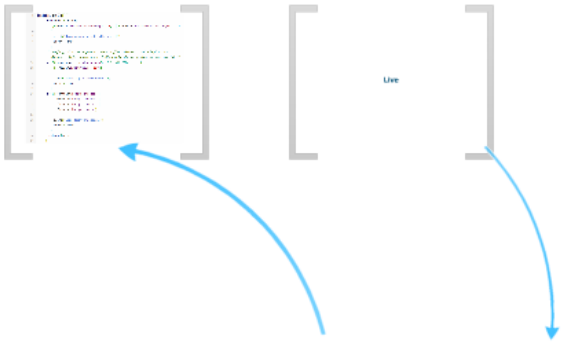


Google Web Toolkit

Einführung

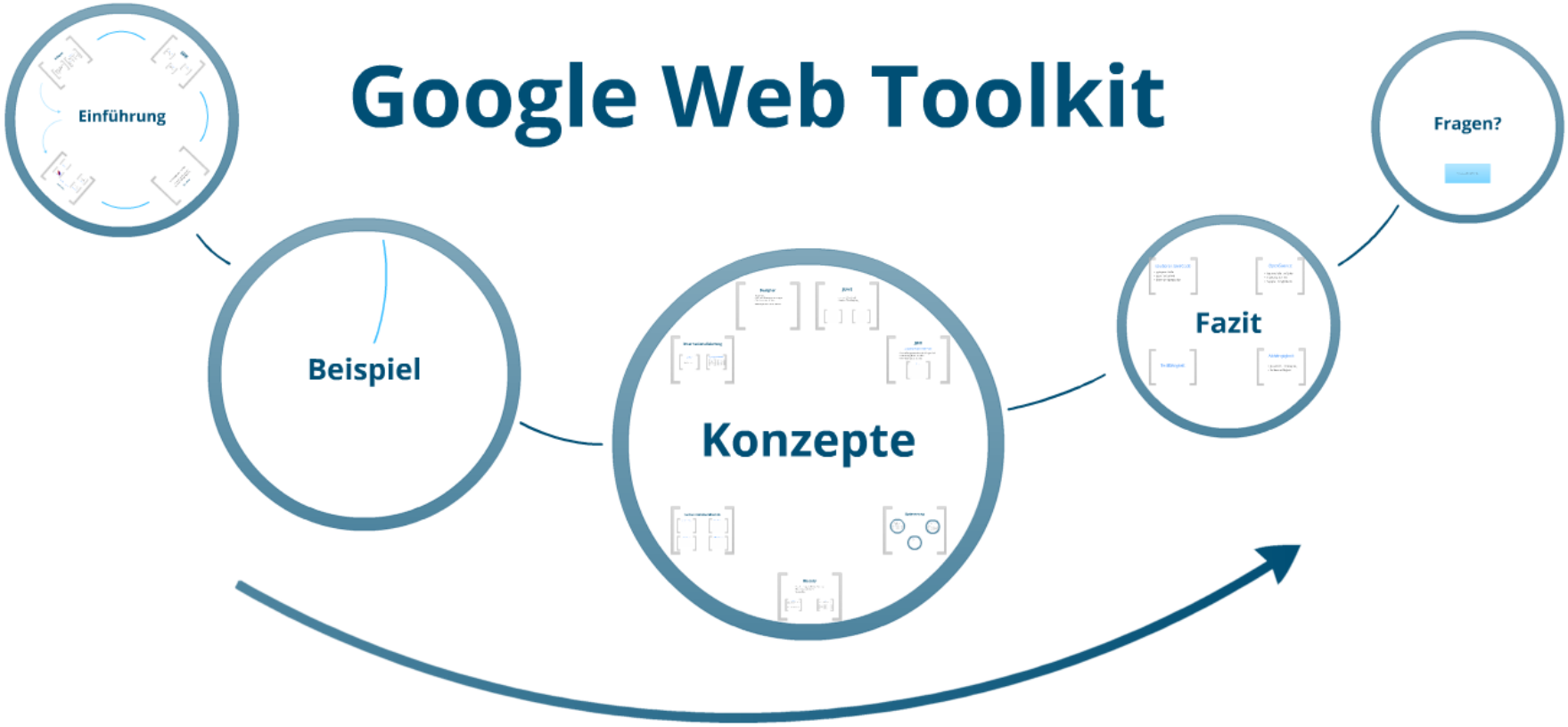
Beispiel

Konzepte

Fazit

Fragen?

Google Web Toolkit




Eclipse
SDK
Einführung
Motivation
JavaScript

Motivation

Entwicklung in Java

Warum Java?

- Entwickler
- Java
  - IDE
  - Modularisierung
  - Tooling

Kommunikation 1

Warum GWT?

- Seit 2006 am Markt
- OpenSource-Projekt
- viele erfolgreiche Projekte:
  - studiVZ / meinVZ / schülerVZ
  - Google Groups
  - Google AdWords
  - u.v.m.

Kommunikation 2

Warum Java?

- Entwickler
- Java
  - IDE
  - Modularisierung
  - Tooling

Warum GWT?

- Seit 2006 am Markt
- OpenSource-Projekt
- viele erfolgreiche Projekte:
  - studiVZ / meinVZ / schülerVZ
  - Google Groups
  - Google AdWords
  - u.v.m.

Kommunikation 1

```
1.  var xmlHttp;
2.  if (window.XMLHttpRequest)
3.    xmlHttp = new XmlHttpRequest();
4.
5.  xmlHttp.onreadestatechange = function() {
6.    if (xmlHttp.readyState == 4)
7.        showProducts(xmlHttp.responseText);
8.  };
9.
10. xmlHttp.open("POST", "product.jsp", true);
11. xmlHttp.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
12. xmlHttp.send("category=1");
```

Kommunikation 2

```
1. $.ajax({
2.   type: "POST",
3.   url: "product.jsp",
4.   data: {
5.     category: 1
6.   }
7. })
8. .done(function(msg) {
9.   showProducts(msg);
10. });
```

Motivation

Entwicklung in Java

Warum Java?

- Entwickler
- Java
  - IDE
  - Modularisierung
  - Tooling

Kommunikation 1

Warum GWT?

- Seit 2006 am Markt
- OpenSource-Projekt
- viele erfolgreiche Projekte:
  - studiVZ / meinVZ / schülerVZ
  - Google Groups
  - Google AdWords
  - u.v.m.

