CtMethod.make(code, ctClass);  
ctClass.addMethod(getMethod);  
  
code = "public void setName(String value) " +  
    "{ this.name = value; }";  
CtMethod setMethod = CtMethod.make(code, ctClass);  
ctClass.addMethod(setMethod);
```

Intercept field access, replace with getter/setter

- `ExprEditor` from javassist is used, can be used to replace calls inside a method body, instead of replacing the method itself
- Intercepts field access, no matter if in model, controller or template
- Invokes either `FieldAccessor.invokeReadProperty()` or `FieldAccessor.invokeWriteProperty()`, which calls either getter or setter



OffbizImporter.java User.java Application.java routes AddDataJob.java index.html setPassword.html *Application2.java

```
37 }
38
39 public static void login(String username, String password) throws Throwable {
40     Object localObject2;
41     try {
42         LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.enter();
43         LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.addVariable("password", password);
44         LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.addVariable("username", username);
45         if (!ControllersEnhancer.ControllerInstrumentation.isActionCallAllowed()) {
46             Controller.redirect("controllers.Application.login", new Object[] { username, password });
47         } else {
48             ControllersEnhancer.ControllerInstrumentation.stopActionCall();
49             User user = (User)User.find("byNameAndPassword", new Object[] { username, Crypto.passwordHash(password) }).first();
50             LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.addVariable("user", user);
51             notFoundIfNull(user);
52             Object localObject1 = null;
53             Scope.Session localSession = null;
54             localSession = (Scope.Session)Java.invokeStatic(Desc.getType("Lplay/mvc/Scope$Session;"), "current");
55             User localUser1 = user;
56             Lona localLona = null;
57             localLong = (Long)PropertiesEnhancer.FieldAccessor.invokeReadProperty(localUser1, "id", "models.User", "controllers.Application.login, line 20");
58             localSession.put("id", localLong);
59             index();
60         }
61     } finally {
62         localObject2 = null;
63         LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.exit();
64     }
65 }
66
67 public static void setPassword(String nw, String old) throws Throwable {
68     Object localObject2;
69     try {
70         LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.enter();
71         LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.addVariable("old", old);
72         LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.addVariable("nw", nw);
73         if (!ControllersEnhancer.ControllerInstrumentation.isActionCallAllowed()) {
74             Controller.redirect("controllers.Application.setPassword", new Object[] { nw, old });
75         } else {
76             ControllersEnhancer.ControllerInstrumentation.stopActionCall();
77             Object localObject1 = null;
78             Scope.Session localSession = null;
79             localSession = (Scope.Session)Java.invokeStatic(Desc.getType("Lplay/mvc/Scope$Session;"), "current");
80             User user = (User)User.findById(Long.valueOf(localSession.get("id")));
81             LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.addVariable("user", user);
82             notFoundIfNull(user);
83             if (!user.hasPassword(old)) {
84                 index();
85             }
86             String str = nw;
87             User localUser1 = user;
88             PropertiesEnhancer.FieldAccessor.invokeWriteProperty(localUser1, "password", String.class, str, "models.User", "controllers.Application.setPassword, line 30");
89             user.save();
90             render(new Object[] { user });
91         }
92     } finally {
93         localObject2 = null; LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.exit();
94     }
95 }
96 }
```

Finders in models

- How does `User.find("byName", "alexander");` work?
- Core problem: `Model.find()` is a static method
- Nothing is overwritten in the User entity
- Hierarchy: `User` > `Model` > `GenericModel` > `JPABase`
- GenericModel:

```
public static JPAQuery find(String query, Object... params)
    throw new UnsupportedOperationException("Please
        annotate your JPA model...");
```

}
10

play.db.jpa.JPAEnhancer

Enhancer loops through all entities and adds static methods

```
// public static play.db.jpa.GenericModel.JPAQuery  
// find(String query, Object[] params) {  
//     return play.db.jpa.JPQL.instance.find(entityName,  
//     query, params);  
// };  
CtMethod find = CtMethod.make(code, ctClass);  
ctClass.addMethod(find);
```

play.db.jpa.JPAEnhancer

- Similar code applies to other static methods
- `count()`, `count(query, params)`
- `findAll()`, `findById(id)`, `find(query, params)`,
`findOneBy(query, params)`
- `deleteAll()`, `delete(query, params)`
- `create()`

Deep dive: Disassembling bytecode

- Problem: Debugging is hard and sucks
- Back to the compile, restart cycle
- You never have exact info how bytecode was applied
- Solution: `javap -verbose -c User`
- Better solution: [JD-Gui](#)

- ▶ controllers
- ▶ exceptions
- ▶ helpers
- ▼ models
 - ▶ CategoryList
 - ▶ ProductSearchRequest
 - ▼ User
 - User
 - name : String
 - all() : GenericModel.JPAQuery
 - count() : long
 - count(String, Object[]) : long
 - create(String, Scope.Params) : JPABase
 - delete(String, Object[]) : int
 - deleteAll() : int
 - find() : GenericModel.JPAQuery
 - find(String, Object[]) : GenericModel.JF
 - findAll() : List
 - findById(Object) : JPABase
 - findOneBy(String, Object[]) : JPABase
 - getName() : String
 - setName(String) : void
- ▶ plugins
- ▶ DocViewerPlugin

User.class x

