

Projektarbeit

Erstellung einer kompletten Website
zum Thema JSONP
(JavaScript Object Notation with Padding)



JSONP.eu

Durchführungszeitraum: 01.10. bis 21.12.2012

im Rahmen einer Weiterbildung "Webprogrammierung"

online unter jsonp.eu

Helmut Karger
Genter Straße 12b
80805 München

Inhaltsverzeichnis

1	Projektbeschreibung.....	4
1.1	Auftrag.....	4
1.1.1	Anforderung.....	4
1.2	Eigene zusätzliche Anforderungen.....	4
1.2.1	Weitere Rahmenbedingungen.....	4
2	Projektplanung.....	4
2.1	Ausgangslage.....	4
2.1.1	Inhaltliche Festlegung.....	4
2.1.2	Ausgangslage des Projekts.....	4
2.2	Inhaltliche Planung.....	5
2.2.1	Seite: Same-Origin-Policy.....	5
2.2.2	Seite: Same-Origin-Policy umgehen.....	5
2.2.3	Seite: JSONP (die eigentliche Homepage).....	5
2.2.4	Seite: JSONP programmieren.....	5
2.2.5	Seite: Beispielapplikation.....	5
2.2.6	Seite: Weitere Informationen.....	6
2.2.7	Seite: Impressum.....	6
2.3	Zeitplanung.....	6
2.4	Kostenplanung.....	6
3	Projektdurchführung.....	6
3.1	Domäne.....	6
3.2	Seitengestaltung.....	7
3.2.1	Header.....	7
3.2.2	Content.....	7
3.2.3	Footer.....	7
3.2.4	Navigation.....	7
3.3	HTML & CSS.....	8
3.4	JavaScript.....	8
3.4.1	Main.js.....	8
3.4.2	Demo-Programme.....	8
3.5	PHP.....	9
3.6	Beispielanwendung.....	9
3.6.1	Fachliche Beschreibung der Anwendung.....	9
3.6.2	Technische Beschreibung der Anwendung.....	9
3.7	Marketing.....	10
4	Projektabschluss.....	11
4.1	Finaler Test.....	11
4.2	Suchmaschineneintrag.....	11
4.3	Restarbeiten.....	11
4.4	Realer Zeitaufwand.....	11
5	Anhang.....	13
5.1	Seitenlayout.....	13
5.1.1	Kopf- und Inhaltsbereich.....	13
5.1.2	Fußbereich.....	13
5.2	HTML.....	13
5.2.1	Auszug aus der index.html.....	13
5.3	CSS.....	15
5.3.1	Main.css.....	15
5.3.2	jsonp.css.....	18
5.4	JavaScript.....	19

5.4.1	Main.js (Steuerung des Hauptmenüs).....	19
5.4.2	sop.js (AJAX-Kommunikation).....	21
5.4.3	jsonp.js (JSONP-Kommunikation).....	22
5.5	PHP.....	23
5.5.1	webservice.php (AJAX-Kommunikation).....	23
5.5.2	jsonp.php (JSONP-Kommunikation).....	23
5.6	Beispielanwendung.....	24
5.6.1	Screenshot.....	24
5.6.2	JavaScript mit Google-Maps.....	24
5.6.3	PHP mit Datenbankabfrage.....	27
5.6.4	Cache-Admin-Programm.....	29
5.6.5	Screenshot.....	29
5.6.6	PHP.....	29

1 Projektbeschreibung

1.1 Auftrag

Im Rahmen einer 3 Monate dauernden Weiterbildung zum Thema "Webprogrammierung" soll eine Projektarbeit durchgeführt werden, die zum einen dem Nachweis der erworbenen Fähigkeiten und zum anderen auch zur Vorlage bei potentiellen künftigen Arbeitgebern dienen soll.

1.1.1 Anforderung

Erstellung einer kompletten Website mit mehreren Seiten unter Einbeziehung der folgenden Techniken:

- HTML
- CSS
- JavaScript
- PHP
- SQL-Datenbank

1.2 Eigene zusätzliche Anforderungen

Neben der schulischen Anforderung soll die Website im Internet unter einer eigenen Domäne veröffentlicht werden und dort dauerhaft Bestand haben. Neben dem formalen Rahmen müssen also auch die Inhalte erstellt werden. Die Site soll bei Google bekanntgemacht werden und durch Werbung die Hostingkosten wieder einspielen (Die Werbeplatzierung erfolgt ggf. erst nach Projektabschluss).

1.2.1 Weitere Rahmenbedingungen

- Verwendung von HTML 5, soweit die Tags und Attribute in allen Browsern funktionieren.
- Folgende Browser sollen unterstützt werden: Firefox, Safari, Chrome und Opera in der jeweils aktuellen Version und der Internet Explorer in der Version 8 und neuer.

2 Projektplanung

2.1 Ausgangslage

2.1.1 Inhaltliche Festlegung

Im Projekt wird eine Website zum Thema "**JSONP** – JavaScript Object Notation with Padding" realisiert. Das ist eine Web2.0-Programmieretechnik, vergleichbar mit AJAX, die es jedoch ermöglicht Daten mit einem anderen Server auszutauschen, als dem, von dem die Webseite geladen wurde.

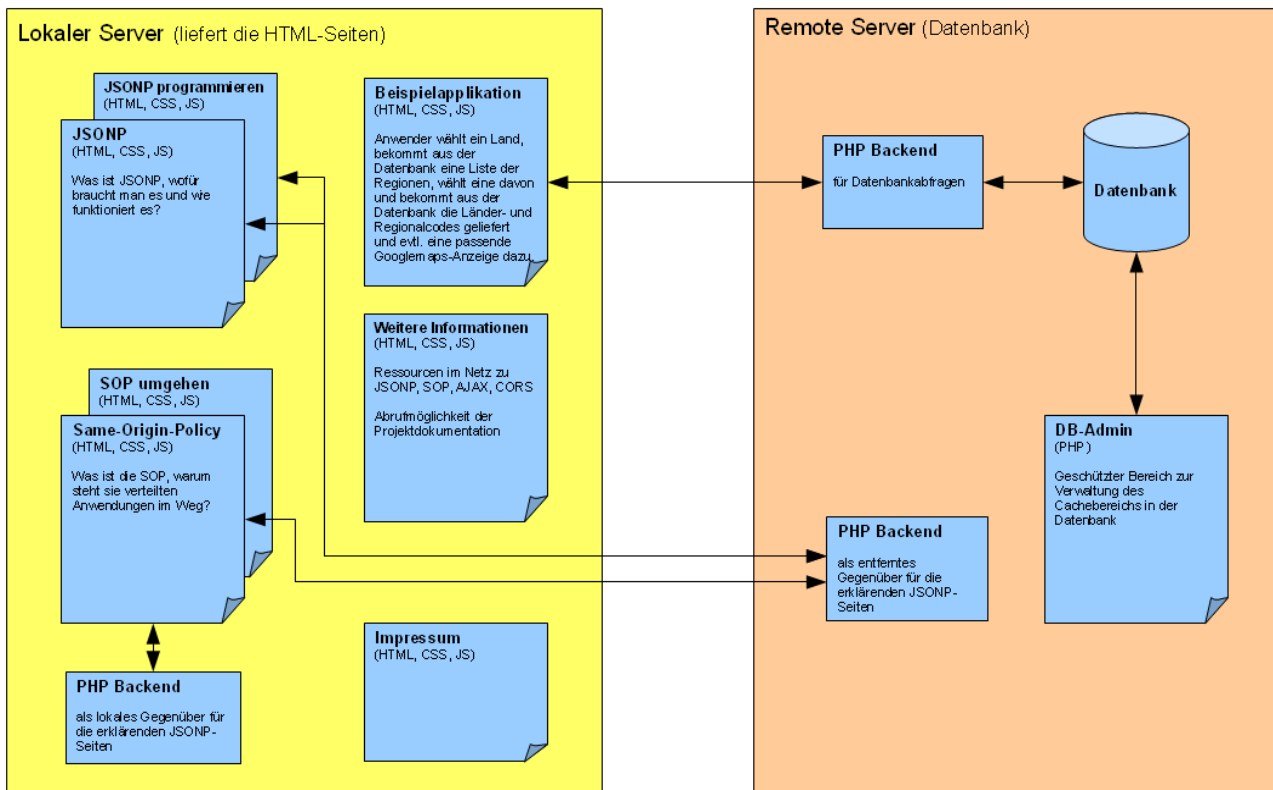
Zu JSONP gibt es einige englischsprachige Informationen im Netz, außer Wikipedia jedoch sehr wenig auf deutsch. Die Website soll Web-Programmierer ansprechen, die diese Technik einsetzen wollen. Dabei soll nicht nur erklärt werden, wie JSONP funktioniert, es soll darüber hinaus auch über das Umfeld (Same-Origin-Policy) informiert werden, in dem JSONP anzusiedeln ist. Besucher der Website sollen JSONP anhand eines funktionierenden Beispiels ausprobieren können und interessierte Programmierer sollen Codebeispiele (client- und serverseitig) im Detail erklärt bekommen.

2.1.2 Ausgangslage des Projekts

Das Projekt kann "auf der grünen Wiese" gestartet werden. Es gibt keine bestehende Website, die berücksichtigt werden muss.

2.2 Inhaltliche Planung

Folgende sieben Einzelseiten sollen realisiert werden:



2.2.1 Seite: Same-Origin-Policy

Ein Sicherheitsmechanismus im Webbrowser, der es verhindert, dass auf Inhalte zugegriffen werden kann, die nicht vom Webserver stammen, von dem die Webseite geladen wurde. Dazu ein Beispiel, das den Unterschied im Zugriff auf den eigenen und einen fremden Server veranschaulicht.

2.2.2 Seite: Same-Origin-Policy umgehen

Warum sollte die SOP überhaupt umgangen werden und was sind die technischen Möglichkeiten dazu?

2.2.3 Seite: JSONP (die eigentliche Homepage)

Was ist JSONP und wie funktioniert es. Dazu eine Demonstration, wie mit JSONP ein Datenzugriff auf einen fremden Server funktioniert, der mit AJAX nicht möglich ist.

2.2.4 Seite: JSONP programmieren

Die detaillierte Beschreibung der JavaScript-Programmierung am Client und eines PHP-Programms auf der Serverseite.

2.2.5 Seite: Beispielapplikation

Eine Beispielanwendung, die JSONP benutzt, um von einem Webservice Daten aus einer Datenbank abzurufen und Daten in die Datenbank zu schreiben.

2.2.6 Seite: Weitere Informationen

Downloadmöglichkeit dieser Projektdokumentation im PDF-Format und eine Linkliste zu Themen rund um JSONP.

2.2.7 Seite: Impressum

Obligatorische Angaben zur Verantwortlichkeit für die Site und zum Urheberrecht. Dazu eine Datenschutzerklärung.

