Wicket 6

Jochen Mader

- Chief Developer @ Senacor Technologies AG
- http://www.senacor.com
- jochen.mader@senacor.com
- Twitter: @codepitbull

?

Quickstart

APACHE WICKET

Meet Wicket

Home
Introduction
Features
Buzz
Vision
Blogs

Get Started

Download Wicket
Quickstart
More archetypes
Get help
Mailing Lists

Welcome to Apache Wicket

With proper mark-up/logic separation, a POJO data mo
makes developing web-apps simple and enjoyable aga
brittle code for powerful, reusable components written v

- Check the feature list
- Read some Wicket buzz, some Wicket blogs
- Find out why you should use Wicket
- Check out some selected examples in detail or see th
- Take a look at our live component reference
- Go and download Wicket
- See what extras are available

Wicket is released under the Apache License, Version 2.

Apache Wicket 6.6.0 released

Creating the project - with Maven

Base Package ur project, copy and paste the command line generated after typing in the groupId, artifactId and version.

GroupId: com.mycompany (?)

ArtifactId: myproject (?)

Version: 6.6.0 (?)

Command Line:

```
mvn archetype:generate -DarchetypeGroupId=org.apache.wicket -
DarchetypeArtifactId=wicket-archetype-quickstart -
DarchetypeVersion=6.6.0 -DgroupId=com.mycompany -
DartifactId=myproject -
DarchetypeRepository=https://repository.apache.org/ -
DinteractiveMode=false
```

Auf geht‘s

Unmanaged aber nicht einsam

Unmanaged aber nicht einsam

Managed

Unmanaged aber nicht einsam

Managed

Glue

Weblayer

ORM

Management

Security

...

Wicket

Weblayer

API

Aufbau


Servlet 2.5 Container
<application>.war
WicketFilter
WebApplication

```java
public class WicketApplication extends AuthenticatedWebApplication {
    @Override
    public HomePage getHomePage() {
        return HomePage.class;
    }

    @Override
    public void init() {
        super.init();
        getSecuritySettings().setAuthorizationStrategy(
            new AnnotationsRoleAuthorizationStrategy(this));
        getMarkupSettings().setAutomaticLinking(true);
        mountPage("/dummy", PlainLayoutPage.class);
    }

    @Override
    protected Class<? extends AbstractAuthenticatedWebSession> getWebSessionClass() {
        return UserAuthenticatedWebSession.class;
    }

    @Override
    protected Class<? extends WebPage> getSignInPageClass() {
        return SignInWithoutRememberMePage.class;
    }
}
```