Kommunikation 2

JavaScript

- keine Einführung notwendig
- Entwicklung erfolgt in Java

- JavaScript über JSNI möglich

SDK

webAppCreator

- Kommandozeilentool
- erzeugt neue Projekte
- vorgefertigte Dateien für eclipse

ant build

- Code Erzeugung mit Optimierung

ant devmode

- Development Mode
- zusätzliches Browser-Plugin
- Debugging-Funktionalität
- On-demand JavaScript-Erzeugung

webAppCreator

- Kommandozeilentool
- erzeugt neue Projekte
- vorgefertigte Dateien für eclipse

ant devmode

- Development Mode
- zusätzliches Browser-Plugin
- Debugging-Funktionalität
- On-demand JavaScript-Erzeugung

ant build

- Code Erzeugung mit Optimierung

SDK

webAppCreator

- Kommandozeilentool
- erzeugt neue Projekte
- vorgefertigte Dateien für eclipse

ant build

- Code Erzeugung mit Optimierung

ant devmode

- Development Mode
- zusätzliches Browser-Plugin
- Debugging-Funktionalität
- On-demand JavaScript-Erzeugung

Eclipse

- Unterstützung mehrerer Versionen
- Erstellung neuer Projekte
- Starten der Applikation
- Code-Erzeugung
- Designer

App Engine

- kostenloser Speicherplatz für Web-Applikationen
- Entwicklung in 4 Sprachen möglich

- Unterstützung mehrerer Versionen
- Erstellung neuer Projekte
- Starten der Applikation
- Code-Erzeugung
- Designer

App Engine

- kostenloser Speicherplatz für Web-Applikationen
- Entwicklung in 4 Sprachen möglich

kostenlos

- 5 Millionen Seitenaufrufe / Monat

- DataStorage: 1GB
- Code & Static Data: 1GB
- Traffic: 1GB, 54MB / Minute

Features

- Versionsverwaltung
- Benutzer-Funktionalität
- Skalierbarkeit der Server
- ...

kostenlos

- 5 Millionen Seitenaufrufe / Monat

- DataStorage: 1GB
- Code & Static Data: 1GB
- Traffic: 1GB, 54MB / Minute

Features

- Versionsverwaltung
- Benutzer-Funktionalität
- Skalierbarkeit der Server
- ...

Eclipse

- Unterstützung mehrerer Versionen
- Erstellung neuer Projekte
- Starten der Applikation
- Code-Erzeugung
- Designer

App Engine

- kostenloser Speicherplatz für Web-Applikationen
- Entwicklung in 4 Sprachen möglich


Eclipse
SDK
Einführung
Motivation
JavaScript

Beispiel

```
1.  function pruefe() {
2.        var Error = false;
3.        if ((formular.vorname.value.length < 1) || (formular.nachname.value.length < 1))
4.          {
5.            alert("Namensangabe nicht vollständig!");
6.            Error = true;
7.          }
8.        /* Prüfung, ob Altersangabe mit einer Ziffer zwischen 1 und 9 beginnt und
9.          mit 0 - 2 Ziffern endet sowie ob die Angabe kleiner gleich dem Wert 120 ist */
10.       if ((formular.alter.value.search(/^[1-9]\d{0,2}$/) == -1) ||
11.           (formular.alter.value > 120))
12.         {
13.           alert("Altersangabe fehlerhaft!");
14.           Error = true;
15.         }
16.       if (!(formular.hobby1.checked ||
17.             formular.hobby2.checked ||
18.             formular.hobby3.checked ||
19.             formular.hobby4.checked))
20.         {
21.           alert("Kein Hobby angegeben!");
22.           Error = true;
23.         }
24.        return !Error;
25.     }
```

Live

Beispiel

Konzepte


Designer


JUnit


Internationalisierung


JSNI
JavaScript Native Interface

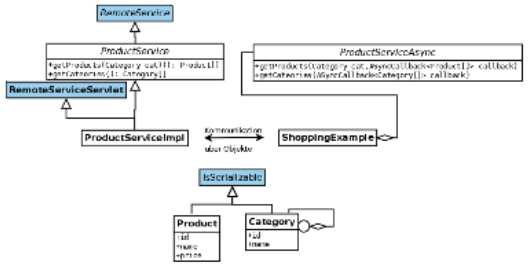

Server-Kommunikation


Optimierung


History

Server-Kommunikation

RPC - Remote Procedure Calls



ProductServiceImpl

```
@Override
public Product[] getProducts(Category aCategory) {
  // Produkt SQL zusammenbauen
  SqlMultiRelationalModel fModel = new SqlMultiRelationalModel();
  fModel.setTable("product");
  fModel.addSelectColumn("idx_product");
  fModel.addSelectColumn("name");
  fModel.addSelectColumn("price");
  fModel.addWhereStatement("product", "idx_category", aCategory.getId());

  Product[] fRes = new Product[0];

  List<List<String>> fResList = new ArrayList<List<String>>();
  if (fModel.select(fResList)) {
    fRes = new Product[fResList.size()];
    int i = 0;

    for (List<String> fIter : fResList)  {
      fRes[i] = new Product(fIter.get(0), aCategory.getId(), fIter.get(1), Float.parseFloat(fIter.get(2)));
      ++i;
    }
  }

  return fRes;
}
```

ProductService

```
package de.fhwedel.gwt.shoppingexample.shared;

@RemoteServiceRelativePath("product")
public interface ProductService extends RemoteService {
  Category[] getCategories();
  Product[] getProducts(Category aCategory);
}
```

```
package de.fhwedel.gwt.shoppingexample.shared;

public interface ProductServiceAsync {
        void getProducts(Category aCategory, AsyncCallback<Product[]> callback);
        void getCategories(AsyncCallback<Category[]> callback);
}
```

ShoppingExample

```
private final ProductServiceAsync _productService = GWT.create(ProductService.class);


public void receiveProductList(final Category aCategory) {
        _productService.getProducts(aCategory, new AsyncCallback<Product[]>() {
                public void onFailure(Throwable caught) {
                        Window.alert(caught.getMessage());
                }

                public void onSuccess(Product[] result) {
                        _productTable.clear();
                        _productTable.removeAllRows();
                        _productTable.setStyleName("productTable");
                        _productTable.setText(0, 0, "Name");
                        _productTable.setText(0, 1, "Preis");

                        for (int i = 0; i < result.length; ++i) {
                                _productTable.setText(i + 1, 0, result[i].getName());
                                _productTable.setText(i + 1, 1, String.valueOf(result[i].getPrice()));
                        }
                }
        });
}
```