2.3 Zeitplanung

Vorgegeben ist ein Zeitraum von 12 Wochen parallel zur Weiterbildung "Webprogrammierung". Folgende Aufwände werden geplant:

Tätigkeit	Geplante Stunden
Planung und Design	5
Erstellung des Seitenlayouts (HTML und CSS)	10
Erstellung von Logo und Grafiken	10
Erstellung eines dynamischen Navigationsmenüs (HTML, CSS und Grafiken)	12
Fachliche Recherche und Erstellung der Textinhalte	10
Erstellung der Programmierbeispiele (JavaScript und PHP)	10
Erstellung der Beispielapplikation (HTML, CSS, JavaScript, PHP und Datenbank), Befüllung der Datenbank mit initialen Daten, Schaffung der Möglichkeit, die Daten zu verwalten.	20
Inbetriebsetzung (Externe Verlinkung, Bekanntmachung bei Google), evtl. Implementation von AdSense-Werbung	4
Dokumentation	15
Summe	96

2.4 Kostenplanung

Es wird alles selbst erstellt und programmiert, deshalb fallen keine Software- oder Lizenzkosten an. Für das Webhosting und die Domänenregistrierung wird mit Jahreskosten im Rahmen von 10,- bis 20,- EUR gerechnet.

Der Hostingaufwand soll durch Werbung mittelfristig wieder eingespielt werden können.

3 Projektdurchführung

3.1 Domäne

Vor Registrierung einer Internetdomäne stellt sich die Frage nach der richtigen Schreibweise also **JSONP** oder **JSON-P**.

Eine kurze Abfrage bei Google ergibt:

"JSONP": 33.500.000 Treffer,

"JSON-P": 42.600 Treffer.

Wikipedia verwendet ebenfalls den Begriff "JSONP". Die Domänen "jsonp.de, .net, .info, und .com"

sind bereits vergeben, lediglich "jsonp.eu" ist noch verfügbar. Zur Vermeidung eines längeren, zusammengesetzten Namens wird "jsonp.eu" verwendet und diese Domäne über den Provider "all-inkl.com" registriert.

3.2 Seitengestaltung

Zielgruppe der Website sind Webprogrammierer und weniger Designer. Folglich soll eine technisch orientierte Ausrichtung zum Tragen kommen, die sich in einem schnörkellosen und reduzierten Design ausdrücken soll.

Gewählt wird ein einfaches Ein-Spalten-Layout mit fester Breite und dunkler Schrift in Arial auf hellem Grund. Die dominante Farbe ist "Grün" in folgenden Abstufungen:

hell:	#00A000
dunkler:	#007000
ganz dunkel (Background, Überschriften):	#005000
Textfeldhintergrund ist ein abgetöntes Weiß:	#F9F7ED

3.2.1 Header

Im Headerbereich soll das Logo das Thema und die Domäne repräsentieren, und ein Codefragment als Bild den technischen Charakter herausstreichen. Der Kopfbereich beinhaltet auch die Hauptnavigation.

3.2.2 Content

Im Contentbereich werden Codebeispiele, Anwendungen und Erklärungstexte in grüne Boxen mit runden Ecken eingeschlossen und so vom umlaufenden Text separiert.

3.2.3 Footer

Der Fußbereich beinhaltet neben einer weiteren Navigation für den Anwender die Auswahlmöglichkeit eines JSONP-Logos, das er sich in die eigene Homepage einbauen kann. (Siehe Kapitel Marketing)

3.2.4 Navigation

Es gibt insgesamt drei Navigationsmenüs:

- Eine Navigationsliste im Kopfbereich. Diese ist optisch ausgeblendet und dient nur Screenreadern, die mit einem javascriptgetriebenen Menü nichts anfangen können. (Stichwort Barrierefreiheit)
- Das grafische Hauptmenü. Ein Button "Weitere Seiten" öffnet bei Mausberührung ein Menü aus schräg angeordneten Reitern. Das Menü bleibt geöffnet, solange sich die Maus darüber befindet und schließt sich danach zeitverzögert automatisch wieder. Die schrägen Reiter werden durch ein grafisches Spriteimage realisiert, das aus 8 Einzelbildern besteht. Darüber wird eine transparente Grafik gelegt, die per Image-Map (verweis-sensitive Grafik) die 7 einzelnen Menübereiche definiert. Per JavaScript wird zum einen das Öffnen und Schließen des gesamten Menüs realisiert und zum anderen werden bei einer Mausbewegung über das Menü die einzelnen Menüpunkte in einer helleren Farbe hervorgehoben.
- Das Menü im Fußbereich ermöglicht die Navigation zu einer anderen Seite, ohne das Fenster zum Hauptmenü nach oben schieben zu müssen. Es ist als einfache Liste realisiert.

Darüber hinaus gibt es am unteren Ende des Contentbereichs jeweils einen Rückwärts- und Vorwärtslink, um die vorhergehende beziehungsweise nächste Seite direkt erreichen zu können.

3.3 HTML & CSS

Die Trennung von Inhalt und Formatierung wird konsequent verfolgt und alle Layout- und Formatangaben in CSS-Dateien ausgelagert. Dabei gibt es eine *main.css*, die das Stylesheet für alle HTML-Seiten darstellt und einige zusätzliche CSS-Dateien, die ergänzende oder überschreibende Angaben für bestimmte Seiten beinhalten. Alle CSS-Dateien befinden sich im Unterordner */css*.

Für die HTML-Dateien wird kein Unterordner angelegt, sie befinden sich alle im Rootverzeichnis. Das hat den Vorteil, dass alle Seiten und nicht nur die *index.html* in der URL keinen Unterverzeichnisnamen benötigen. Darüber hinaus sind die Verweise zu Grafiken, CSS- und JavaScript-Dateien in allen HTML-Dateien gleich.

Es wird HTML 5 verwendet, ohne jedoch Tags und Attribute zu verwenden, die vom Internet Explorer 8 nicht interpretiert werden können. Der IE8 muss berücksichtigt werden, da Ende 2012 Windows XP noch stark verbreitet ist und neuere Internet Explorer Versionen nicht mehr unter Windows XP laufen. Bei anderen Browsern sind lediglich die aktuellen Versionen von Interesse.

HTML und CSS werden W3C-validiert.

3.4 JavaScript

Alle JavaScript-Programme befinden sich im Unterverzeichnis */js*.

3.4.1 Main.js

Die *main.js* beinhaltet die JavaScript-Logik für das animierte Hauptmenü. Die Events "onMouseOver" und "onClick" auf den Button "Weitere Seiten" veranlasst das Ausfahren eines horizontalen Balkens und danach das Aufschieben der Reiterleiste nach rechts oben. Nun können die Menüpunkte geklickt werden. Nach Verlassen des Menübereichs mit der Maus, schließt sich das Menü zeitverzögert in umgekehrter Reihenfolge und der Button "Weitere Seiten" wird wieder eingeblendet. Die Zeit und Bewegungssteuerung wird dabei durch JavaScript veranlasst. Das Objekt "animation" enthält die gesamte Logik für das Hauptmenü. Die Funktion "tabOver" regelt darüber hinaus die Ansteuerung des Spriteimages in Abhängigkeit von der Mausposition im Menü.

Hinweis: Für derartige Animationen empfiehlt sich die Verwendung eines JavaScript-Frameworks, zum Beispiel jQuery – das minimiert den Programmieraufwand wesentlich. In diesem Projekt wurde auf die Verwendung von Frameworks jedoch verzichtet.

In der *main.js* befindet sich auch die Logik für die Iconauswahl im Fußbereich. Durch Anklicken eines Radiobuttons zu einer Grafik passt eine JavaScript-Funktion den HTML-Code an, der dem Anwender in einem Rahmen zur Übernahme auf dessen Website angeboten wird.

3.4.2 Demo-Programme

Die Seiten zum Thema *Same-Origin-Policy* und *JSONP* beinhalten Demonstrationsprogramme, damit der Besucher der Website den AJAX-Datenverkehr zum eigenen und einem fremden Server, sowie JSONP zu einem fremden Server direkt ausprobieren kann. Dabei wird auf der Clientseite per JavaScript eine AJAX- bzw. JSONP-Anfrage abgeschickt und vom Server mit einem kurzen Textstring beantwortet, der die IP-Adresse des Clients und die aktuelle Uhrzeit enthält. Die Uhrzeit deshalb, damit bei mehrfachem Auslösen der Anfrage eine Antwort von der vorhergehenden zu unterscheiden ist und der Browser die Serverantwort nicht cached. Die genaue Funktion des JSONP-Abfrageprogramms wird auf der Seite *JSONP programmieren* erläutert.

3.5 PHP

Im Rahmen der oben erwähnten Demo-Programme wird PHP als serverseitiges Gegenüber für die aufrufenden JavaScript-Programme verwendet. Diese Programme geben kein HTML aus, sondern arbeiten als **Webservice**, der auf Anfrage ein Datenergebnis zurückreicht.

Von der Lokation her wird zwischen **eigenen** und **fremden** Webservices unterschieden. Bei gleicher Funktion liegen **eigene Webservices** auf demselben Server, von dem die HTML-Seite geladen wurde (hier: im Unterverzeichnis /php von jsonp.eu) und **fremde Webservices** auf webservice.helmutkarger.de im Unterverzeichnis /php.

Der JSONP-Webservice wird ebenso wie der korrespondierende JavaScript-Teil auf der Seite *JSONP programmieren* näher beschrieben.

3.6 Beispielanwendung

Die Beispielanwendung veranschaulicht an einem Realbeispiel die **JSONP-Funktionalität**. Über eine Webseite ruft ein Anwender interaktiv Daten von einem fremden Webservice ab. Die Abfragen erfolgen im Hintergrund, ein Neuladen der Webseite erfolgt dabei nicht.