enabling component-oriented, programmatic manipulation of markup*

*http://wicket.apache.org/meet/vision.html

Good old Java

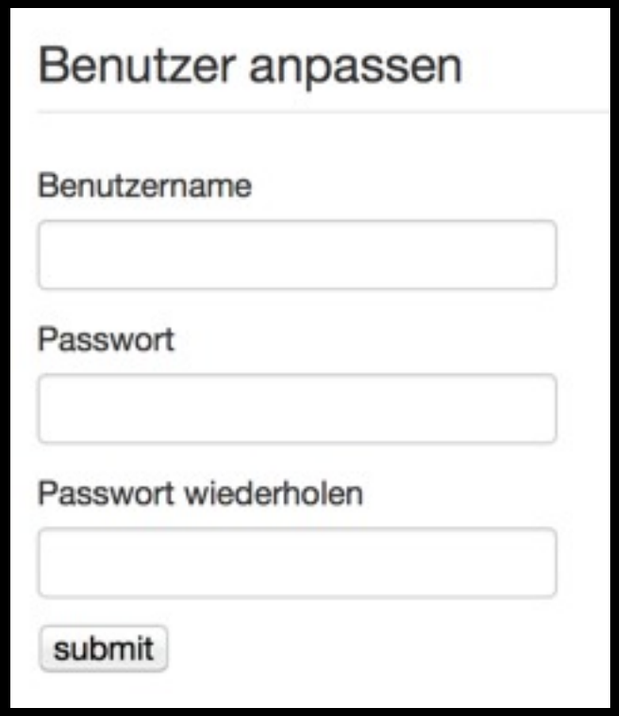
+ HTML

Component

Component.java

Component.html

Component.properties



UserUpdatePanel.java

```java
public class UserUpdatePanel extends Panel{
    public static final String PASSWORD = "password";
    public static final String PASSWORD_REPEAT = "passwordRepeat";
    public static final String USER_UPDATE_FORM = "userUpdateForm";

    @Autowired
    private UserRepository userRepository;

    public UserUpdatePanel(String id, IModel<User> model) {
        super(id, model);

        FormComponent<String> password = new PasswordTextField(PASSWORD);
        FormComponent<String> passwordRepeat = new PasswordTextField(PASSWORD_REPEAT, Model.of(""));

        Form<User> userUpdateForm =
                new Form<User>(USER_UPDATE_FORM, new CompoundPropertyModel<User>(model)) {
                    @Override
                    protected void onSubmit() {
                        userRepository.save(getModelObject());
                    }
                };

        add(userUpdateForm
                .add(new TextField<String>("name"))
                .add(password)
                .add(passwordRepeat)
                .add(new AjaxFallbackButton("submit", userUpdateForm) {})
                .add(new EqualInputValidator(password, passwordRepeat))
        );
    }
}
```

UserUpdatePanel.html

```html
<!DOCTYPE html>
<html xmlns:wicket="http://wicket.apache.org">
    <head>
        <meta charset="utf-8">
        <title>Wicket Example App</title>
    </head>
    <body>
        <wicket:panel>
            <form wicket:id="userUpdateForm">
                <fieldset>
                    <legend><wicket:message key="userUpdateTitle"/></legend>
                    <label wicket:for="name"><wicket:message key="username"/></label>
                    <input type="text" wicket:id="name"/>
                    <label wicket:for="password"><wicket:message key="password"/></label>
                    <input type="password" wicket:id="password"/>
                    <label wicket:for="passwordRepeat"><wicket:message key="passwordRepeat"/></label>
                    <input type="password" wicket:id="passwordRepeat"/>
                    <p><input type="submit" value="submit" wicket:id="submit"/></p>
                </fieldset>
            </form>
        </wicket:panel>
    </body>
</html>
```

UserUpdatePanel.properties

```
userUpdateTitle=Benutzer anpassen
username=Benutzername
password=Passwort
passwordRepeat=Passwort wiederholen
```

```java
public class UserUpdatePanel extends Panel{
    public static final String PASSWORD = "password";
    public static final String PASSWORD_REPEAT = "passwordRepeat";
    public static final String USER_UPDATE_FORM = "userUpdateForm";

    @Autowired
    private UserRepository userRepository;

    public UserUpdatePanel(String id, IModel<User> model) {
        super(id, model);

        FormComponent<String> password = new PasswordTextField(PASSWORD);
        FormComponent<String> passwordRepeat = new PasswordTextField(PASSWORD_REPEAT, Model.of(""));

        Form<User> userUpdateForm =
                new Form<User>(USER_UPDATE_FORM, new CompoundPropertyModel<User>(model)) {
                    @Override
                    protected void onSubmit() {
                        userRepository.save(getModelObject());
                    }
                };

        add(userUpdateForm
                .add(new TextField<String>("name"))
                .add(password)
                .add(passwordRepeat)
                .add(new AjaxFallbackButton("submit", userUpdateForm) {})
                .add(new EqualInputValidator(password, passwordRepeat))
        );
    }
}
```

```
<!DOCTYPE html>
<html xmlns:wicket="http://wicket.apache.org">
    <head>
        <meta charset="utf-8">
        <title>Wicket Example App</title>
    </head>
    <body>
        <wicket:panel>
            <form wicket:id="userUpdateForm">
                <fieldset>
                    <legend><wicket:message key="userUpdateTitle"/></legend>
                    <input type="text" wicket:id="name"/>
                    <input type="password" wicket:id="password"/>
                    <input type="password" wicket:id="passwordRepeat"/>
                    <p><input type="submit" value="submit" wicket:id="submit"/></p>
                </fieldset>
            </form>
        </wicket:panel>
    </body>
</html>
```