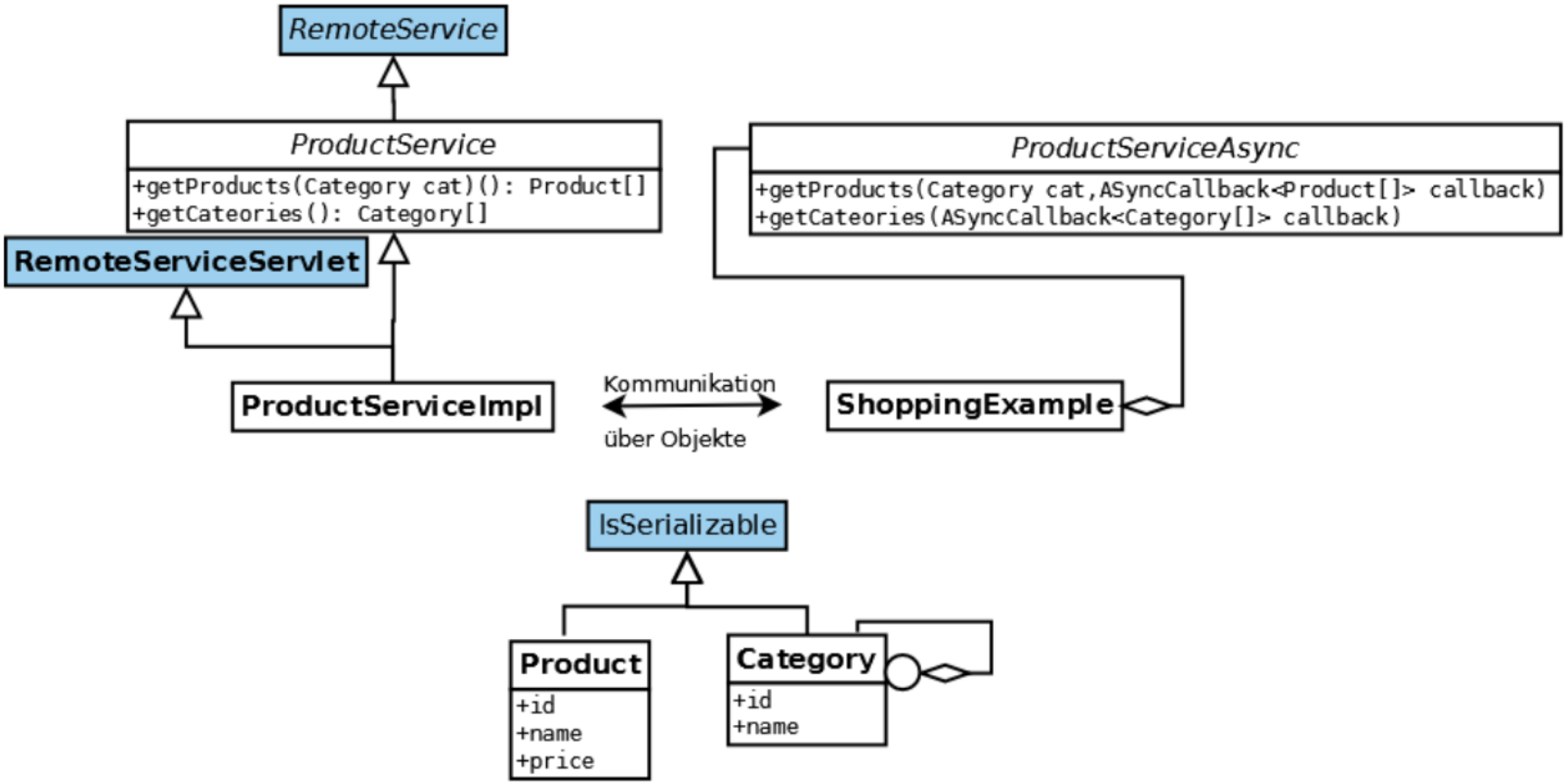
Kommunikation 1

```
1.  var xmlHttp;
2.  if (window.XMLHttpRequest)
3.    xmlHttp = new XmlHttpRequest();
4.
5.  xmlHttp.onreadestatechange = function() {
6.    if (xmlHttp.readyState == 4)
7.        showProducts(xmlHttp.responseText);
8.  };
9.
10. xmlHttp.open("POST", "product.jsp", true);
11. xmlHttp.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
12. xmlHttp.send("category=1");
```

Kommunikation 2

```
1. $.ajax({
2.   type: "POST",
3.   url: "product.jsp",
4.   data: {
5.     category: 1
6.   }
7. })
8. .done(function(msg) {
9.   showProducts(msg);
10. });
```

RPC - Remote Procedure Calls



ProductService

```java
package de.fhwedel.gwt.shoppingexample.shared;

@RemoteServiceRelativePath("product")
public interface ProductService extends RemoteService {
  Category[] getCategories();
  Product[] getProducts(Category aCategory);
}
```

```java
package de.fhwedel.gwt.shoppingexample.shared;

public interface ProductServiceAsync {
        void getProducts(Category aCategory, AsyncCallback<Product[]> callback);
        void getCategories(AsyncCallback<Category[]> callback);
}
```

ShoppingExample

```
1.  private final ProductServiceAsync _productService = GWT.create(ProductService.class);
2.
3.
4.  public void receiveProductList(final Category aCategory) {
5.          _productService.getProducts(aCategory, new AsyncCallback<Product[]>() {
6.                  public void onFailure(Throwable caught) {
7.                          Window.alert(caught.getMessage());
8.                  }
9.
10.                 public void onSuccess(Product[] result) {
11.                         _productTable.clear();
12.                         _productTable.removeAllRows();
13.                         _productTable.setStyleName("productTable");
14.                         _productTable.setText(0, 0, "Name");
15.                         _productTable.setText(0, 1, "Preis");
16.
17.                         for (int i = 0; i < result.length; ++i) {
18.                                 _productTable.setText(i + 1, 0, result[i].getName());
19.                                 _productTable.setText(i + 1, 1, String.valueOf(result[i].getPrice()));
20.                         }
21.                 }
22.         });
23. }
```