3.6.1 Fachliche Beschreibung der Anwendung

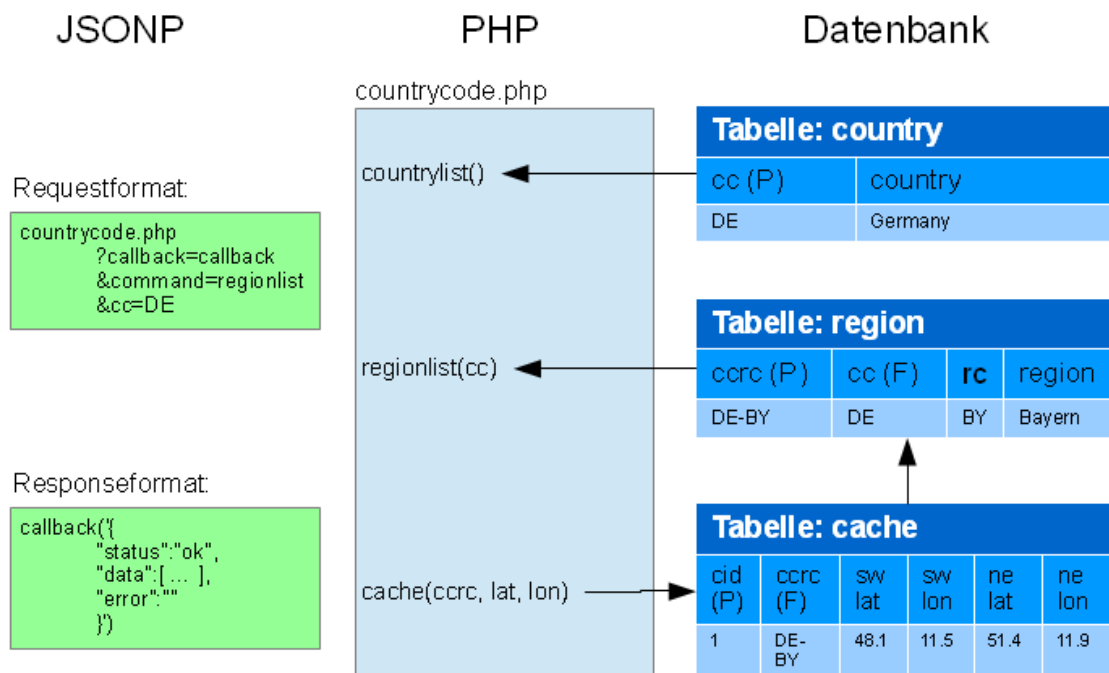
Der Anwender wählt in einem Select-Feld ein Land seiner Wahl (z.B. "Germany") aus und danach in einem zweiten Selectfeld eine Region aus diesem Land (z.B. "Bayern"). Daraufhin bekommt er den ISO 3166-1 Ländercode (two-letter-country-code) und den ISO 3166-2 Regionalcode (country-subdivision-code) ausgegeben (z.B. "DE-BY"). Dazu erfolgt eine Anzeige der geografischen Region in einer Google-Map.

3.6.2 Technische Beschreibung der Anwendung

Nach dem Laden der Webseite von *jsonp.eu* wird automatisch im Hintergrund das Nachladen der Länderliste initiiert. Die Länderliste liegt in einer Datenbank auf dem Server *webservice.helmutkarger.de* – der Zugriff über die Domänengrenzen hinweg erfolgt per JSONP. Nachdem der Anwender ein Land selektiert hat, erfolgt eine erneute JSONP-Abfrage, diesmal nach den Regionen des ausgewählten Landes und, falls im Datenbank-Cache vorhanden, nach den Geo-Koordinaten der Region. Sollten die Koordinaten im Cache nicht verfügbar sein, so fragt der Client diese bei Google-Maps an und schreibt sie (wiederum per JSONP) in den Datenbank-Cache.

Die Datenbank enthält 243 Ländereinträge und insgesamt 3786 Regionen.

Hier eine Übersicht der Datenbanktabellen (rechts), der JSONP-Aufruf und -Antwortformate und der PHP-Funktionen, die über das Statement "command" im JSONP-Request angesprochen werden können:



In der Cachetabelle werden zwei Koordinaten gespeichert: jeweils die geografische Länge und Breite des südwestlichen und nordöstlichen Punktes eines Rechtecks, welches die Region umschließt. Der Cachebereich wird regelmäßig gelöscht. Das hat rechtliche Gründe – Google gestattet in seinen Nutzungsbedingungen für Google-Maps nicht, dass Google-Maps-Daten abgezogen und in eigenen Datenbanken gespeichert werden.

Zur Cachepflege wurde ein eigenes PHP-Programm entwickelt. Das Programm liegt auf dem Datenbankserver und ist passwortgeschützt. Es ermöglicht die seitenweise Anzeige aller Cache-Datensätze und das Löschen selektierter oder aller Cacheeinträge. (Siehe im Anhang: Cache-Admin-Programm.)

3.7 Marketing

Einnahmen aus dieser Website können nur erwirtschaftet werden, wenn **Werbung** platziert und die Site **bei Google gut gerankt** ist und damit viele Besucher zu erwarten sind. Folgende Marketingmaßnahmen werden realisiert:

- Eintrag der Site bei *dmoz.org* und bei **Google** über die *Google Webmaster Tools*
- Verlinkung der Site über andere Sites (z.B. Wikipedia)
- Anbieten eines **JSONP-Icons** (mit Backlink) als Give-Away für andere Web-Programmierer, die JSONP auf ihrer Seite einsetzen, mit dem Ziel zahlreicher Rückverlinkungen
- Einbindung von **Google AdSense**
- Eintrag bei der **VG-Wort**

4 Projektabschluss

4.1 Finaler Test

Bei der Seitenerstellung wird nicht auf die Unzulänglichkeiten des Internet Explorer 6 eingegangen (zum Beispiel keine transparenten PNG-Grafiken). Die Seiten und Applikationen wurden erfolgreich mit folgenden Browsern getestet:

- Internet Explorer 7, 8 und 9 (Einschränkung: keine runden Ecken)
- Firefox 8, 13, 15, 16 und 17
- Chrome 23
- Opera 12
- Safari 5
- Konqueror 4 (Einschränkung: leichter Überlauf im Fußbereich)

dabei wurde auf folgende Testkriterien geachtet:

- Richtige Darstellung des Seitenlayouts insbesondere der Floatbereiche und der Hintergrundgrafiken
- Funktion des animierten Hauptmenüs
- Funktion der AJAX-Kommunikation, auch bei mehrfachem Aufruf
- Funktion der JSONP-Kommunikation, auch bei mehrfachem Aufruf
- Funktion der Beispielapplikation und der Google-Maps-Darstellung

4.2 Suchmaschineneintrag

Eine Woche nach Einreichung einer Sitemap über die Google Webmaster-Tools, taucht die Site für den Suchbegriff "jsonp" erstmalig an Position 29 in der deutschen Google-Suche auf. Alle sieben Seiten sind im Google Index vorhanden und es besteht die Hoffnung, über die Zeit eine bessere Position zu erreichen. Die Aufnahme in *dmoz.org* ist – wie erwartet – eine Woche nach Einreichung noch nicht erfolgt.

4.3 Restarbeiten

Die Einbindung der Zählpixel für die VG-Wort (Verwertungsgesellschaft Wort) und die Schaltung erster Google-AdSense-Werbeinblendungen ist erst sinnvoll, wenn die schulische Entwicklungsumgebung aufgelöst ist und Zugriffe auf den Inhalt nur noch über die Domäne *jsonp.eu* erfolgen. Geplant ist das für die KW 51.

4.4 Realer Zeitaufwand

Das Projekt wurde zeitgerecht fertiggestellt und findet seinen Abschluss mit einer Präsentation in KW 51. Die Zeitaufwände stellen sich wie folgt dar:

Datum	Stunden	Tätigkeit
KW 41	3,0	Konzeptentwicklung, Abstimmung mit Auftraggeber, erstes Schaubild
	3,0	Anlage der Verzeichnisstruktur, HTML-Template, CSS; JSONP vs. JSON-P; Farbgestaltung, Anlage weiterer Dokumentationen
KW 42	5,0	Domainbestellung, Logodesign, Hintergründe und Navigationsmenü
	3	CSS-Nachjustierungen, Überschriften, Fußzeile, Ideensammlung für den Inhalt der Seiten
KW 43	7,0	HTML, CSS, JS, PHP für die Seite <i>SOP</i> , dazu Wireshark-Aufzeichnung und ein Teil des Textes
	3,0	<i>SOP</i> Seite Inhaltstext, erneute Anpassung der Überschriften
KW 44	5,0	<i>JSONP</i> Webservice und JavaScript, <i>JSONP</i> -Seiteninhalt
KW 45	4,0	Inhalt zur Programmierungs-Seite, Schaubild Serverbild, <i>SOP</i> -Inhalte auf zwei HTML-Seiten aufteilen, Favicon, Codeimage für den Header

	3,0	Dokumentation
KW 46	7,0	Navigation neu
	2,0	Dokumentation
	2,0	Menüanimation
	2,0	Gestaltung und Inhalt des Footers
	0,5	Gestaltung Navigation
	0,5	Codedokumentation HTML und CSS
KW 47	1,0	Generierung von JSONP-Icons
	1,5	Menüanimation
	3,0	Seiten <i>Impressum</i> und <i>Weitere Informationen</i>
KW 48	1,0	Test in verschiedenen Browsern und kleinere Anpassungen
	2,0	Datenbankdesign, Tabellen anlegen, XML-Dateien importieren
	0,5	Text "Exkurs AJAX" auf <i>SOP</i> ergänzen
	2,5	remote JSONP-PHP Programm
KW 49	3,0	PHP und Datenbankentwicklung für die Beispielapplikation
	3,0	Vervollständigung und Validierung aller Seiten bis auf <i>appl</i>
	1,5	Eintrag der Site bei dmoz.org und Google und weitere Verlinkungen
	4,0	Applikationsentwicklung JavaScript
	5,0	Applikationsentwicklung JavaScript
	1,5	Dokumentation
	1,5	Text für die Beispielapplikation
KW 50	4,5	<i>Cache-Admin</i> -Seite
	7,0	Dokumentation und Präsentationsvorbereitung
KW 51	3,0	Präsentation
Summe	95,5	

5 Anhang

5.1 Seitenlayout

5.1.1 Kopf- und Inhaltsbereich

mit Hauptnavigation und Bereichen für Applikationstest und Exkurs

The screenshot shows the top part of the JSONP.eu website. At the top left, the text 'JSONP.eu' is displayed in large, green, 3D-style letters. Below it is a navigation menu with seven green arrow-shaped buttons pointing to the right, labeled: 'Same-Origin-Policy', 'SOP umgehen', 'JSONP', 'JSONP programmieren', 'Beispiel', 'Weitere Informationen', and 'Impressum'. To the right of the menu, there is a snippet of JavaScript code for a function named 'jsonp'. Below the navigation, the main heading reads 'JavaScript Object Notation with Padding'. The text below explains that it is not possible to access a server via AJAX due to the Same-Origin Policy. A section titled 'Ausprobieren:' contains a form with a button 'IP-Adresse abfragen' and an input field 'Antwort:'. To the right of the form, there is an explanatory text about using JSONP instead of AJAX. Below this, there is a section 'Was ist JSONP?' and an 'Exkurs: Was ist eigentlich ...' section with a sub-heading 'JSONP?'.

5.1.2 Fußbereich

mit unterer Navigation, Iconauswahl und Copyright

The screenshot shows the footer of the JSONP.eu website. It features a navigation bar with links '<< SOP umgehen' and 'JSONP programmieren >>'. Below this is a section titled 'Seitenüberblick:' with a list of links: 'Same-Origin-Policy', 'SOP umgehen', 'JSONP (Home)', 'JSONP programmieren', 'Beispiel', 'Weitere Informationen', and 'Impressum'. To the right of the links is a section titled 'Sie verwenden JSONP auf Ihrer eigenen Website?' with a copyright notice '(c) 2012 by Helmut Karger'. Below this section is a code block showing the HTML code for the JSONP icon: ''. The icon itself is a small green square with the text 'JSONP' inside.

