

Intrexx Professional Intrexx Compact

RELEASE 5.2



Whitepaper

Intrexx & Mobile Endgeräte

Inhaltsverzeichnis


1. Einführung	4
2. Kategorien mobiler Endgeräte	4
2.1. iPhone.....	7
2.2. BlackBerry.....	8
2.3. Geräte „älterer“ Generation.....	9
3. Entwicklung für mobile Endgeräte	10
3.1. Allgemein.....	10
3.2. Display.....	12
3.3. Bedienung.....	12
3.4. Navigation.....	13
3.5. Layout.....	14
3.6. JavaScript.....	15
3.7. Zu vermeidende Elemente.....	16
4. HTML-Strukturen	17
5. Layouts und Applikationen mit Intrexx	18
5.1. Layouts.....	18
5.2. Applikationen.....	20
5.2.1. Seitenassistent.....	21
5.2.2. Seitenassistent Fazit.....	27
5.2.3. Kontrollen.....	28
5.2.4. Abhängigkeiten.....	30
5.2.5. HTML-Elemente und Styles.....	30
5.2.6. Menüstruktur bearbeiten.....	33
5.3. Benutzerdefiniertes JavaScript.....	34
5.3.1. Globale Definition.....	35
5.3.2. Definition innerhalb einer Applikation.....	36
6. Links (Stand 26.10.2009)	37
6.1. Einführung in mobiles Design.....	37
6.2. Best Practices.....	37
6.3. Tests.....	37
6.4. Emulatoren.....	38
6.5. Apple.....	38
6.6. BlackBerry.....	38
7. Über Intrexx	39
8. Über United Planet	41



Copyright






Das vorliegende Dokument ist in all seinen Teilen urheberrechtlich geschützt. Alle Rechte sind vorbehalten, insbesondere das Recht der Übersetzung, des Vortrags, der Reproduktion und der Vervielfältigung. Ungeachtet der Sorgfalt, die auf die Erstellung von Text, Abbildungen und Programmen verwendet wurde, können weder Autor, Herausgeber oder Übersetzer für mögliche Fehler und deren Folgen eine juristische Verantwortung oder irgendeine Haftung übernehmen.

Die in diesem Werk wiedergegebenen Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. können auch ohne besondere Kennzeichnung Marken sein und als solche den gesetzlichen Bestimmungen unterliegen.



Schreibkonventionen

In diesem Handbuch werden Textstellen *kursiv* dargestellt, wenn sie sich auf Einstellungen in den abgebildeten Dialogen beziehen. Menüpunkte, die in Kontextmenüs erreichbar sind, sind immer auch über das Hauptmenü erreichbar. Hauptmenüpunkte werden nicht beschrieben, es sei denn, sie sind nicht über das Kontextmenü erreichbar. Eine Beschreibung der allgemeinen Hauptmenüpunkte finden Sie im Handbuch  *Portale*. Programmiercode im Text wird in der Schriftart *Courier* dargestellt. Kontextmenüs können mit einem Klick mit der rechten Maustaste auf das beschriebene Element geöffnet werden.

<intrexx> bezeichnet im Folgenden Ihren Intrexx Installationspfad, unter Windows z.B.  c:\programme\intrexx\, unter Linux z.B.  /opt/intrexx/. Folgende Symbole werden für die Kennzeichnung von speziellen Informationen verwendet:

-  Informationen
-  Verweis auf ein Intrexx Handbuch
-  Verzeichnisse
-  URLs
-  Klick auf Schaltflächen

Vorkenntnisse

Für das Verständnis dieser Dokumentation sind keine speziellen Vorkenntnisse erforderlich. Hilfreiche Informationen finden Sie in den Intrexx Handbüchern  *Setup* und  *Start*.