```

package models;

import java.util.List;

@Entity
public class User extends Model
{
    public String name;

    public String getName()
    {
        return this.jdField_name_of_type_JavaLangString;
    }

    @PropertiesEnhancer.PlayPropertyAccessor
    public void setName(String paramString)
    {
        this.jdField_name_of_type_JavaLangString = paramString;
    }

    public static long count()
    {
        return JPQL.jdField_instance_of_type_PlayDbJpaJPQL.count("models.User");
    }

    public static long count(String paramString, Object[] paramArrayOfObject)
    {
        return JPQL.jdField_instance_of_type_PlayDbJpaJPQL.count("models.User", paramString, paramArrayOfObject);
    }

    public static List findAll()
    {
        return JPQL.jdField_instance_of_type_PlayDbJpaJPQL.findAll("models.User");
    }

    public static JPABase findById(Object paramObject)
    {
        return JPQL.jdField_instance_of_type_PlayDbJpaJPQL.findById("models.User", paramObject);
    }

    public static GenericModel.JPAQuery find(String paramString, Object[] paramArrayOfObject)
    {
        return JPQL.jdField_instance_of_type_PlayDbJpaJPQL.find("models.User", paramString, paramArrayOfObject);
    }

    public static GenericModel.JPAQuery find()
    {
        return JPQL.jdField_instance_of_type_PlayDbJpaJPQL.find("models.User");
    }

    public static GenericModel.JPAQuery all()
    {
        return JPQL.jdField_instance_of_type_PlayDbJpaJPQL.all("models.User");
    }
}

```

Controller redirects

Calling another controller in java issues a HTTP redirect

```
public static void redirect() {  
    index();  
}
```

```
curl -v localhost:9000/redirect  
< HTTP/1.1 302 Found  
< Location: http://localhost:9000/
```

Controller redirects - logic

- `play.classloading.enhancers.ControllersEnhancer`
- Every action is enhanced
- As soon as an action is started, a boolean is set
- On every action invocation this boolean is checked
- If it is already true, this means the action is called from another java method and a redirect is issued
- If it is false, the controller logic is executed

Controller redirects - bytecode

Inserted before every controller call

```
if(!play.classloading.enhancers.ControllersEnhancer.  
  ControllerInstrumentation.isActionCallAllowed()) {  
  play.mvc.Controller.redirect(  
    "controllers.Application.index", $args);  
  return;  
}  
play.classloading.enhancers.ControllersEnhancer.  
  ControllerInstrumentation.stopActionCall();
```

Disassembled controller code - is different

```
if (!ControllersEnhancer.  
    ControllerInstrumentation.isActionCallAllowed()) {  
    Controller.redirect("controllers.Application.index",  
        new Object[0]);  
} else {  
    ControllersEnhancer.ControllerInstrumentation  
        .stopActionCall();  
    // controller logic comes here...  
}
```



OffbizImporter.java User.java Application.java routes AddDataJob.java index.html setPassword.html *Application2.java