5.2 HTML

5.2.1 Auszug aus der index.html

```
<html>
<head>
  <meta charset='utf-8' />
  <title>JSONP</title>
  <meta name="description" content="Die JavaScript Object Notation with Padding ermöglicht im Gegensatz zu AJAX den Datenzugriff auf fremde Server. Eine wichtige Web 2.0 Technologie." />
  <meta name="keywords" content="JavaScript Object Notation with Padding, JSONP, AJAX, SOP" />
  <meta name="author" content="Helmut Karger" />
```

```

<link rel="shortcut icon" href="img/favicon.ico" type="image/x-icon" />
<link rel="stylesheet" type="text/css" href="css/main.css" />
<link rel="stylesheet" type="text/css" href="css/jsonp.css" />
</head>
<body>
<div id="wrapper"> <!-- Zentrierter Bereich mit Inhalt -->
<div id="header"> <!-- Header mit Logo, einem Image und Navigationsmenüs -->

<h1>JSONP</h1> <!-- wird ausgeblendet und durch Logo ersetzt -->
<div id="nav"> <!-- Klassische Navigation wird ausgeblendet und durch animierte Navigation
ersetzt -->
<ul>
<li><a href="sop.html">Same-Origin-Policy</a></li>
<li><a href="sop-umgehen.html">SOP umgehen</a></li>
<li><a href="index.html">JSONP</a></li>
<li><a href="programmierung.html">JSONP programmieren</a></li>
<li><a href="appl.html">Beispiel</a></li>
<li><a href="weiteres.html">Weitere Informationen</a></li>
<li><a href="impressum.html">Impressum</a></li>
</ul>
</div> <!-- Ende nav -->
<div id="menuspace"> <!-- Raum für die animierte Navigation -->
<div id="tabmenu"> <!-- Navigation aus ImageMap und Spritesets -->
 <!-- ImageMap
transparent -->
<map name="menumap">
<area id="tab1" shape="poly" onmouseover="tabOver(1)" onmouseout="tabOver(0)" alt="menutab"
coords="0,99,98,2,100,0,173,0,178,4,176,9,88,99" href="sop.html" />
<area id="tab2" shape="poly" onmouseover="tabOver(2)" onmouseout="tabOver(0)" alt="menutab"
coords="89,99,177,9,179,4,178,0,253,0,258,4,256,9,168,99" href="sop-umgehen.html" />
<area id="tab3" shape="poly" onmouseover="tabOver(3)" onmouseout="tabOver(0)" alt="menutab"
coords="169,99,257,9,259,4,258,0,333,0,338,4,336,9,248,99" href="index.html" />
<area id="tab4" shape="poly" onmouseover="tabOver(4)" onmouseout="tabOver(0)" alt="menutab"
coords="249,99,337,9,339,4,338,0,413,0,418,4,416,9,328,99" href="programmierung.html" />
<area id="tab5" shape="poly" onmouseover="tabOver(5)" onmouseout="tabOver(0)" alt="menutab"
coords="329,99,417,9,419,4,418,0,493,0,498,4,496,9,408,99" href="appl.html" />
<area id="tab6" shape="poly" onmouseover="tabOver(6)" onmouseout="tabOver(0)" alt="menutab"
coords="409,99,497,9,499,4,498,0,573,0,578,4,576,9,488,99" href="weiteres.html" />
<area id="tab7" shape="poly" onmouseover="tabOver(7)" onmouseout="tabOver(0)" alt="menutab"
coords="489,99,577,9,579,4,578,0,653,0,658,4,656,9,568,99" href="impressum.html" />
</map>
</div>
<div id="button" onmouseover="animation.show()" onclick="animation.show()"></div> <!--
Menübutton -->
<div id="balken"></div>
</div> <!-- Ende menuspace -->
<h2>JavaScript Object Notation with Padding</h2> <!-- Seitenüberschrift -->
</div>
<div id="content"> <!-- Contentbereich -->
...
<p class="previous"><a href="sop-umgehen.html">&lt;&lt; SOP umgehen</a></p>
<p class="next"><a href="programmierung.html">JSONP programmieren &gt;&gt;</a></p>
<div class="stopfloat"></div>
</div>
<div id="footer">
<div id="footernav"> <!-- Weitere Seitennavigation im Fußbereich -->
<ul>
<li><h6>Seitenüberblick:</h6></li>
<li><a href="sop.html">Same-Origin-Policy</a></li>
<li><a href="sop-umgehen.html">SOP umgehen</a></li>
<li><a href="index.html">JSONP (Home)</a></li>
<li><a href="programmierung.html">JSONP programmieren</a></li>
<li><a href="appl.html">Beispiel</a></li>
<li><a href="weiteres.html">Weitere Informationen</a></li>
<li><a href="impressum.html">Impressum</a></li>
</ul>
</div> <!-- Ende footernav -->
<div id="iconauswahl"> <!-- Auswahl eines JSONP-Icons -->
<form>

<input type="radio" name="icon" onClick="imageSelect('jsonp1.png')" checked="checked" /><br
/>

<input type="radio" name="icon" onClick="imageSelect('jsonp2.png')" /><br />

  <input type="radio" name="icon" onClick="imageSelect('jsonp3.png')"/><br />
  
  <input type="radio" name="icon" onClick="imageSelect('jsonp4.png')"/><br />
  
  <input type="radio" name="icon" onClick="imageSelect('jsonp5.png')"/><br />
  
  <input type="radio" name="icon" onClick="imageSelect('jsonp6.png')"/><br />
</form>
</div> <!-- Ende iconauswahl -->
<div id="codeabruf"> <!-- Darstellung des HTML-Codes für das JSONP-Icon -->
<h6>Sie verwenden JSONP auf Ihrer eigenen Website?</h6>
<p>Dann zeigen Sie es Ihren Besuchern und platzieren Sie ein <strong>JSONP</strong>-Icon
auf Ihrer Seite (einfach ein Icon auswählen und den angezeigten
HTML-Code in die eigene Seite kopieren):
</p><br />
<div id="codeanzeige"></div>
</div> <!-- Ende codeabruf -->
<p id="copyright">(c) 2012 by Helmut Karger</p>
</div>
</div>
<script src="js/main.js" type="text/javascript"></script>
<script src="js/jsonp.js" type="text/javascript"></script>
</body>
</html>

```

5.3 CSS

5.3.1 Main.css

```

/* MAIN.CSS für JSONP.EU */
/* Einstellungen für alle Seiten */

/* Reset der Ränder */
* {
  margin: 0;
  padding: 0;
}

body {
  background-color: #005000; /* Body wird zum Hintergrund */
  background-image: url("../img/bodybg.png");
  background-repeat: repeat-x;
  font-family: arial, sans-serif;
  font-size: 14px;
}

#wrapper {
  width: 900px; /* Zentrierter Inhaltsbereich */
  margin: 0 auto;
  margin-bottom: 20px;
  background-color: #F9F7ED;
}

/* Headerbereich und Navigationsmenü */
#header {
  position: relative; /* gesamter Header */
  background-image: url("../img/headerbg.png");
  background-repeat: repeat-x;
}

#header h1 {
  background-image: url("../img/logo.png"); /* Textüberschrift wird ausgeblendet und durch Logo ersetzt */
  background-repeat: no-repeat;
  background-position: 10px 0;
  height: 150px;
  text-indent: -9999px;
}

#header h2 {
  padding: 90px 20px 10px 20px; /* Seitenüberschrift */
}

```

```
    color: #005000;
    font-size: 23px;
}

#skriptauszug {                               /* Skriptgrafik rechts oben */
    position: absolute;
    right: 5px;
    top: 5px;
}

#nav {                                          /* wird ausgeblendet und durch animiertes Menü ersetzt */
    height: 0;
    text-indent: -9999px;
}

#nav ul {                                      /* Aufzählungspunkte entfernen */
    list-style-type: none;
}

#menuspace {                                   /* Bereich des animierten Menüs */
    position: absolute;
    top: 110px;
    width: 780px;
    height: 115px;
}

#tabmenu {                                     /* Menü mit Tabs */
    position: absolute;
    left: 120px;
    width: 660px;
    height: 100px;
    background-image: url(../img/menusprite.png);
    background-repeat: no-repeat;
    background-position: 0 0;
    overflow: hidden;                          /* für Opera, damit das Menü nicht nach unten herausragt */
}

#tabmenu img {                                /* nur so hoch wie parent (inherit funktioniert nicht im
IE)*/
    border: none;                              /* IE: blauen Rahmen um img entfernen */
}

#button {                                      /* Menü Button */
    position: absolute;
    top: 100px;
    left: 1px;
    width: 200px;
    height: 0;
    background-image: url(../img/button.png);
    background-repeat: no-repeat;
    background-position: 0 0;
}

#balken {                                      /* horizontaler Menübalken */
    position: absolute;
    top: 100px;
    width: 690px;
    height: 15px;
    background-image: url(../img/balken.png);
    background-repeat: repeat-x;
    background-position: 0 0;
}

/* Contentbereich */
#content {
    padding: 20px 20px 5px 20px;
    background-color: #F9F7ED;
}

#content p , #content li {
    margin-bottom: 10px;
    line-height: 20px;
}

#content h3, #content h4 {
    color: #005000;
    margin: 25px 0 6px 0;
}
```



```
#content h3 {
    font-size: 17px;
}

#content h4 {
    font-size: 14px;
}

#content ul, #content ol {
    margin-left: 20px;
    margin-bottom: 10px;
}

#content strong {
    color: #00a000;
}

.previous {                                /* Link zur vorhergehenden Seite */
    float: left;
    margin-top: 20px;
}

.next {                                    /* Link zur nächsten Seite */
    float: right;
    margin-top: 20px;
}

/* Darstellungsboxen */
.rightbox {                                /* rechte grüne Box: Exkurs */
    float: right;
    width: 450px;
    margin: 0 0 2px 8px;
    border: solid 2px #007000;
    font-size: 12px;
    color: #007000;
    background-color: #F9F7ED;
    -webkit-border-radius: 8px;
    -moz-border-radius: 8px;
    border-radius: 8px;
}

.leftbox {                                 /* linke grüne Box: Ausprobieren oder Codeschnipsel */
    float: left;
    display: inline-block;
    margin: 0 8px 2px 0;
    border: solid 2px #00A000;
    font-size: 11px;
    color: #00A000;
    background-color: #F9F7ED;
    -webkit-border-radius: 8px;
    -moz-border-radius: 8px;
    border-radius: 8px;
}

.rightbox h5 {                             /* Überschrift im grünen oberen Rand */
    font-size: 10px;
    letter-spacing: 2px;
    color: #F9F7ED;
    background-color: #007000;
    padding: 2px 6px 2px 6px;
}

.leftbox h5 {                              /* Überschrift im grünen oberen Rand */
    font-size: 10px;
    letter-spacing: 2px;
    color: #F9F7ED;
    background-color: #00A000;
    padding: 2px 6px 2px 6px;
}

.rightbox h6, .leftbox h6 {               /* Überschriften innerhalb der Box */
    font-size: 12px;
}

.leftbox div, .rightbox div {            /* DIVs innerhalb der Box */
    overflow: auto;
    padding: 2px 6px 6px 6px;
}
```

```
.stopfloat {                                /* Beendet Float */
  clear: both;
}

/* Footer */
#footer {
  height: 130px;
  background-color: #bbbbbb;
  border-top: solid 1px black;
  padding: 4px 20px;
  font-size: 11px;
}

#footernav {                                /* weitere Seitennavigation im Fußbereich */
  width: 180px;
  float: left;
}

#footernav ul {                             /* Aufzählungspunkte entfernen */
  list-style-type: none;
}

#footernav ul li{
  padding-bottom: 2px;
}

#footer h6 {
  font-weight: bold;
  font-size: 11px;
}

#iconauswahl {                              /* Auswahl eines JSONP-Icons */
  width: 120px;
  float: left;
  margin-top: 10px;
}

#codeabruf {                                /* Spalte für die Codeanzeige */
  width: 350px;
  float: left;
}

#codeanzeige {                              /* Darstellung des HTML-Codes für das JSONP-Icon */
  width: 350px;
  height: 42px;
  padding: 6px;
  border: 1px solid black;
  color: #005000;
}

#copyright {                                /* Copyright */
  text-align: right;
}
```