wicket:body

wicket:message

wicket:child

wicket:fragment

wicket:remove

```
<html xmlns:wicket="http://wicket.apache.org">
```

wicket:extend

wicket:container

wicket:head

wicket:id

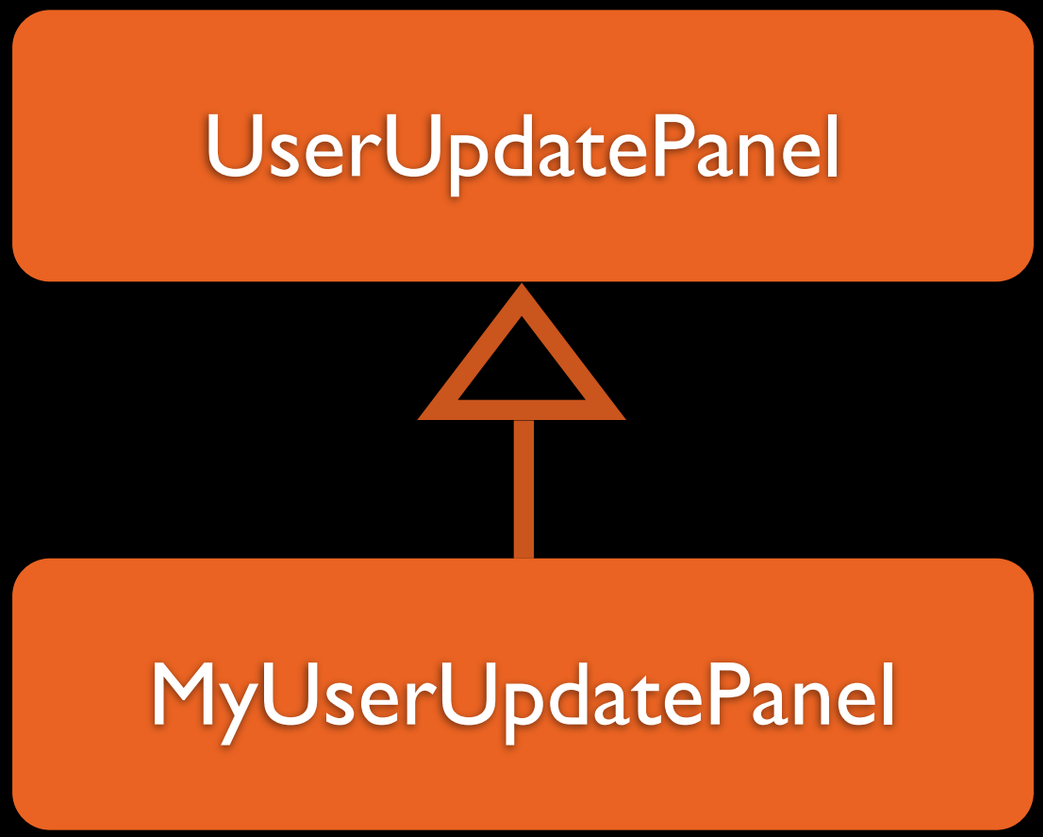
wicket:panel

wicket:link

wicket:border

Keine Logik im Template

Inheritance



```java
public class MyUserUpdatePanel extends UserUpdatePanel{

    public UserUpdatePanel(String id, IModel<User> model) {
        super(id, model);
    }
}
```

MyUserUpdatePanel.html

```html
<!DOCTYPE html>
<html xmlns:wicket="http://wicket.apache.org">
    <head>
        <meta charset="utf-8">
        <title>Wicket Example App</title>
    </head>
    <body>
        <wicket:panel>
            <form wicket:id="userUpdateForm">
                <fieldset>
                    <legend><wicket:message key="userUpdateTitle"/></legend>
                    <input type="text" wicket:id="name"/>
                    <input type="password" wicket:id="password"/>
                    <input type="password" wicket:id="passwordRepeat"/>
                    <p><input type="submit" value="submit" wicket:id="submit"/></p>
                </fieldset>
            </form>
        </wicket:panel>
    </body>
</html>
```

MyUserUpdatePanel.properties

```
userUpdateTitle=Lustiges Felderraten
```

Benutzer anpassen

Benutzername

Passwort

Passwort wiederholen

submit

Lustiges Felderraten

submit

Got my stuff?

Markup

Properties

Datenbeschaffung

```
public UserUpdatePanel(String id, IModel<User> model)
```

```
package de.bootcamp.wicket.web.page;

public interface IModel<T> extends IDetachable
{
    T getObject();
    void setObject(final T object);
}
```

```java
public class UserUpdatePanel extends Panel{


...


    public UserUpdatePanel(String id, IModel<User> model) {
        super(id, model);

        FormComponent<String> password = new PasswordTextField(PASSWORD, new PropertyModel<String>(model, "password"));
        FormComponent<String> passwordRepeat = new PasswordTextField(PASSWORD_REPEAT, Model.of(""));

        Form<User> userUpdateForm =
                new Form<User>(USER_UPDATE_FORM, new CompoundPropertyModel<User>(model)) {
                    @Override
                    protected void onSubmit() {
                        userRepository.save(getModelObject());
                    }
                };

...

    }
}
```

IModel

- Entkopplung Beschaffung / Verwendung
- Lazy
- Chainable
- Leicht testbar

Behavior

Component .add(Behavior)

- AttributeAppender/AttributeModifier
- AjaxSelfUpdatingBehavior

Behavior

Component .add(Behavior)

- AttributeAppender/AttributeModifier
- AjaxSelfUpdatingBehavior
- AjaxEventBehavior
- ...