Kommunikation 2

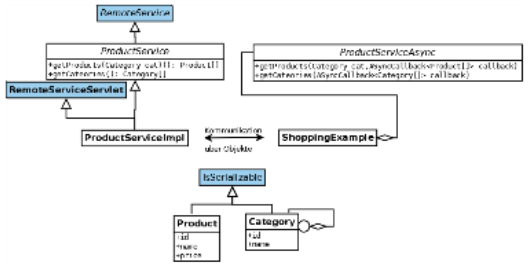
```
1. $.ajax({
2.   type: "POST",
3.   url: "product.jsp",
4.   data: {
5.     category: 1
6.   }
7. })
8. .done(function(msg) {
9.   showProducts(msg);
10. });
```

ProductServiceImpl

```java
@Override
public Product[] getProducts(Category aCategory) {
  // Produkt SQL zusammenbauen
  SqlMultiRelationalModel fModel = new SqlMultiRelationalModel();
  fModel.setTable("product");
  fModel.addSelectColumn("idx_product");
  fModel.addSelectColumn("name");
  fModel.addSelectColumn("price");
  fModel.addWhereStatement("product", "idx_category", aCategory.getId());

  Product[] fRes = new Product[0];

  List<List<String>> fResList = new ArrayList<List<String>>();
  if (fModel.select(fResList)) {
    fRes = new Product[fResList.size()];
    int i = 0;

    for (List<String> fIter : fResList)  {
      fRes[i] = new Product(fIter.get(0), aCategory.getId(), fIter.get(1), Float.parseFloat(fIter.get(2)));
      ++i;
    }
  }

  return fRes;
}
```

Server-Kommunikation

RPC - Remote Procedure Calls



ProductServiceImpl

```
@Override
public Product[] getProducts(Category aCategory) {
  // Produkt SQL zusammenbauen
  SqlMultiRelationalModel fModel = new SqlMultiRelationalModel();
  fModel.setTable("product");
  fModel.addSelectColumn("idx_product");
  fModel.addSelectColumn("name");
  fModel.addSelectColumn("price");
  fModel.addWhereStatement("product", "idx_category", aCategory.getId());

  Product[] fRes = new Product[0];

  List<List<String>> fResList = new ArrayList<List<String>>();
  if (fModel.select(fResList)) {
    fRes = new Product[fResList.size()];
    int i = 0;

    for (List<String> fIter : fResList)  {
      fRes[i] = new Product(fIter.get(0), aCategory.getId(), fIter.get(1), Float.parseFloat(fIter.get(2)));
      ++i;
    }
  }

  return fRes;
}
```

ProductService

```
package de.fhwedel.gwt.shoppingexample.shared;

@RemoteServiceRelativePath("product")
public interface ProductService extends RemoteService {
  Category[] getCategories();
  Product[] getProducts(Category aCategory);
}
```

```
package de.fhwedel.gwt.shoppingexample.shared;

public interface ProductServiceAsync {
        void getProducts(Category aCategory, AsyncCallback<Product[]> callback);
        void getCategories(AsyncCallback<Category[]> callback);
}
```

ShoppingExample

```
private final ProductServiceAsync _productService = GWT.create(ProductService.class);


public void receiveProductList(final Category aCategory) {
        _productService.getProducts(aCategory, new AsyncCallback<Product[]>() {
                public void onFailure(Throwable caught) {
                        Window.alert(caught.getMessage());
                }

                public void onSuccess(Product[] result) {
                        _productTable.clear();
                        _productTable.removeAllRows();
                        _productTable.setStyleName("productTable");
                        _productTable.setText(0, 0, "Name");
                        _productTable.setText(0, 1, "Preis");

                        for (int i = 0; i < result.length; ++i) {
                                _productTable.setText(i + 1, 0, result[i].getName());
                                _productTable.setText(i + 1, 1, String.valueOf(result[i].getPrice()));
                        }
                }
        });
}
```

History

- Unterstützung der History-Funktion
- Benutzerfreundlichkeit
- Lesezeichen

History-Token

- jede Seite bekommt über die URL einen Token
- Token in Form eines Ankers

http://www.example.com/home.html#page1

Beispiel

- Klasse muss ValueChangeHandler implementieren

History-Token

- jede Seite bekommt über die URL einen Token
- Token in Form eines Ankers

http://www.example.com/home.html#page1

Beispiel

- Klasse muss ValueChangeHandler implementieren

```
public void onModuleLoad() {
  History.addValueChangeHandler(this);

  buildFormLayout();
  buildSubmitLayout();

  if (History.getToken().isEmpty())
    History.newItem("home");
  else
    changePage(History.getToken());
}
```

```
@Override
public void onValueChange(ValueChangeEvent<String> event) {
  changePage(History.getToken());
}

public void changePage(String token) {
  _formPanel.setVisible(false);
  _submitLabel.setVisible(false);

  if (History.getToken().equals("submit")) {
    _submitLabel.setVisible(true);
    _submitLabel.setText("Die Daten wurden erfolgreich übermittelt.");
  }
  else
    _formPanel.setVisible(true);
}

_formPanel.addSubmitCompleteHandler(new SubmitCompleteHandler() {
  @Override
  public void onSubmitComplete(SubmitCompleteEvent event) {
    History.newItem("submit");
  }
});
```

```
1.  public void onModuleLoad() {
2.    History.addValueChangeHandler(this);
3.
4.    buildFormLayout();
5.    buildSubmitLayout();
6.
7.    if (History.getToken().isEmpty())
8.      History.newItem("home");
9.    else
10.     changePage(History.getToken());
11. }
```

```
1.  @Override
2.  public void onValueChange(ValueChangeEvent<String> event) {
3.    changePage(History.getToken());
4.  }
5.
6.  public void changePage(String token) {
7.    _formPanel.setVisible(false);
8.    _submitLabel.setVisible(false);
9.
10.   if (History.getToken().equals("submit")) {
11.     _submitLabel.setVisible(true);
12.     _submitLabel.setText("Die Daten wurden erfolgreich übermittelt.");
13.   }
14.   else
15.     _formPanel.setVisible(true);
16. }
17.
18.   _formPanel.addSubmitCompleteHandler(new SubmitCompleteHandler() {
19.     @Override
20.     public void onSubmitComplete(SubmitCompleteEvent event) {
21.       History.newItem("submit");
22.     }
23.   });
```