5.3.2 jsonp.css

```
/* Einstellungen für INDEX.HTML */

#jsonp_test {                               /* Anwendungsbox */
  width: 440px;
}
#jsonp_test p {
  margin-bottom: 0;
}

#jsonp_antwort {                            /* Anzeigebereich der Antwort */
  float:right;
  width: 220px;
  padding: 2px;
  min-height: 18px;
  border: solid 1px black;
}

#jsonp_button {                             /* Absendebutton */
  float: left;
  width: 140px;
```

```

}

#jsonp_p_antwort {
    line-height: 22px;
    margin-right: 228px;
    text-align: right;
    font-size: 13px;
}

```

5.4 JavaScript

5.4.1 Main.js (Steuerung des Hauptmenüs)

```

// JavaScript-Funktionen für alle Seiten
// für Menü im Header und Iconauswahl im Fußbereich
// (c) 2012 Helmut Karger

function imageSelect(image) {
    // Darstellung eines Icon-Abruf-Codes im
    // Fußbereich
    ausgabe = "&lt;a href='http://jsonp.eu'&gt;"
    + "&lt;img src='http://jsonp.eu/img/'+" + image
    + "' width='80px' height='15px' alt='enabled by JSONP'"
    + " title='enabled by JSONP' /&gt;&lt;/a&gt;";
    // HTML-Code zusammensetzen
    document.getElementById("codeanzeige").innerHTML = ausgabe;
    // und ausgeben
}

animation = {
    // Animation des Hauptmenüs
    tabmenu: {
        open: {height: 100, top: 0, left: 120},
        // Dimensionen
        closed: {height: 0, top: 100, left: 20},
        current: {height: 100, top: 0, left: 120},
        step: {height: 3, top: 3, left: 3},
        intervall: 10,
        show: function() {
            // Anzeigen
            // Zähler erhöhen
            this.current.height += this.step.height;
            if(this.current.height > this.open.height) {
                this.current.height = this.open.height;
                // bis Maximum
            }
            this.current.top -= this.step.top;
            // Zähler vermindern
            if(this.current.top < this.open.top) {
                this.current.top = this.open.top;
                // bis Minimum
            }
            this.current.left += this.step.left;
            // Zähler erhöhen
            if(this.current.left > this.open.left) {
                this.current.left = this.open.left;
                // bis Maximum
            }
            if(this.current.height == this.open.height &&
                this.current.top == this.open.top &&
                this.current.left == this.open.left) {
                // Ende Bedingung (ist Menü ganz offen?)
                clearInterval(this.timer);
                // Intervall-Timer löschen
                animation.close.active = true;
                // Menü schließen ermöglichen
                if(!animation.mouseIn) {
                    // wenn die Maus außerhalb des Menüs ist
                    animation.close.start();
                    // Close TimeOut starten
                }
            }
            document.getElementById("tabmenu").style.height = this.current.height+"px";
            // CSS ändern
            document.getElementById("tabmenu").style.top = this.current.top+"px";
            document.getElementById("tabmenu").style.left = this.current.left+"px";
        },
        hide: function() {
            // Einklappen
            animation.close.stop();
            // Close TimeOut stoppen
            this.current.height -= this.step.height;
            // Zähler vermindern
            if(this.current.height < this.closed.height) {
                this.current.height = this.closed.height;
                // bis Minimum
            }
            this.current.top += this.step.top;
            // Zähler erhöhen
            if(this.current.top > this.closed.top) {
                this.current.top = this.closed.top;
                // bis Maximum
            }
            this.current.left -= this.step.left;
            // Zähler vermindern
            if(this.current.left < this.closed.left) {
                this.current.left = this.closed.left;
                // bis Minimum
            }
            if(this.current.height == this.closed.height &&
                // Ende Bedingung (ist Menü ganz zu?)

```

```

    this.current.top == this.closed.top &&
    this.current.left == this.closed.left) {
        clearInterval(this.timer); // Intervall-Timer löschen
        animation.balken.timer = setInterval(function() { // Balken schließen anstoßen
            animation.balken.hide();
        }, animation.balken.intervall);
    }
    document.getElementById("tabmenu").style.height = this.current.height+"px"; // CSS ändern
    document.getElementById("tabmenu").style.top = this.current.top+"px";
    document.getElementById("tabmenu").style.left = this.current.left+"px";
}
},
balken: { // Horizontal ausfahrender Balken
    open: {width: 690}, // Dimensionen
    closed: {width: 0},
    current: {width: 690},
    step: {width: 15},
    intervall: 10,
    show: function() { // Anzeigen
        animation.button.hide();
        this.current.width += this.step.width; // Zähler erhöhen
        if(this.current.width > this.open.width) {
            this.current.width = this.open.width; // bis Maximum
        }
        if(this.current.width == this.open.width) { // Ende Bedingung
            clearInterval(this.timer); // Intervall-Timer löschen
            animation.tabmenu.timer = setInterval(function() { // TabMenü ausfahren anstoßen
                animation.tabmenu.show();
            }, animation.tabmenu.intervall);
        }
        document.getElementById("balken").style.width = this.current.width+"px"; // CSS ändern
    },
    hide: function() { // Einklappen
        if(this.current.width < 400) animation.button.show()
        this.current.width -= this.step.width; // Zähler vermindern
        if(this.current.width < this.closed.width) {
            this.current.width = this.closed.width; // bis Minimum
        }
        if(this.current.width == this.closed.width) { // Ende Bedingung
            clearInterval(this.timer); // Intervall-Timer löschen
            animation.closed = true; // Menü eingeklappt
        }
        document.getElementById("balken").style.width = this.current.width+"px"; // CSS ändern
    }
},
button: { // Button: "Weitere Seiten >>"
    open: {height: 35, top: 65}, // Dimensionen
    closed: {height: 0, top: 100},
    current: {height: 0, top: 100},
    step: {height: 2, top: 2},
    show: function() { // Anzeigen
        this.current.height += this.step.height; // Zähler erhöhen
        if(this.current.height > this.open.height) {
            this.current.height = this.open.height; // bis Maximum
        }
        this.current.top -= this.step.top; // Zähler vermindern
        if(this.current.top < this.open.top) {
            this.current.top = this.open.top; // bis Minimum
        }
        document.getElementById("button").style.height = this.current.height+"px"; // CSS ändern
        document.getElementById("button").style.top = this.current.top+"px";
    },
    hide: function() { // Einklappen
        this.current.height -= this.step.height; // Zähler vermindern
        if(this.current.height < this.closed.height) {
            this.current.height = this.closed.height; // bis Minimum
        }
        this.current.top += this.step.top; // Zähler erhöhen
        if(this.current.top > this.closed.top) {
            this.current.top = this.closed.top; // bis Maximum
        }
        document.getElementById("button").style.height = this.current.height+"px"; // CSS ändern
        document.getElementById("button").style.top = this.current.top+"px";
    }
},
closed: false, // gesamtes Menü ist geschlossen
mouseIn: false, // Maus steht im Menü
show: function() { // ganzes Menü ausfahren
    if(this.closed) { // nur wenn geschlossen

```

```

    this.closed = false;
    animation.balken.timer = setInterval(function() { // Balken ausfahren veranlassen per Intervall
        animation.balken.show()
    }, animation.balken.intervall);
},
hide: function() { // ganzes Menü einfahren
    animation.tabmenu.timer = setInterval(function() {
        animation.tabmenu.hide() // TabMenü einfahren veranlassen per Intervall
    }, animation.tabmenu.intervall);
},
close: { // Autoclose des Menüs nach Timeout
    start: function() { // Timeout starten
        clearTimeout(this.timeout); // evtl. bestehendes T/O löschen
        this.timeout = setTimeout(function() { // neuen T/O anlegen
            animation.hide(); // zum Schließen
        }, 2000); // nach Zeitablauf
    },
    stop: function() { // Timeout abbrechen
        clearTimeout(this.timeout); // (wenn Mauszeiger erneut ins Menü geführt
// wird
    }
}

function tabOver(tab) { // Reagiert auf Mausbewegungen über der
// ImageMap
    pix = tab*-100; // vertikalen Abstand festlegen
    document.getElementById("tabmenu").style.backgroundPosition = "0 "+pix+"px"; // und SpriteSet
// positionieren
    if(tab == 0) { // beim Verlassen einer ImageMap Region
        animation.mouseIn = false; // Maus ist außerhalb des Menüs
        animation.close.start(); // Timer für Schließen des Menüs starten
    } else { // beim Eintreten in eine ImageMap Region
        animation.mouseIn = true; // Maus ist im Menü
        animation.close.stop(); // Timer stoppen
    }
}

// sofort nach Laden der Seite ausführen:
setTimeout(function() { // Menü 2 sec nach Laden der Seite schließen
    animation.close.start();
}, 2000);

imageSelect("jsonp1.png"); // Im Fußbereich ein Icon vorauswählen