1. Einführung

Mobile, webfähige Endgeräte wie Netbooks und Smartphones ermöglichen mittlerweile den Zugriff auf das Internet von nahezu jedem beliebigen Ort und zu jedem Zeitpunkt. Auch Unternehmen können auf vielfältige Art und Weise von diesen neuen Technologien profitieren - sei es zur Anbindung der Außendienstmitarbeiter oder um Vorgesetzten und Management den Zugriff auf wichtige Informationen auch außerhalb des Büros zu ermöglichen. Damit wird ein reibungsloser Ablauf der betrieblichen Prozesse sichergestellt – unabhängig davon, ob die Entscheidungsträger persönlich im Unternehmen anwesend sind.

Diesen Anforderungen trägt der Softwarehersteller United Planet mit Version 5 seiner Portalsoftware Intrexx Rechnung. Intrexx 5 bietet die Möglichkeit, einzelne Anwendungen und sogar komplette Enterprise Portale speziell für mobile Endgeräte zu entwickeln. Um die technische Umsetzung und die speziellen Anforderungen der verschiedenen Geräte-Typen (wie Bandbreite oder Sliding-Effekte) kümmert sich Intrexx von alleine. Hierbei spielt es keine Rolle, ob es sich um ein iPhone, einen BlackBerry, einen Tablet PC oder auch ein internetfähiges Handy älterer Generation handelt.

Besonderer Wert wurde hierbei auf diejenigen Anforderungen gelegt, die der Einsatz von mobilen Endgeräten mit sich bringt: Hierzu gehören u.a. das Ladezeitverhalten, um möglichst wenig Datenvolumen zu erzeugen, sowie erweiterte Design- & Layoutmöglichkeiten, um Anwendungen und Portale benutzerfreundlich und einfach handhabbar an mobile Endgeräte und deren Spezifikationen wie Bedienmöglichkeiten und Displaygröße anpassen zu können.

Im hier vorliegenden Dokument soll Anwendern und Designern eine Einführung in das Thema *Intrexx und mobile Endgeräte* gegeben werden, um einen Einblick zu erhalten, welche Besonderheiten beim Design von Anwendungen für mobile Endgeräte zu beachten sind, welche Ideen und Überlegungen dahinterstecken und was für Möglichkeiten vorhanden sind.

2. Kategorien mobiler Endgeräte

Auf dem Markt befindet sich eine nahezu unendliche Zahl an mobilen, internetfähigen Geräten. Sie unterscheiden sich alle jeweils durch die Art der Bedienung, den Bedienkomfort, die technischen Fähigkeiten wie CSS-Darstellungsmöglichkeiten sowie Displaygrößen.

Im Folgenden werden verschiedene Kategorien von Endgeräten mit jeweils einem Beispielgerät aufgelistet, um einen Einblick zu geben, auf welche Gerätebesonderheiten zu achten ist. Auf Grund der Modellvielfalt kann hier allerdings lediglich ein Ausschnitt der vorhandenen Produktpalette betrachtet und analysiert werden.

Die folgenden Abbildungen zeigen die unterschiedliche Darstellung einer Internetseite auf verschiedenen Endgeräten. Anhand dieses Beispiels wird die Notwendigkeit einer Einordnung der diversen Geräte deutlich.