History

- Unterstützung der History-Funktion
- Benutzerfreundlichkeit
- Lesezeichen

History-Token

- jede Seite bekommt über die URL einen Token
- Token in Form eines Ankers

http://www.example.com/home.html#page1

Beispiel

- Klasse muss ValueChangeHandler implementieren

Optimierung

Compiler

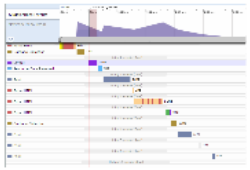
- "dead"-Code
- Speicherplatz
- Browser-Optimierung
- optional Closure-Compiler

Code Splitting

- Code wird geladen wenn er benötigt wird
- schnellere Ladezeiten
- große Projekte

Speed Tracer

- Zeitmessung
- Optimierung an richtigen Stellen



Compiler

- "dead"-Code
- Speicherplatz
- Browser-Optimierung
- optional Closure-Compiler

Speicherplatz

Compile-Report

Live

Speicherplatz

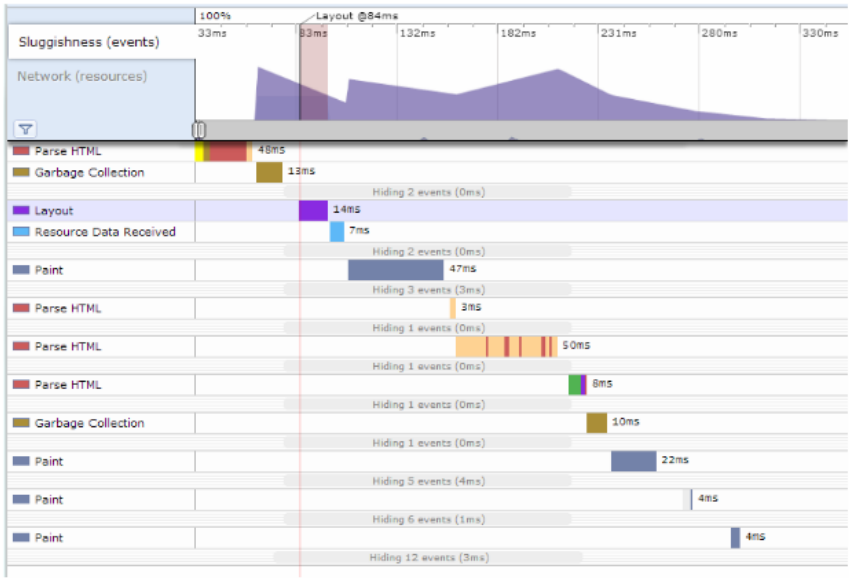
```
function datenpruefen(){var O='',wb='" for "gwt:onLoadErrorFn"',ub='" for "gwt:onPropertyError
function C(){if(q&&r){var b=n.getElementById(P);var c=b.contentWindow;if(B()){c.__gwt_getPrope
function D(){function e(a){var b=a.lastIndexOf(Y);if(b==-1){b=a.length}var c=a.indexOf(Z);if(c
function f(a){if(a.match(/^\w+:\/\//)){}else{var b=n.createElement(_);b.src=a+ab;a=e(b.src)}re
function g(){var a=F(bb);if(a!=null){return a}return O}
function h(){var a=n.getElementsByTagName(cb);for(var b=0;b<a.length;++b){if(a[b].src.indexOf(
function i(){var a;if(typeof isBodyLoaded==eb||!isBodyLoaded()){var b=fb;var c;n.write(gb+b+hb
function j(){var a=n.getElementsByTagName(jb);if(a.length>0){return a[a.length-1].href}return
function k(){var a=n.location;return a.href==a.protocol+kb+a.host+a.pathname+a.search+a.hash}
var l=g();if(l==O){l=h()}if(l==O){l=i()}if(l==O){l=j()}if(l==O&&k()){l=e(n.location.href)}l=f(
function E(){var b=document.getElementsByTagName(lb);for(var c=0,d=b.length;c<d;++c){var e=b[c
function F(a){var b=u[a];return b==null?null:b}
function G(a){var b=w[a](),c=v[a];if(b in c){return b}var d=[];for(var e in c){d[c[e]]=e}if(A)
var H;function I(){if(!H){H=true;var a=n.createElement(xb);a.src=yb;a.id=P;a.style.cssText=zb;
w[Bb]=function(){var b=navigator.userAgent.toLowerCase();var c=function(a){return parseInt(a[1
if(n.addEventListener){n.addEventListener(Yb,function(){I();N()},false)}var M=setInterval(func
datenpruefen();
```