```
37 }
38
39 public static void login(String username, String password) throws Throwable {
40     Object localObject2;
41     try {
42         LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.enter();
43         LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.addVariable("password", password);
44         LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.addVariable("username", username);
45         if (!ControllersEnhancer.ControllerInstrumentation.isActionCallAllowed()) {
46             Controller.redirect("controllers.Application.login", new Object[] { username, password });
47         } else {
48             ControllersEnhancer.ControllerInstrumentation.stopActionCall();
49             User user = (User)User.find("byNameAndPassword", new Object[] { username, Crypto.passwordHash(password) }).first();
50             LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.addVariable("user", user);
51             notFoundIfNull(user);
52             Object localObject1 = null;
53             Scope.Session localSession = null;
54             localSession = (Scope.Session)Java.invokeStatic(Desc.getType("Lplay/mvc/Scope$Session;"), "current");
55             User localUser1 = user;
56             Long localLong = null;
57             localLong = (Long)PropertiesEnhancer.FieldAccessor.invokeReadProperty(localUser1, "id", "models.User", "controllers.Application.login, line 20");
58             localSession.put("id", localLong);
59             index();
60         }
61     } finally {
62         localObject2 = null;
63         LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.exit();
64     }
65 }
66
67 public static void setPassword(String nw, String old) throws Throwable {
68     Object localObject2;
69     try {
70         LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.enter();
71         LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.addVariable("old", old);
72         LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.addVariable("nw", nw);
73         if (!ControllersEnhancer.ControllerInstrumentation.isActionCallAllowed()) {
74             Controller.redirect("controllers.Application.setPassword", new Object[] { nw, old });
75         } else {
76             ControllersEnhancer.ControllerInstrumentation.stopActionCall();