```

5.4.2 sop.js (AJAX-Kommunikation)

```

// JS-Funktionen für einen AJAX-Aufruf zum 'eigenen' Server
// (Server, von dem die HTML-Seite bezogen wurde)
function ajax1() {
    ausgabe = document.getElementById('sop_antwort1');
    try { // ein XMLHttpRequest Object anlegen
        xmlhttp = new XMLHttpRequest();
    }
    catch(e) { // Fehler: veralteter Browser IE6 o. älter
        ausgabe.innerHTML = "<p>Sorry, Ihr Browser versteht keinen XMLHttpRequest.</p>"
    }
    if (xmlhttp) {
        try { // Versuch den Webservice zu kontaktieren
            var now = new Date();
            xmlhttp.open("GET", "php/webservice.php?t="+now.getTime(), true);
            xmlhttp.onreadystatechange = antwortAuswerten1; // url + timestamp, um Caching zu vermeiden
            xmlhttp.send(null);
        }
        catch (e) { // ggf Fehler anzeigen
            ausgabe.innerHTML = "<p>Keine Verbindung zum Server:<br />"
                + e.toString() + "</p>";
        }
    }
}

function antwortAuswerten1() { // immer, wenn sich der Status ändert
    ausgabe = document.getElementById('sop_antwort1');
}

```

```

if (xmlHttp.readyState == 4) { // ist der Prozess abgeschlossen?
  if (xmlHttp.status == 200) { // Ist die Antwort "OK"
    try { // Versuch die Antwort abzufragen
      ausgabe.innerHTML = xmlHttp.responseText;
    }
    catch(e) { // ggf. Fehler anzeigen
      ausgabe.innerHTML = "<p>Beim Empfang der Serverantwort ist "
        + "folgender Fehler aufgetreten: "
        + e.toString()
        + "</p>";
    }
  } else { // ggf. Statusmeldung anzeigen
    ausgabe.innerHTML = "<p>Der Abruf führte zu folgendem Fehler: "
      + xmlHttp.status + " "
      + xmlHttp.statusText + "</p>";
  }
}
}

// JS-Funktionen für einen AJAX-Aufruf zu einem 'fremden' Server
// (hier: http://webservice.helmutkarger.de)
function ajax2() {
  ausgabe = document.getElementById('sop_antwort2');
  try { // ein XMLHttpRequest Object anlegen
    xmlHttp = new XMLHttpRequest();
  }
  catch(e) { // Fehler: veralteter Browser IE6 o. älter
    ausgabe.innerHTML = "<p>Sorry, Ihr Browser versteht keinen XMLHttpRequest.</p>"
  }
  if (xmlHttp) {
    try { // Versuch den Webservice zu kontaktieren
      var now = new Date();
      xmlHttp.open("GET", "http://webservice.helmutkarger.de/php/webservice.php?t="+now.getTime(),
true); // url + timestamp, um Caching zu vermeiden
      xmlHttp.onreadystatechange = antwortAuswerten2;
      xmlHttp.send(null);
    }
    catch (e) { // ggf Fehler anzeigen
      ausgabe.innerHTML = "<p>Keine Verbindung zum Server:<br />"
        + e.toString() + "</p>";
    }
  }
}

function antwortAuswerten2() { // immer, wenn sich der Status ändert
  ausgabe = document.getElementById('sop_antwort2');
  if (xmlHttp.readyState == 4) { // ist der Prozess abgeschlossen?
    if (xmlHttp.status == 200) { // Ist die Antwort "OK"
      try { // Versuch die Antwort abzufragen
        ausgabe.innerHTML = xmlHttp.responseText;
      }
      catch(e) { // ggf. Fehler anzeigen
        ausgabe.innerHTML = "<p>Beim Empfang der Serverantwort ist "
          + "folgender Fehler aufgetreten: "
          + e.toString()
          + "</p>";
      }
    } else { // ggf. Statusmeldung anzeigen
      ausgabe.innerHTML = "<p>Der Abruf führte zu folgendem Fehler: "
        + xmlHttp.status + " "
        + xmlHttp.statusText + "</p>";
    }
  }
}
}

```

5.4.3 jsonp.js (JSONP-Kommunikation)

```

// JS-Funktionen für einen JSONP-Aufruf zu einem 'fremden' Server
// (hier: http://webservice.helmutkarger.de)

function jsonp(){
  // Löschen bereits vorhandener JSONP Skripte
  var scripts = document.getElementsByTagName("script");
  for (i=0; i<scripts.length; i++) {
    var url = scripts[i].getAttribute("src");
    if(url.indexOf("callback")>=0) {

```

```
    scripts[i].parentNode.removeChild(scripts[i]);
  }
}

// Anlegen und Einfügen des neuen Skripts
var now = new Date();
url = "http://webservice.helmutkarger.de/php/jsonp.php?time="
    +now.getTime()+"&callback=callback";
var script = document.createElement("script");
script.setAttribute("src", url);
script.setAttribute("type", "text/javascript");
document.getElementsByTagName("head")[0].appendChild(script);
}

// Entgegennahme der Serverantwort
function callback(data) {
  document.getElementById("jsonp_antwort").innerHTML = data;
}
```

5.5 PHP

5.5.1 webservice.php (AJAX-Kommunikation)

```
<?php
$ip = $_SERVER['REMOTE_ADDR'];
echo "Ihre IP-Adresse lautet:<br/>"
    .$ip."<br />"
    ."Abgefragt am:<br/> "
    .date("d.m.Y")
    ." um "
    .date("H:i:s")
    ." Uhr (Serverzeit).";
?>
```

5.5.2 jsonp.php (JSONP-Kommunikation)

```
<?php
# auf Callback-Parameter prüfen
if (isset($_GET["callback"]) && !empty($_GET["callback"])) {
  $callback = $_GET["callback"];
  # IP-Adresse ermitteln
  $ip = $_SERVER['REMOTE_ADDR'];
  # Rückmeldung zusammenbauen
  $nachricht = "Ihre IP-Adresse lautet:<br />"
    .$ip."<br />"
    ."Abgefragt am:<br /> "
    .date("d.m.Y") . " um " . date("H:i:s")
    ." Uhr (Serverzeit).";
  # Header für ein JavaScript
  header("Content-Type: application/javascript");
  # Rückmeldung senden
  echo $callback."('".$nachricht."')";
} else {
  # Fehlermeldung bei fehlendem Callback
  echo "Dieser Aufruf ist nur als JSONP-Call gültig";
}
?>
```

5.6 Beispielanwendung

5.6.1 Screenshot

Weitere Seiten >>


```

      script.setAttribute("type", "text/javascript");
      document.getElementsByTagName("head")[0].appendChild(script);
    }
    // Entgegennahme der Serverantwort
    function callback(data) {
      document.getElementById("jsonp_antwort").innerHTML = data;
    }
    
```

JSONP Beispielapplikation

Der Standard **ISO 3166** definiert geografische Einheiten, wie zum Beispiel zweistellige Länderkürzel, die im Internet als Top-Level-Domains Verwendung finden. Dabei widmet sich die **ISO 3166-1** den Ländern und die **ISO 3166-2** den untergeordneten Regionen der jeweiligen Länder. Beispielfhaft würde 'DE' für das Land Deutschland stehen und 'BW' für das Bundesland Baden Württemberg. Zusammengesetzt als 'DE-BW' stellt das eine **geo.region** dar, wie sie in HTML-Seiten als **Geo-Meta-Tag** eingesetzt wird.

Country- und Regionalcode abrufen:



Wählen Sie aus der Liste ein Land und dann die zugehörige Region aus und Sie erhalten den Country- und Regionalcode:

Länder:

Regionen:

Countrycode =

5.6.2 JavaScript mit Google-Maps

```
// JavaScript-Funktionen für die Beispielapplikation appl.html
// (c) 2012 Helmut Karger
```

```
// Allgemeine JSONP-Funktion
function jsonp(para) {
  // Löschen bereits vorhandener JSONP Skripte
  var scripts = document.getElementsByTagName("script");
  for (var i=0; i<scripts.length; i++) {
    var url = scripts[i].getAttribute("src");
    if (url.indexOf("callback")>=0) {
      scripts[i].parentNode.removeChild(scripts[i]);
    }
  }
  // Anlegen und Einfügen des neuen Skripts
  var now = new Date();
  // url = "http://localhost/kursordner/countrycode.php?time="
  url = "http://webservice.helmutkarger.de/php/countrycode.php?time="
  +now.getTime();
  for (var key in para) {
    url += "&";
    url += key;
    url += "=";
    url += para[key];
  }
  var script = document.createElement("script");
  script.setAttribute("src", url);
  script.setAttribute("type", "text/javascript");
  document.getElementsByTagName("head")[0].appendChild(script);
}

// Array Suchfunktion
function array_search(key, needle, haystack) { // Key-Bezeichnung, Key-Wert, Arrayname
  for (var i in haystack) {
    if (haystack[i][key] == needle) {
      return haystack[i];
    }
  }
}

```



```

return false;
}

// Abruf der Länderliste
function countries() {
    document.getElementById("laenderliste").disabled = true; // Selectboxen deaktivieren
    document.getElementById("regionenliste").disabled = true;
    var para = { // Anforderungsparameter
        "command" : "countrylist",
        "callback" : "cb_country"
    }
    jsonp(para);
}

// Entgegennahme der Serverantwort
function cb_country(data) {
    appl.response = eval("(function(){return " + data + "});")(); // JSON in Objekt wandeln
    if(typeof(appl.response.status)=="undefined" || appl.response.status != "ok") { // Status prüfen
        if(typeof(appl.response.error)=="undefined") { // Fehlerbehandlung
            document.getElementById("error").innerHTML = "Datenabruf nicht möglich";
        } else {
            document.getElementById("error").innerHTML = appl.response.error;
        }
    }
    return; // und Abbruch
}
var s = document.getElementById("laenderliste"); // Select 'laenderliste'
appl.countries = appl.response.data; // Länderliste speichern
var d = appl.response.data;
s.options[0] = new Option("Bitte auswählen", 0); // erster Listeneintrag
for(var record in d) {
    s.options[s.options.length] = new Option(d[record]["country"], d[record]["cc"]); // mit
Options versorgen
}
s.disabled = false; // und aktivieren

// Abruf der Regionenliste, Land wurde ausgewählt
function regions() {
    var s = document.getElementById("regionenliste"); // Select 'regionenliste'
    s.options.length = 0; // vorhandene Options löschen
    s.options[0] = new Option("- - - - -"); // und Select deaktivieren
    s.disabled = true; // Regionenliste löschen
    appl.regions = new Array(); // Code löschen
    appl.cccr = ""; // Regionnamen löschen
    appl.region = "";
    appl.swlat = null;
    appl.swlon = null;
    appl.nelat = null;
    appl.nelon = null;
    document.getElementById("error").innerHTML = ""; // Anzeigen löschen
    document.getElementById("notice").innerHTML = "";
    document.getElementById("result").innerHTML = "";
    appl.cc = document.getElementById("laenderliste").value; // Selektiertes Land
    if(appl.cc == "0") { // Ausstieg, wenn "Bitte Auswählen" selektiert
        appl.country = ""; // Ländernamen löschen
        return;
    }
    appl.country = array_search("cc", appl.cc, appl.countries).country; // Ländernamen merken
    var para = { // Anforderungsparameter
        "command" : "regionlist",
        "cc" : appl.cc,
        "callback" : "cb_region"
    }
    jsonp(para);
}

// Entgegennahme der Serverantwort Regionenabfrage
function cb_region(data) {
    appl.response = eval("(function(){return " + data + "});")(); // JSON in Objekt wandeln
    if(typeof(appl.response.status)=="undefined" || appl.response.status != "ok") { // Status prüfen
        if(typeof(appl.response.error)=="undefined") { // Fehlerbehandlung
            document.getElementById("error").innerHTML = "Datenabruf nicht möglich";
        } else { // keine Regionen erhalten
            document.getElementById("notice").innerHTML = "Dieses Land hat keine Regionen";
        }
    }
    appl.rc = "";
    appl.region = "";
}