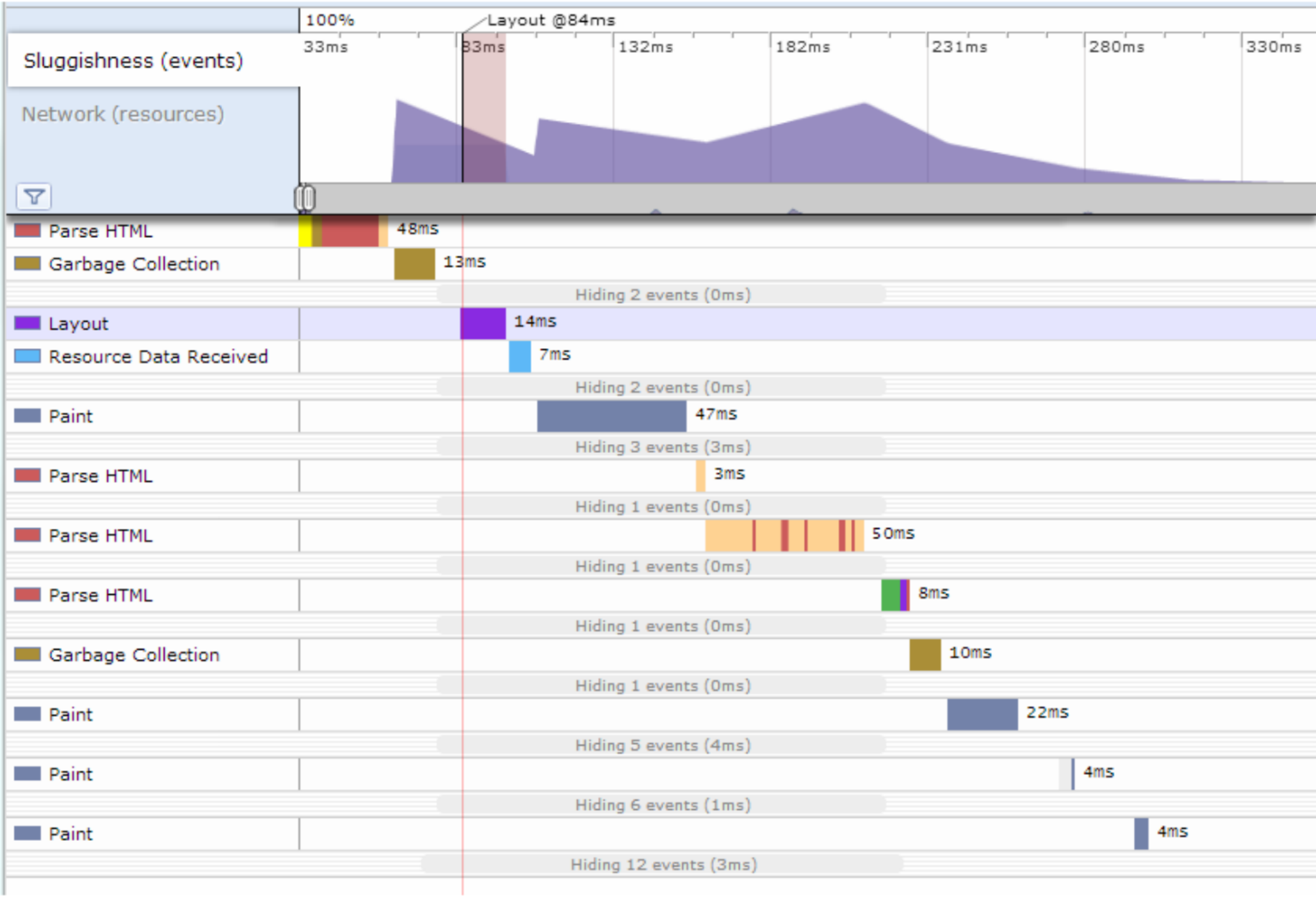
Compile-Report

Live

Speed Tracer

- Zeitmessung
- Optimierung an richtigen Stellen




100%
Layout @84ms
33ms
83ms
132ms
182ms
231ms
280ms
330ms
Sluggishness (events)
Network (resources)
Parse HTML
48ms
Garbage Collection
13ms
Hiding 2 events (0ms)
Layout
14ms
Resource Data Received
7ms
Hiding 2 events (0ms)
Paint
47ms
Hiding 3 events (3ms)
Parse HTML
3ms
Hiding 1 events (0ms)
Parse HTML
50ms
Hiding 1 events (0ms)
Parse HTML
8ms
Hiding 1 events (0ms)
Garbage Collection
10ms
Hiding 1 events (0ms)
Paint
22ms
Hiding 5 events (4ms)
Paint
4ms
Hiding 6 events (1ms)
Paint
4ms
Hiding 12 events (3ms)

Code Splitting

- Code wird geladen wenn er benötigt wird
- schnellere Ladezeiten
- große Projekte

```
if (History.getToken().equals("submit")) {
  _submitLabel.setVisible(true);
  _submitLabel.setText("Die Daten wurden erfolgreich übermittelt.");
}
```

```
if (History.getToken().equals("submit")) {
  GWT.runAsync(new RunAsyncCallback() {
    public void onFailure(Throwable caught) {
      Window.alert("Code download failed");
    }

    public void onSuccess() {
      _submitLabel.setVisible(true);
      _submitLabel.setText("Die folgenden Daten wurden erfolgreich übermittelt.");
    }
  });
}
```

```
1. if (History.getToken().equals("submit")) {
2.   _submitLabel.setVisible(true);
3.   _submitLabel.setText("Die Daten wurden erfolgreich übermittelt.");
4. }
```

```
1.  if (History.getToken().equals("submit")) {
2.    GWT.runAsync(new RunAsyncCallback() {
3.      public void onFailure(Throwable caught) {
4.        Window.alert("Code download failed");
5.      }
6.
7.      public void onSuccess() {
8.        _submitLabel.setVisible(true);
9.        _submitLabel.setText("Die folgenden Daten wurden erfolgreich übermittelt.");
10.     }
11.   });
12. }
```

Optimierung

Compiler

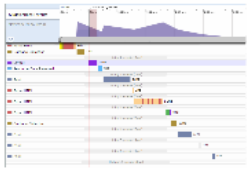
- "dead"-Code
- Speicherplatz
- Browser-Optimierung
- optional Closure-Compiler

Code Splitting

- Code wird geladen wenn er benötigt wird
- schnellere Ladezeiten
- große Projekte

Speed Tracer

- Zeitmessung
- Optimierung an richtigen Stellen



JSNI

JavaScript Native Interface

- Einbindung von nativen JavaScript-Code
- Umsetzung älterer Projekte
- Einbindung von Librarys

Beispiel

```
private native boolean checkAndMarkErrornousInput() /*-{
  var Error = false;

  var vorname = this.@de.fhwedel.gwt.datenpruefen.client.DatenPruefen::getForname()();
  var nachname = this.@de.fhwedel.gwt.datenpruefen.client.DatenPruefen::getSurname()();
  var alter =  this.@de.fhwedel.gwt.datenpruefen.client.DatenPruefen::getAge()();
  var hobbySelected = this.@de.fhwedel.gwt.datenpruefen.client.DatenPruefen::isHobbySelected()();

  if ((vorname.length < 1) || (nachname.length < 1))
      {
        alert("Namensangabe nicht vollständig!");
        Error = true;
      }
    // Prüfung, ob Altersangabe mit einer Ziffer zwischen 1 und 9 beginnt und
    //  mit 0 - 2 Ziffern endet sowie ob die Angabe kleiner gleich dem Wert 120 ist
    if ((alter.search(/^[1-9]\d{0,2}$/) == -1) ||
        (alter > 120))
      {
        alert("Altersangabe fehlerhaft!");
        Error = true;
      }
    if (!(hobbySelected))
      {
        alert("Kein Hobby angegeben!");
        Error = true;
      }

     return !Error;

}-*/;
```