Alles zusammen

Page

```
public class HomePage extends WebPage {
    private static final long serialVersionUID = 1L;

    @SpringBean
    private BusinessService businessService;

    public HomePage(final PageParameters parameters) {
        super(parameters);

        IModel<User> userModel = new LoadableDetachableModel<User>() {
            @Override
            protected User load() {
                return businessService.findUserByName("user1");
            }
        };
        add(new Label("name", new PropertyModel<String>(userModel,"name")));

        add(new MyUserUpdatePanel("userUpdatePanel", userModel));

        add(new AjaxLink("adminLink") {
            @Override
            public void onClick(AjaxRequestTarget target) {
                setResponsePage(AdminPage.class);
            }
        });
    }
}
```

Praxis!!!

AJAX

- Seit Wicket 6: JQuery!
- Transparente Verwendung
- Maßgeblicher Bestandteil des Wicket-Kerns
- Einfach über Behaviors erweiterbar
- Weiterführung der Komponentisierung

Tasker

Logout

Bummer!
Something went wrong.

64x64

Hello Tasksolver

Open tasks: **200**

AUFGABENLISTEN

Home

Library

Add a new Task

Current List: Main

✔	Do something ... fast!!!	21.03.2013
✔		03.12.1976
✔	~~This task is done.~~	22.03.2013
✔	Check out Angular.JS	
✔	Hate Struts	
✔	Hate Struts some more	
✔	Show love to Spring-Data	
✔	Some test task.	
✔	And another won.	

Tasker Logout

Bummer!
Something went wrong.

64x64

Hello Tasksolver

Open tasks: **200**

AUFGABENLISTEN

Home

Library

Add a new Task

Current List: Main

✔	Do something ... fast!!!	21.03.2013
✔		03.12.1976
✔	~~This task is done.~~	22.03.2013
✔	Check out Angular.JS	
✔	Hate Struts	
✔	Hate Struts some more	
✔	Show love to Spring-Data	
✔	Some test task.	
✔	And another won.	

AJAX > NoScript

AjaxFallbackButton
AjaxFallbackDefaultDataTable
AjaxFallbackHeadersToolbar
AjaxFallbackLink
...

Kopplung

~~Callbacks~~

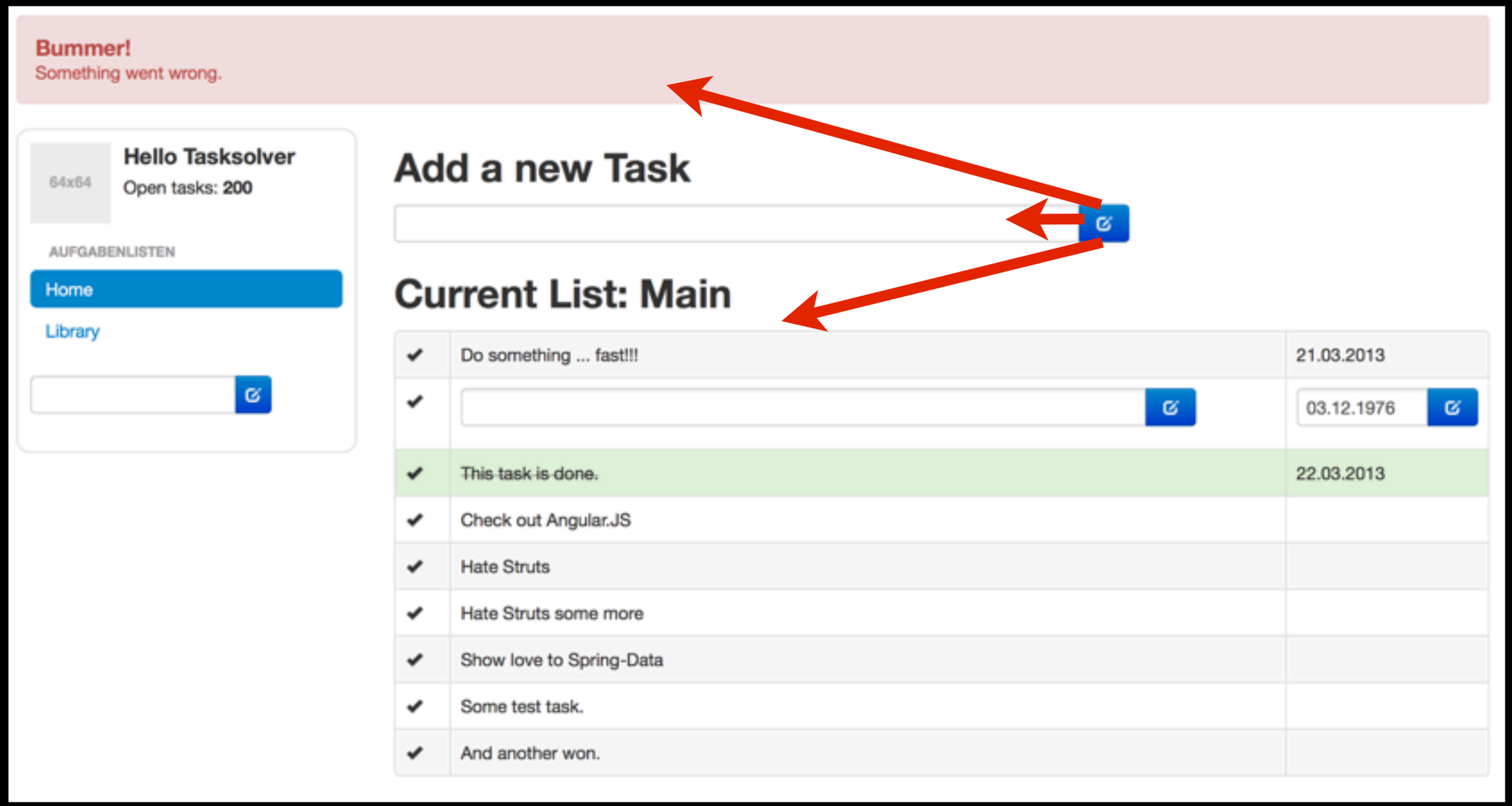
Events

```java
public interface IEventSink
{
    void onEvent(IEvent<?> event);
}
```

```java
public interface IEventSource
{
    <T> void send(IEventSink sink,
        Broadcast broadcast, T payload);
}
```