```

```

    appl.ccrs = appl.cc;
    document.getElementById("result").innerHTML = 'Countrycode = <span id="ccfeld">'
    +appl.ccrs+'</span>';
    mapUpdate();
}
return; // und Abbruch
}
var s = document.getElementById("regionenliste"); // Select 'regionenliste'
appl.regions = appl.response.data; // Regionenliste speichern
var d = appl.response.data;
s.options[0] = new Option("Bitte auswählen", 0); // erster Listeneintrag
for(var record in d) {
    s.options[s.options.length] = new Option(d[record]["region"], d[record]["rc"]); // mit
Options versorgen
}
s.disabled = false; // und enablen
}

// Region wurde ausgewählt
function regselected() {
    appl.rc = document.getElementById("regionenliste").value; // Selektierte Region
    if(appl.rc === "0") { // Ausstieg, wenn "Bitte Auswählen" selektiert
        document.getElementById("error").innerHTML = ""; // Anzeigen löschen
        document.getElementById("notice").innerHTML = "";
        document.getElementById("result").innerHTML = "";
        appl.ccrs = ""; // Code löschen
        appl.region = ""; // Regionnamen löschen
        appl.swlat = null;
        appl.swlon = null;
        appl.nelat = null;
        appl.nelon = null;
        return;
    }
    var regionsatz = array_search("rc", appl.rc, appl.regions); // zugehörigen Datensatz suchen
    appl.ccrs = regionsatz.ccrs; // Code merken
    appl.region = regionsatz.region; // Regionnamen merken
    appl.swlat = regionsatz.swlat; // Koordinaten, wenn vorhanden
    appl.swlon = regionsatz.swlon;
    appl.nelat = regionsatz.nelat;
    appl.nelon = regionsatz.nelon;
    document.getElementById("error").innerHTML = ""; // Anzeigen löschen
    document.getElementById("notice").innerHTML = "";
    document.getElementById("result").innerHTML = 'Countrycode = <span id="ccfeld">'
    +appl.ccrs+'</span>';
    mapUpdate();
}

// Cacheupdate
function cache() {
    var para = { // Anforderungsparameter
        "command" : "cache",
        "ccrc" : appl.ccrs,
        "swlat" : appl.swlat,
        "swlon" : appl.swlon,
        "nelat" : appl.nelat,
        "nelon" : appl.nelon,
        "callback" : "cb_cache"
    }
    jsonp(para);
}

// Entgegenname der Serverantwort bei Cacheupdate
function cb_cache(data) {
    appl.response = eval("(function(){return " + data + "});")(); // JSON in Objekt wandeln
}

// Google Maps Funktionen
function mapInit() { // Initialisierung Google Maps
    geocoder = new google.maps.Geocoder(); // Geocoder
    var latlng = new google.maps.LatLng(48, 11);
    // initiale Koordinaten
    var myOptions = { // Kartenoptionen
        zoom: 1,
        center: latlng,
        mapTypeId: google.maps.MapTypeId.ROADMAP
    };
    map = new google.maps.Map(document.getElementById("map"), myOptions); // Karte darstellen
}

```

```

}

function mapUpdate() { // Änderung in der Karte
    if(!appl.nelat || !appl.nelon || !appl.swlat || !appl.swlon) { // Koordinaten noch nicht
        vorhanden
        var address = "";
        if(appl.region) { // falls gegeben
            address += appl.region+", "; // die Region
        }
        address += appl.country; // und das Land
        geocoder.geocode( { 'address': address}, function(results, status) { // Koordinaten abfragen
            appl.geocoding = results;
            if (status == google.maps.GeocoderStatus.OK) { // erfolgreich
                map.fitBounds(results[0].geometry.viewport); // in Karte anzeigen
                appl.nelat = results[0].geometry.viewport.getNorthEast().lat(); // Koordinaten merken
                appl.nelon = results[0].geometry.viewport.getNorthEast().lng();
                appl.swlat = results[0].geometry.viewport.getSouthWest().lat();
                appl.swlon = results[0].geometry.viewport.getSouthWest().lng();
                if(appl.rc) cache(); // und in Cache speichern
            } else { // keine Koordinaten erhalten
                map.setZoom(1); // Weltkarte anzeigen
                document.getElementById("error").innerHTML = "Leider keine Anzeigegergebnisse";
            }
        });
    } else { // vorhandene Koordinaten
        map.fitBounds(new google.maps.LatLngBounds(
            new google.maps.LatLng(appl.swlat, appl.swlon),
            new google.maps.LatLng(appl.nelat, appl.nelon)
        )); // in Karte anzeigen
    }
}

var appl = new Object;
var map;
countries();
mapInit();

```

5.6.3 PHP mit Datenbankabfrage

```

<?php
# PHP Programm für appl.html
# JSONP-Webservice zur Datenbankabfrage
# JSONP.eu Projekt 2012 Helmut Karger

# Funktion zum Öffnen der Datenbank
function opendb() {
    $server = "localhost";
    $user = "*****"; # Info entfernt
    $pass = "*****"; # Info entfernt
    $verbindung = mysql_connect($server, $user, $pass); # zum DB-Server verbinden
    if(!$verbindung) return false;
    mysql_set_charset('utf8', $verbindung);
    # Verbindungszeichensatz
    if(!mysql_select_db("*****", $verbindung)) return false; # Datenbank auswählen Info entfernt
    return $verbindung;
}

# Funktion zum Abruf der Länderliste aus der Datenbank
function countrylist($db) {
    $sql = "SELECT * FROM country ORDER BY country"; # DB Select (alle Länder sortiert)
    $ergebnis = mysql_query($sql, $db); # DB Abfrage ausführen
    if(!$ergebnis) return false;
    $countries = array(); # Länderarray
    while ($record = mysql_fetch_array($ergebnis)) { # alle Datensätze lesen und
        $countries[] = array("cc" => $record["cc"], # in Array speichern
            "country" => $record["country"]);
    }
    return $countries;
}

# Funktion zum Abruf einer Regionenliste aus der Datenbank
function regionlist($db, $cc) {
    $sql = "SELECT r.ccrcc, r.rc, r.region, c.swlat, c.swlon, c.nelat, c.nelon FROM region r ";
    $sql .= "LEFT JOIN cache c ON r.ccrcc = c.ccrcc ";
    $sql .= "WHERE cc = '". $cc. "' ORDER BY region"; # DB Select (alle Regionen eines Landes + evtl.
    Cacheinhalt)
    $ergebnis = mysql_query($sql, $db); # DB Abfrage ausführen
}

```

```

if(!$ergebnis) return false;
$regions = array(); # Regionenarray
while ($record = mysql_fetch_array($ergebnis)) { # alle Datensätze lesen und
$record["region"] = str_replace("", "\u0027", $record["region"]); # Hochkomma escapen
$regions[] = array("ccrc" => $record["ccrc"],
"rc" => $record["rc"],
"region" => $record["region"],
"swlat" => $record["swlat"],
"swlon" => $record["swlon"],
"nelat" => $record["nelat"],
"nelon" => $record["nelon"]); # in Array speichern
}
return $regions;
}

# Funktion zum Setzen eines Cacheintrags in der Datenbank
function cache($db, $ccrc, $swlat, $swlon, $nelat, $nelon) {
$sql = "REPLACE INTO cache (ccrc, swlat, swlon, nelat, nelon) ";
$sql .= "VALUES ('$ccrc', '$swlat', '$swlon', '$nelat', '$nelon')"; # DB Insert
$ergebnis = mysql_query($sql, $db); # DB Query ausführen
if(!$ergebnis) return false;
return true;
}

# auf Callback-Parameter prüfen
if (isset($_GET["callback"]) && !empty($_GET["callback"])) {
$callback = $_GET["callback"];

# Ergebnisarray vorbereiten
$result = array(
"status" => "", # Status ok oder fail
"data" => "", # eigentliche Ergebnisdaten
"error" => "" # Fehlertext
);

# auf Kommando-Parameter prüfen
if (isset($_GET["command"]) && !empty($_GET["command"])) {
$command = $_GET["command"];
$db = opendir(); # Datenbank öffnen
if($db !== false) { # DB erfolgreich geöffnet
switch ($command) { # Unterscheidung der Kommandos
case "countrylist": # Anforderung Länderliste
$countries = countrylist($db); # Länderliste abfragen
if($countries) { # Länderliste erfolgreich erstellt
$result["data"] = $countries; # Ergebnis eintragen
$result["status"] = "ok"; # Erfolg
} else { # keine Daten aus DB erhalten
$result["status"] = "fail"; # Fehler
$result["error"] = "Datenbankfehler: Länderliste kann nicht erzeugt werden.";
}
break;
case "regionlist": # Anforderung Regionenliste
if (isset($_GET["cc"]) && !empty($_GET["cc"])) { # Parameter Land vorhanden?
$cc = $_GET["cc"];
$regions = regionlist($db, $cc); # Regionenliste abfragen
if($regions) { # Regionenliste erfolgreich erstellt
$result["data"] = $regions; # Ergebnis eintragen
$result["status"] = "ok"; # Erfolg
} else { # keine Daten aus DB erhalten
$result["status"] = "fail"; # Fehler
$result["error"] = "Datenbankfehler: Regionenliste kann nicht erzeugt werden.";
}
} else { # Fehler
$result["status"] = "fail"; # Fehler
$result["error"] = "Kein Ländername angegeben.";
}
break;
case "cache": # Befüllung Cachebereich
if (isset($_GET["ccrc"]) && !empty($_GET["ccrc"])
&& isset($_GET["swlat"]) && !empty($_GET["swlat"])
&& isset($_GET["swlon"]) && !empty($_GET["swlon"])
&& isset($_GET["nelat"]) && !empty($_GET["nelat"])
&& isset($_GET["nelon"]) && !empty($_GET["nelon"])) { # nötige Parameter vorhanden?
$cc_rc = $_GET["ccrc"]; # Land und Region
$swlat = $_GET["swlat"]; # Südwest Latitude
$swlon = $_GET["swlon"]; # Südwest Longitude
$nelat = $_GET["nelat"]; # Nordost Latitude
$nelon = $_GET["nelon"]; # Nordost Longitude
}
}
}
}
}