77             Object localObject1 = null;
78             Scope.Session localSession = null;
79             localSession = (Scope.Session)Java.invokeStatic(Desc.getType("Lplay/mvc/Scope$Session;"), "current");
80             User user = (User)User.findById(Long.valueOf(localSession.get("id")));
81             LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.addVariable("user", user);
82             notFoundIfNull(user);
83             if (!user.hasPassword(old)) {
84                 index();
85             }
86             String str = nw;
87             User localUser1 = user;
88             PropertiesEnhancer.FieldAccessor.invokeWriteProperty(localUser1, "password", String.class, str, "models.User", "controllers.Application.setPassword, line 30");
89             user.save();
90             render(new Object[] { user });
91         }
92     } finally {
93         localObject2 = null; LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.exit();
94     }
95 }
96 }
```

Rendering context

How does this work?

```
List<User> users = User.findAll();  
String message = "FooBarBaz";  
render( users , message );
```

```
<div> ${message} </div>  
#{list users , as:'user'}  
<li>${user.name}</li>  
#{/list}
```

Rendering context - how it works

`play.classloading.enhancers.LocalVariablesNamesEnhancer`

- Every variable declaration in an action creates a sort of history entry in `LocalVariablesNamesTracer.localVariables`
- When rendering the template every object added to the `render()` method is checked with this `localVariables` variable
- Example:

```
List<User> users = User.findAll();  
LocalVariablesNamesTracer.addVariable("users", users);
```

Rendering context - how it works

```
protected static void renderTemplate(String templateName,  
Object... args) {  
    Map templateBinding = new HashMap(16);  
    for (Object o : args) {  
        List names =  
            LocalVariablesNamesTracer.getAllLocalVariableNames(o);  
        for (String name : names) {  
            templateBinding.put(name, o);  
        }  
    }  
    renderTemplate(templateName, templateBinding);  
}
```



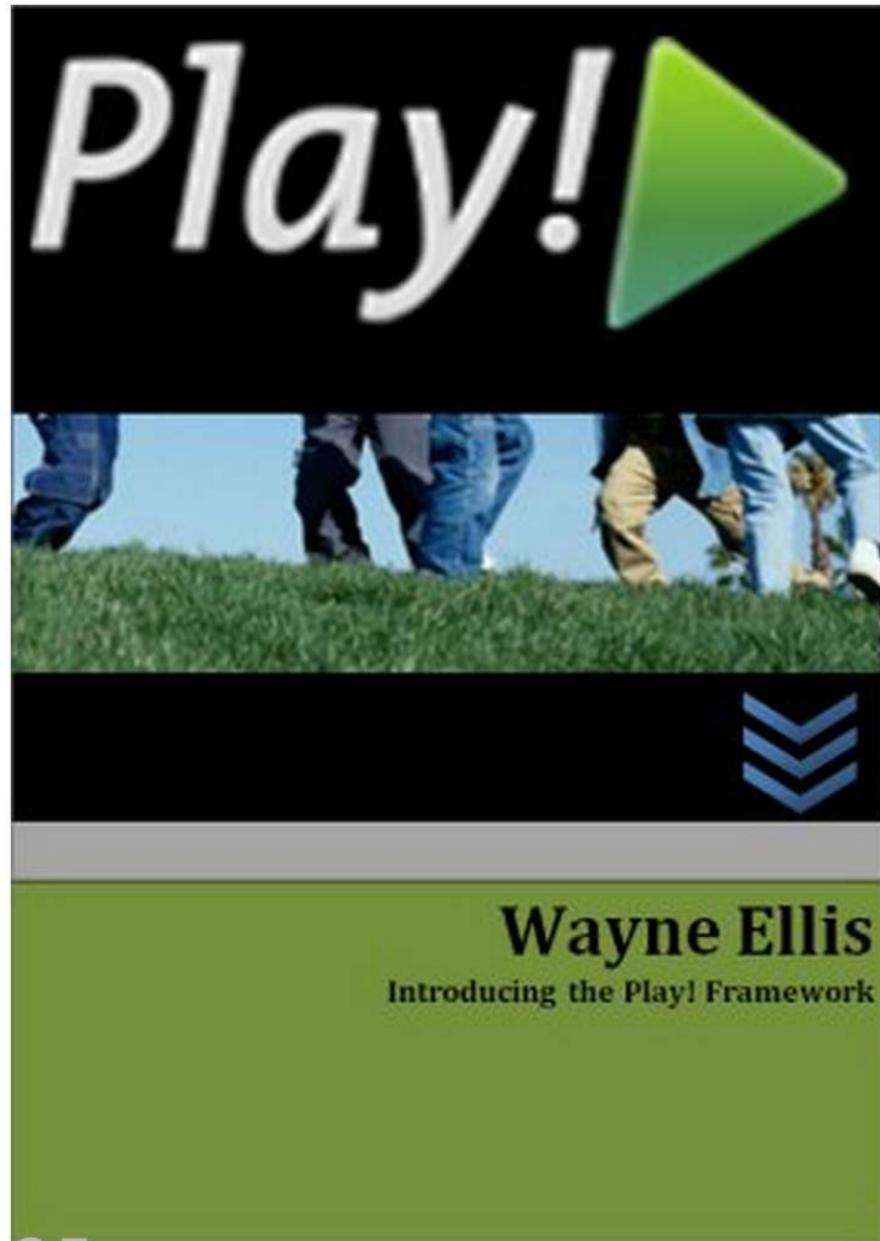
OffbizImporter.java User.java Application.java routes AddDataJob.java index.html setPassword.html *Application2.java