Beispiel

```
private native boolean checkAndMarkErrornousInput() /*-{
  var Error = false;

  var vorname = this.@de.fhwedel.gwt.datenpruefen.client.DatenPruefen::getForname()();
  var nachname = this.@de.fhwedel.gwt.datenpruefen.client.DatenPruefen::getSurname()();
  var alter =  this.@de.fhwedel.gwt.datenpruefen.client.DatenPruefen::getAge()();
  var hobbySelected = this.@de.fhwedel.gwt.datenpruefen.client.DatenPruefen::isHobbySelected()();

  if ((vorname.length < 1) || (nachname.length < 1))
      {
        alert("Namensangabe nicht vollständig!");
        Error = true;
      }
    // Prüfung, ob Altersangabe mit einer Ziffer zwischen 1 und 9 beginnt und
    //  mit 0 - 2 Ziffern endet sowie ob die Angabe kleiner gleich dem Wert 120 ist
    if ((alter.search(/^[1-9]\d{0,2}$/) == -1) ||
        (alter > 120))
      {
        alert("Altersangabe fehlerhaft!");
        Error = true;
      }
    if (!(hobbySelected))
      {
        alert("Kein Hobby angegeben!");
        Error = true;
      }

     return !Error;

}-*/;
```

JSNI

JavaScript Native Interface

- Einbindung von nativen JavaScript-Code
- Umsetzung älterer Projekte
- Einbindung von Librarys

Beispiel

```
private native boolean checkAndMarkErrornousInput() /*-{
  var Error = false;

  var vorname = this.@de.fhwedel.gwt.datenpruefen.client.DatenPruefen::getForname()();
  var nachname = this.@de.fhwedel.gwt.datenpruefen.client.DatenPruefen::getSurname()();
  var alter =  this.@de.fhwedel.gwt.datenpruefen.client.DatenPruefen::getAge()();
  var hobbySelected = this.@de.fhwedel.gwt.datenpruefen.client.DatenPruefen::isHobbySelected()();

  if ((vorname.length < 1) || (nachname.length < 1))
      {
        alert("Namensangabe nicht vollständig!");
        Error = true;
      }
    // Prüfung, ob Altersangabe mit einer Ziffer zwischen 1 und 9 beginnt und
    //  mit 0 - 2 Ziffern endet sowie ob die Angabe kleiner gleich dem Wert 120 ist
    if ((alter.search(/^[1-9]\d{0,2}$/) == -1) ||
        (alter > 120))
      {
        alert("Altersangabe fehlerhaft!");
        Error = true;
      }
    if (!(hobbySelected))
      {
        alert("Kein Hobby angegeben!");
        Error = true;
      }

     return !Error;

}-*/;
```

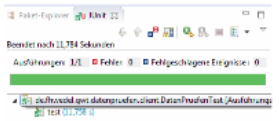
JUnit

- testen von "JavaScript"
- bewährte Test-Umgebung

Beispiel

```
package de.fhwedel.gwt.datenpruefen.client;

import org.junit.Test;
import com.google.gwt.junit.client.GWTTestCase;

public class DatenPruefenTest extends GWTTestCase {

  @Test
  public void test() {
    DatenPruefen fDatenPruefen = new DatenPruefen();

    fDatenPruefen.setSurname("");
    fDatenPruefen.setForename("Klaus");
    fDatenPruefen.setAge("42");
    fDatenPruefen.setHobbyCooking(true);
    fDatenPruefen.setHobbi();

    assertFalse(fDatenPruefen.checkAndSetInformationInput());
  }

  @Override
  public String getModuleName() {
    return "de.fhwedel.gwt.datenpruefen.DatenPruefen";
  }
}
```

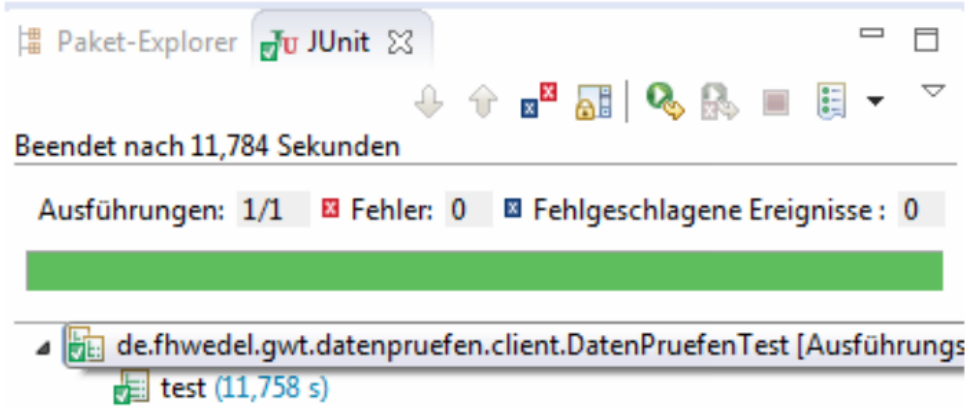
Beispiel



Beispiel

```
1.  package de.fhwedel.gwt.datenpruefen.client;
2.
3.  import org.junit.Test;
4.  import com.google.gwt.junit.client.GWTTestCase;
5.
6.  public class DatenPruefenTest extends GWTTestCase {
7.
8.    @Test
9.    public void test() {
10.     DatenPruefen fDatenPruefen = new DatenPruefen();
11.
12.     fDatenPruefen.setSurname("");
13.     fDatenPruefen.setForname("Nico");
14.     fDatenPruefen.setAge("42");
15.     fDatenPruefen.setHobbyCooking(true);
16.     fDatenPruefen.setMale();
17.
18.     assertFalse(fDatenPruefen.checkAndMarkErrornousInput());
19.   }
20.
21.   @Override
22.   public String getModuleName() {
23.     return "de.fhwedel.gwt.datenpruefen.DatenPruefen";
24.   }
25. }
```

Beispiel


Paket-Explorer
JUnit
Beendet nach 11,784 Sekunden
Ausführungen: 1/1
Fehler: 0
Fehlgeschlagene Ereignisse : 0
de.fhwedel.gwt.datenpruefen.client.DatenPruefenTest [Ausführungs
test (11,758 s)