```

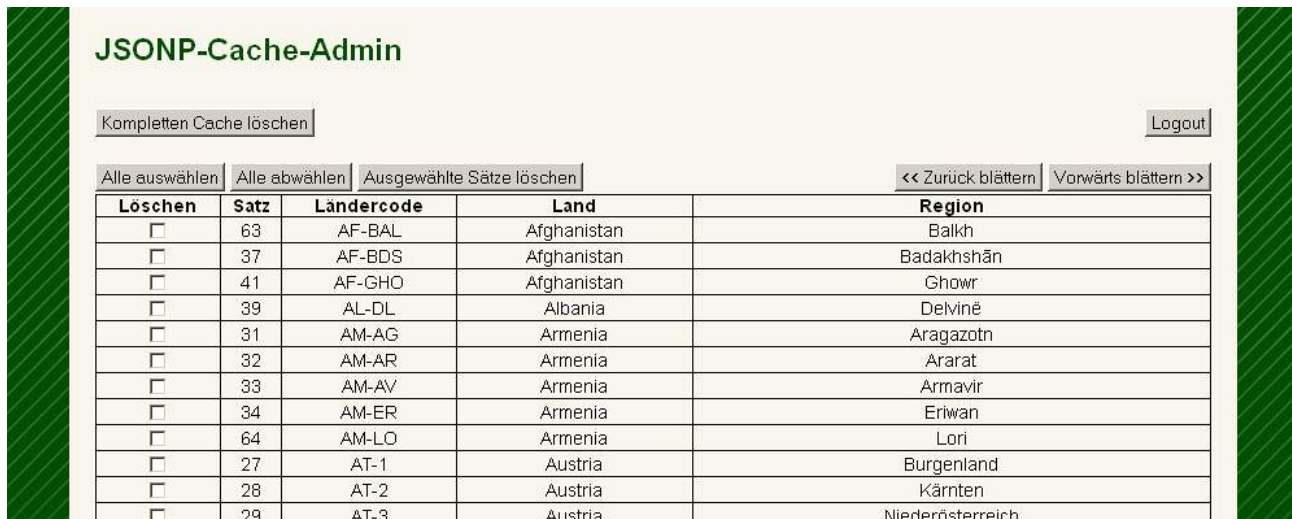
```

    $success = cache($db, $cc_rc, $swlat, $swlon, $nelat, $nelon);    # in Cache eintragen
    if($success) {                                               # Regionenliste erfolgreich erstellt
        $result["status"] = "ok";                                # Erfolg
    } else {                                                      # keine Daten aus DB erhalten
        $result["status"] = "fail";                             # Fehler
        $result["error"] = "Datenbankfehler: Cache Update nicht möglich.";
    }
    } else {
        $result["status"] = "fail";                             # Fehler
        $result["error"] = "Es fehlen die nötigen Parameter.";
    }
    break;
}
} else {
    $result["status"] = "fail";                                # Fehler
    $result["error"] = "Ungültiger Aufruf";                    # kein gültiges Kommando
}
mysql_close($db);                                             # Datenbank schließen möglich
} else {                                                       # kein Datenbankzugriff
    $result["status"] = "fail";                                # Fehler
    $result["error"] = "Datenbankzugriff nicht möglich";      # kein Datenbankzugriff
}
# Header für ein JavaScript
header("Content-Type: application/javascript");
# Rückmeldung senden
echo $callback."('".json_encode($result)."'");
# Rückgabe JSON codiert
} else {
    echo "Dieser Aufruf ist nur als JSONP-Call gültig"; # Fehlermeldung bei fehlendem Callback
}
?>

```

5.6.4 Cache-Admin-Programm

5.6.5 Screenshot



5.6.6 PHP

```

<?php
ini_set("session.use_trans_sid", 1);
session_start();
?>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>JSONP-Admin</title>
<link rel="shortcut icon" href="http://jsonp.eu/img/favicon.ico" type="image/x-icon" />
<link rel="stylesheet" type="text/css" href="http://jsonp.eu/css/main.css" />
<link rel="stylesheet" type="text/css" href="admin.css" />
</head>

```

```

<body>
<div id="wrapper"> <!-- Zentrierter Bereich mit Inhalt -->
<div id="header"> <!-- Header mit Logo, einem Image und Navigationsmenüs -->

<h1>JSONP</h1> <!-- wird ausgeblendet und durch Logo ersetzt -->
<h2>JSONP-Cache-Admin</h2> <!-- Seitenüberschrift -->
</div>
<div id="content"> <!-- Contentbereich -->
<?php
function login() { # Anmelden
if(isset($_SESSION["login"])) return TRUE; # bereits angemeldet
if(empty($_POST)) { # Passwortabfrage
echo "<form action='admin.php' method='POST'>\n";
echo "<p>Geben Sie bitte Ihr Passwort ein: <br />\n";
echo "<input type='password' name='password' />\n";
echo "<input type='submit' />\n";
echo "</p>\n</form>\n";
} else {
if(isset($_POST["password"]) && !empty($_POST["password"])) {
$pw = md5($_POST["password"]);
if($pw === "f931b13aead002d7fcd02f84e0f794f") { # Passwortcheck
$_SESSION["login"] = TRUE;
$_SESSION["index"] = 0;
return TRUE;
} else { # falsches Passwort
echo "Sorry, falsches Passwort!";
}
}
}
return FALSE;
}

if(login()) { # bereits angemeldet?
$server = "localhost";
$user = "*****"; # Info entfernt
$pass = "*****"; # Info entfernt
$verbindung = mysql_connect ($server, $user, $pass)
or die ("Keine Verbindung zum Server");
mysql_select_db("*****") # Info entfernt
or die ("Kein Zugriff auf die Datenbank");
mysql_set_charset('utf8',$verbindung);

if (isset($_POST["logout"])) { # Logout
$_SESSION = array(); # Session löschen
session_destroy();
echo "<p>Abmeldung erfolgreich - <a href='admin.php'> Neu anmelden</a>!\n";
exit;
} elseif (isset($_POST["deleteall"])) { # gesamten Cache löschen
$sql = "DELETE FROM cache"; # alle Sätze in DB löschen
$abfrage = mysql_query($sql);
} elseif (isset($_POST["delete"])) { # ausgewählte Sätze löschen
if (isset($_POST["auswahl"])) { # Auswahlarray
foreach ($_POST["auswahl"] as $id) { # durchlaufen
$sql = "DELETE FROM cache WHERE cid = ".$id; # und Sätze in DB löschen
$abfrage = mysql_query($sql);
}
}
} elseif (isset($_POST["next"])) { # nächste Sätze anzeigen
$sql = "SELECT * FROM cache"; # alle Datensätze in cache
$abfrage = mysql_query($sql);
$rows = mysql_num_rows($abfrage); # Anzahl der Datensätze
if ($rows > $_SESSION["index"] + 30) { # weitere vorhanden
$_SESSION["index"] +=30; # die nächsten 30 anzeigen
}
} elseif (isset($_POST["previous"])) { # vorhergehende Sätze anzeigen
$_SESSION["index"] -=30; # vorherige 30 anzeigen
if ($rows < 0) $_SESSION["index"] = 0; # oder am Anfang beginnen
}

echo "<form name='formular' action='admin.php' method='POST'>\n";
echo "<div class='links'><input type='submit' name='deleteall' value='Kompletten Cache

```

```

löschen' /></div>\n";
echo "<div class='rechts'><input type='submit' name='logout' value='Logout' /></div>\n";
echo "<div class='stopfloat'></div>\n";
$sql = "SELECT ch.cid, ch.ccrcc, c.country, r.region
FROM cache ch
LEFT JOIN region r ON ch.ccrcc = r.ccrcc
LEFT JOIN country c ON r.cc = c.cc
LIMIT ".$_SESSION["index"].", 30;";
$abfrage = mysql_query($sql); # SQL-Abfrage
# absenden
if(mysql_num_rows($abfrage) == 0) { # keine Einträge
echo "<br /><p>Der Cachebereich ist leer!</p>\n";
} else { # in Tabelle ausgeben
echo "<br />\n";
echo "<div class='links'><input type='button' name='all' value='Alle auswählen'
onClick='for(i=0; i<document.formular.elements.length; i++)
{document.formular.elements[i].checked=true;}' />\n";
echo "<input type='button' name='none' value='Alle abwählen'
onClick='for(i=0; i<document.formular.elements.length; i++)
{document.formular.elements[i].checked=false;}' />\n";
echo "<input type='submit' name='delete' value='Ausgewählte Sätze löschen' /></div>\n";
echo "<div class='rechts'><input type='submit' name='previous' value='<< Zurück blättern' />\n";
echo "<input type='submit' name='next' value='Vorwärts blättern >>' /></div>\n";
echo "<div class='stopfloat'></div>\n";
echo "<table border='1'>\n";
echo "<tr><th>Löschen</th><th>Satz</th><th>Ländercode</th><th>Land</th><th>Region</th></tr>\n";
while ($eintrag = mysql_fetch_array($abfrage)) { # für jeden DB-Eintrag
echo "<tr>\n";
echo "<td>\n";
echo "<input type='checkbox' name='auswahl[]' value='".$eintrag["cid"]." ' />";
echo "</td>\n";
echo "<td>\n";
echo $eintrag["cid"];
echo "</td>\n";
echo "<td>\n";
echo $eintrag["ccrcc"];
echo "</td>\n";
echo "<td>\n";
echo $eintrag["country"];
echo "</td>\n";
echo "<td>\n";
echo $eintrag["region"];
echo "</td>\n";
echo "</tr>\n";
}
echo "</table>\n";
echo "<div class='links'><input type='button' name='all' value='Alle auswählen'
onClick='for(i=0; i<document.formular.elements.length; i++)
{document.formular.elements[i].checked=true;}' />\n";
echo "<input type='button' name='none' value='Alle abwählen'
onClick='for(i=0; i<document.formular.elements.length; i++)
{document.formular.elements[i].checked=false;}' />\n";
echo "<input type='submit' name='delete' value='Ausgewählte Sätze löschen' /></div>\n";
echo "<div class='rechts'><input type='submit' name='previous' value='<< Zurück blättern' />\n";
echo "<input type='submit' name='next' value='Vorwärts blättern >>' /></div>\n";
echo "<div class='stopfloat'></div>\n";
echo "</form>\n";
}
}
?>
<br />
</div>
<div id="footer">
<p id="weiterlink">Weiter zur Anwendungsseite <a href="http://jsonp.eu">JSONP.eu</a>.</p>
<p id="copyright">(c) 2012 by Helmut Karger</p>
</div>
</div>
</body>
</html>

```