```
37 }
38
39 public static void login(String username, String password) throws Throwable {
40     Object localObject2;
41     try {
42         LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.enter();
43         LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.addVariable("password", password);
44         LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.addVariable("username", username);
45         if (!ControllersEnhancer.ControllerInstrumentation.isActionCallAllowed()) {
46             Controller.redirect("controllers.Application.login", new Object[] { username, password });
47         } else {
48             ControllersEnhancer.ControllerInstrumentation.stopActionCall();
49             User user = (User)User.find("byNameAndPassword", new Object[] { username, Crypto.passwordHash(password) }).first();
50             LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.addVariable("user", user);
51             notFoundIfNull(user);
52             Object localObject1 = null;
53             Scope.Session localSession = null;
54             localSession = (Scope.Session)Java.invokeStatic(Desc.getType("Lplay/mvc/Scope$Session;"), "current");
55             User localUser1 = user;
56             Long localLong = null;
57             localLong = (Long)PropertiesEnhancer.FieldAccessor.invokeReadProperty(localUser1, "id", "models.User", "controllers.Application.login, line 20");
58             localSession.put("id", localLong);
59             index();
60         }
61     } finally {
62         localObject2 = null;
63         LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.exit();
64     }
65 }
66
67 public static void setPassword(String nw, String old) throws Throwable {
68     Object localObject2;
69     try {
70         LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.enter();
71         LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.addVariable("old", old);
72         LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.addVariable("nw", nw);
73         if (!ControllersEnhancer.ControllerInstrumentation.isActionCallAllowed()) {
74             Controller.redirect("controllers.Application.setPassword", new Object[] { nw, old });
75         } else {
76             ControllersEnhancer.ControllerInstrumentation.stopActionCall();
77             Object localObject1 = null;
78             Scope.Session localSession = null;
79             localSession = (Scope.Session)Java.invokeStatic(Desc.getType("Lplay/mvc/Scope$Session;"), "current");
80             User user = (User)User.findById(Long.valueOf(localSession.get("id")));
81             LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.addVariable("user", user);
82             notFoundIfNull(user);
83             if (!user.hasPassword(old)) {
84                 index();
85             }
86             String str = nw;
87             User localUser1 = user;
88             PropertiesEnhancer.FieldAccessor.invokeWriteProperty(localUser1, "password", String.class, str, "models.User", "controllers.Application.setPassword, line 30");
89             user.save();
90             render(new Object[] { user });
91         }
92     } finally {
93         localObject2 = null; LocalVariablesNamesEnhancer.LocalVariablesNamesTracer.exit();
94     }
95 }
96 }
```

Questions?



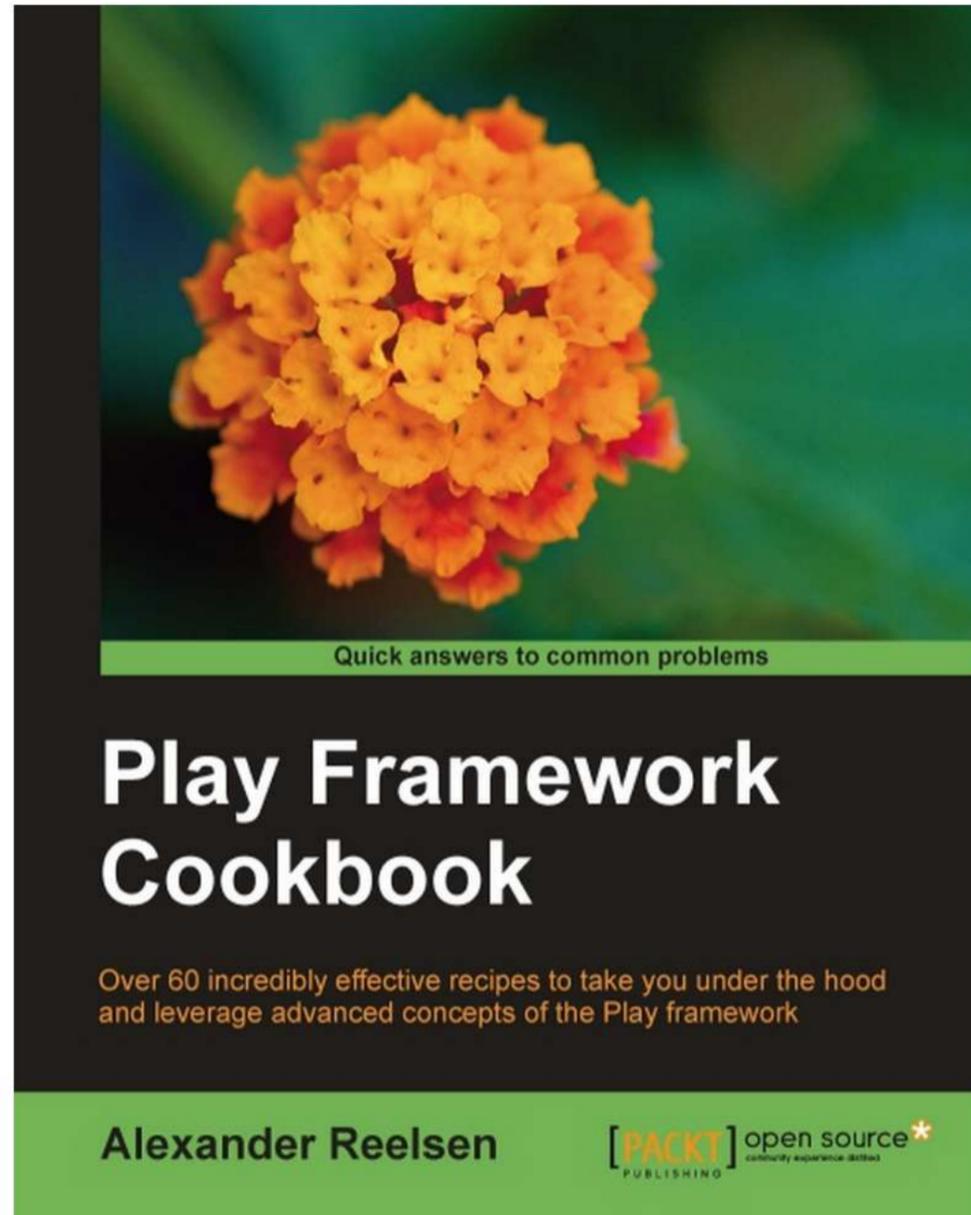
Books: Introducing the Play! framework



[Introducing the Play! framework](#)

by Wayne Ellis

Books: Play Framework Cookbook



[Play Framework Cookbook](#)

by Alexander Reelsen

Resources



- [Playframework](#) website, [@playframework](#)
- [Introductory presentation](#) at Java User Group Munich in June 2010
- [Introductory presentation](#) at TNG Technologies in Oct 2011
- [A look inside the machine room](#): Play!ground Presentation at Lunatech in Rotterdam in Sep 2011
- [Play framework cookbook](#) (example sources on [github](#))
- Presentation has been done with [shower](#)