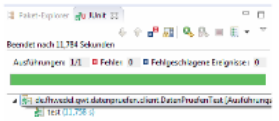
JUnit

- testen von "JavaScript"
- bewährte Test-Umgebung

Beispiel

```
package de.fhwedel.gwt.datenpruefen.client;

import org.junit.Test;
import com.google.gwt.junit.client.GWTTestCase;

public class DatenPruefenTest extends GWTTestCase {

  @Test
  public void test() {
    DatenPruefen fDatenPruefen = new DatenPruefen();

    fDatenPruefen.setSurname("");
    fDatenPruefen.setForename("Klaus");
    fDatenPruefen.setAge("42");
    fDatenPruefen.setHobbyCooking(true);
    fDatenPruefen.setHobbi();

    assertFalse(fDatenPruefen.checkAndSetInformationInput());
  }

  @Override
  public String getModuleName() {
    return "de.fhwedel.gwt.datenpruefen.DatenPruefen";
  }
}
```

Beispiel



Designer

- WYSIWYG
- komfortable Übersicht über alle Widgets
- Einbindung eigener Widgets

- Abbildung auf native HTML-Elemente

Internationalisierung

2 Arten

- Konstanten
- parametrisierte Nachrichten

2 gängige Methoden

2 Arten

- Konstanten
- parametrisierte Nachrichten

2 gängige Methoden

statisch

- Java Properties Dateien
- Interfaces, welche Methoden für den Zugriff auf die Properties bereitstellen

dynamisch

- in der HostPage wird JavaScript eingebunden
- dieser kann mittels der Dictionary-Klasse ermittelt werden

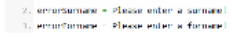
JavaScript in HTML

Java

statisch

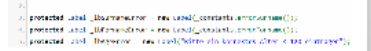
- Java Properties Dateien
- Interfaces, welche Methoden für den Zugriff auf die Properties bereitstellen

Constants.properties



Constants-Interface

Umsetzung



Constants.properties

```
1. appTitle = Input your personal data
2. errorSurname = Please enter a surname!
3. errorForname = Please enter a forname!
```

Constants-Interface

```
1.  package de.fhwedel.gwt.datenpruefen.client;
2.
3.  public interface DatenPruefenConstants extends Constants {
4.    @DefaultStringValue("Bitte einen Vornamen angeben")
5.    String errorForname();
6.
7.    @DefaultStringValue("Bitte einen Nachnamen angeben")
8.    String errorSurname();
9.
10.   @DefaultStringValue("Angaben zur Person")
11.   String appTitle();
12. }
```

Umsetzung

```
1. protected DatenPruefenConstants _constants = GWT.create(DatenPruefenConstants.class);
2. 
3. protected Label _lbSurnameError = new Label(_constants.errorSurname());
4. protected Label _lbFornameError = new Label(_constants.errorForname());
5. protected Label _lbAgeError = new Label("Bitte ein korrektes Alter < 120 eintragen");
```

dynamisch

- in der HostPage wird JavaScript eingebunden
- dieser kann mittels der Dictionary-Klasse ermittelt werden

JavaScript in HTML

```
<body>
  <script type="text/javascript">
    var DatenPruefenConstants_de = {
      appTitle: "Angaben zur Person",
      errorSurname: "Bitte einen Nachnamen eintragen!",
      errorForname: "Bitte einen Vornamen eintragen!"
    }

    var DatenPruefenConstants_en = {
      appTitle: "Input your personal data",
      errorSurname: "Please enter a surname!",
      errorForname: "Please enter a forname!"
    }
  </script>
```

Java

```
Dictionary fConstantsDict = null;
LocaleInfo fLocal = LocaleInfo.getCurrentLocale();

if (fLocal.getLocaleName().startsWith("en"))
  fConstantsDict = Dictionary.getDictionary("DatenPruefenConstants_en");
else
  fConstantsDict = Dictionary.getDictionary("DatenPruefenConstants_de");

_lbSurnameError.setText(fConstantsDict.get("errorSurname"));
_lbForenameError.setText(fConstantsDict.get("errorForname"));
```

JavaScript in HTML

```
1.  <body>
2.    <script type="text/javascript">
3.      var DatenPruefenConstants_de = {
4.        appTitle: "Angaben zur Person",
5.        errorSurname: "Bitte einen Nachnamen eintragen!",
6.        errorForname: "Bitte einen Vornamen eintragen!"
7.      }
8.
9.      var DatenPruefenConstants_en = {
10.       appTitle: "Input your personal data",
11.       errorSurname: "Please enter a surname!",
12.       errorForname: "Please enter a forname!"
13.     }
14.   </script>
```

Java

```
1.  Dictionary fConstantsDict = null;
2.  LocaleInfo fLocal = LocaleInfo.getCurrentLocale();
3.
4.  if (fLocal.getLocaleName().startsWith("en"))
5.    fConstantsDict = Dictionary.getDictionary("DatenPruefenConstants_en");
6.  else
7.    fConstantsDict = Dictionary.getDictionary("DatenPruefenConstants_de");
8.
9.  _lbSurnameError.setText(fConstantsDict.get("errorSurname"));
10. _lbFornameError.setText(fConstantsDict.get("errorForname"));
```

Internationalisierung

2 Arten

- Konstanten
- parametrisierte Nachrichten

2 gängige Methoden

Konzepte


Designer


JUnit


Internationalisierung


JSNI
JavaScript Native Interface


Server-Kommunikation


Optimierung


History

Fazit

sauberer JavaCode

- getypter Code
- gute Lesbarkeit
- keine ScriptSprache

OpenSource

- SourceCode verfügbar
- Tracking-System
- Support-Möglichkeit

Testfähigkeit

Abhängigkeit

- JavaScript-Erzeugung
- Fehleranfälligkeit

sauberer JavaCode

- getypter Code
- gute Lesbarkeit
- keine ScriptSprache

Testfähigkeit

Abhängigkeit

- JavaScript-Erzeugung
- Fehleranfälligkeit

OpenSource

- SourceCode verfügbar
- Tracking-System
- Support-Möglichkeit

Fazit

sauberer JavaCode

- getypter Code
- gute Lesbarkeit
- keine ScriptSprache

OpenSource

- SourceCode verfügbar
- Tracking-System
- Support-Möglichkeit

Testfähigkeit

Abhängigkeit

- JavaScript-Erzeugung
- Fehleranfälligkeit

Fragen?

Vielen Dank für Ihre Aufmerksamkeit

Vielen Dank für Ihre Aufmerksamkeit

Google Web Toolkit