- Schwache Kopplung
- hohe Flexibilität
- weniger Boilerplate

Ajax Default Event



```
...
    @Override
    public void onEvent(IEvent<?> event) {
        if(event.getPayload() instanceof AjaxRequestTarget) {
            ((AjaxRequestTarget)event.getPayload()).add(this);
        }
    }
...
```

WicketTester

```
private WicketTester wicketTester;

@Before
public void setUp() {
    wicketTester = new WicketTester();
}

@After
public void tearDown() {
    wicketTester.destroy();
}

@Test
public void testCreateTask() throws Exception{
    IModel<TaskList> taskListMock = mock(IModel.class);
    User testUser = new User();

    CreateTaskPanel createTaskPanel = new CreateTaskPanel("testPanel", testUser, taskListMock);

    wicketTester.startComponentInPage(createTaskPanel);
    wicketTester.assertComponent("testPanel", CreateTaskPanel.class);

    FormTester formTester = wicketTester.newFormTester("testPanel:taskForm");
    formTester.setValue("title", "taskTitle");
    formTester.submit();

    ArgumentCaptor<Task> taskCaptor = ArgumentCaptor.forClass(Task.class);
    verify(taskServiceMock).create(taskCaptor.capture());

    assertEquals("taskTitle", taskCaptor.getValue().getTitle());
    assertEquals(testUser, taskCaptor.getValue().getUser());
}
```

Ressourcen

- JavaScript
- CSS
- Bilder
- PDF
- …

```java
public class DataTablesResource extends JavaScriptResourceReference {

    private static final Set<HeaderItem> deps;
    static {
        HashSet<HeaderItem> items = new HashSet<HeaderItem>();
        items.add(JavaScriptHeaderItem.forReference(JQueryResourceReference.get()));
        deps = Collections.unmodifiableSet(items);
    }

    private static final DataTablesResource instance = new DataTablesResource();
    private DataTablesResource() {
        super(DataTablesResource.class, "jquery.dataTables.js");
    }

    public static DataTablesResource get() {
        return instance;
    }

    @Override
    public Iterable<? extends HeaderItem> getDependencies() {
        return deps;
    }
}
```

```java
...

@Override
public void renderHead(IHeaderResponse response) {
    super.renderHead(response);
    response.render(JavaScriptHeaderItem.forReference(DataTablesResource.get()));
}

...
```

Und weiter?

- Optimierung der verwendeten Ressourcen
- Kontrolle der URL
- Dependency Management
- JavaScript ResourceBundles
- Übersteuern von Versionen

Wicket-Guide

http://code.google.com/p/wicket-guide/

GET IT!

DANKE