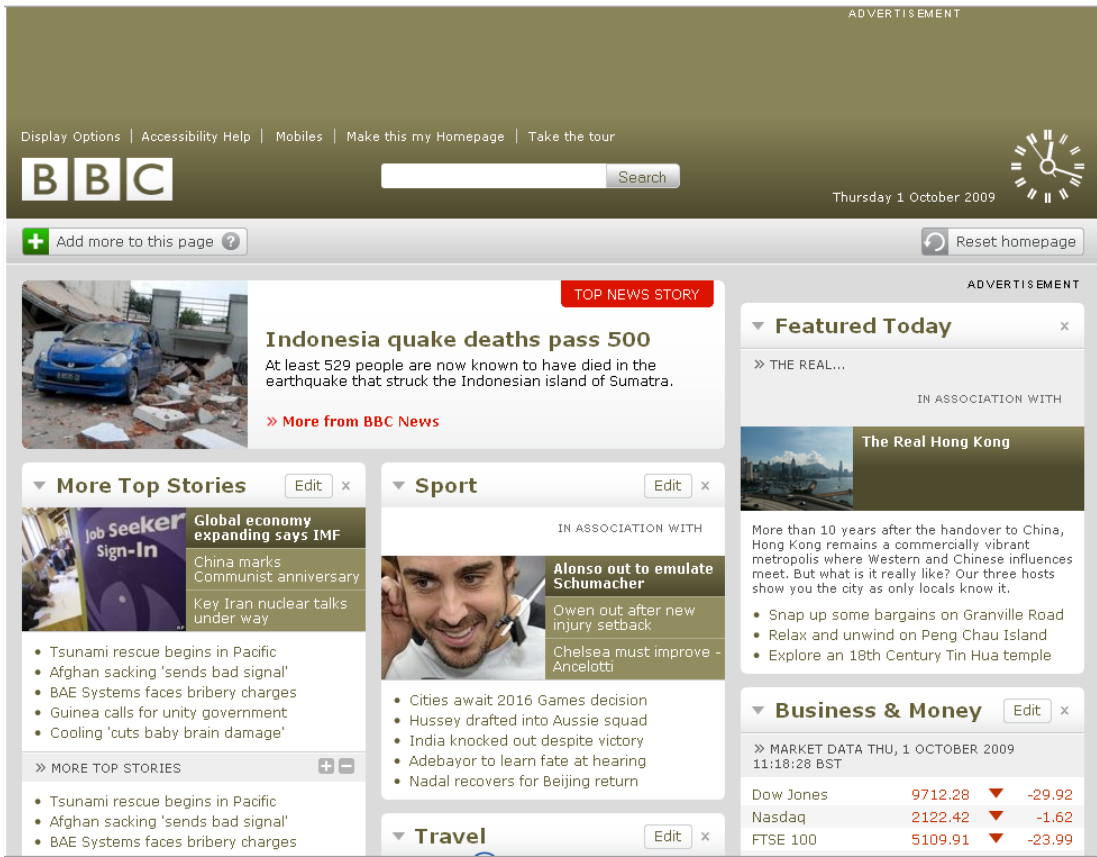


Abbildung 1: Aufruf per Desktop-PC

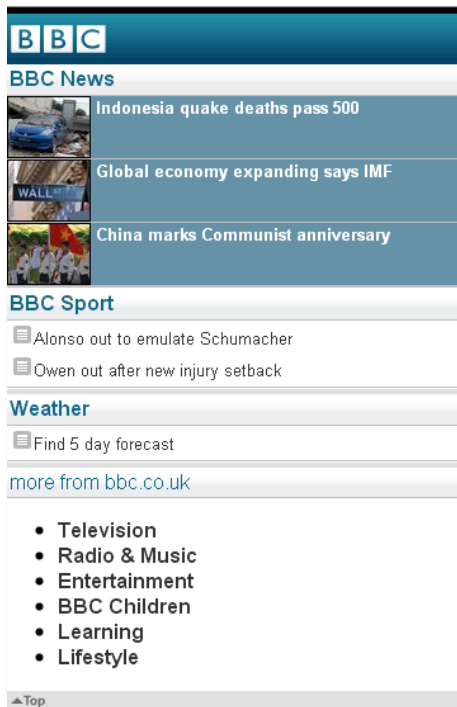


Abbildung 2: Aufruf per iPhone



Abbildung 3: Aufruf per BlackBerry 9000

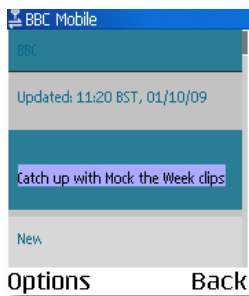


Abbildung 4: Aufruf per Nokia N70

Diese Beispiele zeigen, wie sehr sich der Umfang bereitstehender Informationen und deren Darstellung je nach Gerät unterscheiden. Hieran wird auch deutlich, welche Grundsätze bei der Entwicklung für mobile Endgeräte zu beachten sind.

Als Grundstrategie gelten hierbei die Ansätze des *default delivery context* und des *progressive enhancement*. Beim *default delivery context*¹ handelt es sich um Mindestanforderungen, die an die Anwendungen und das Layout gestellt werden. Hält man diese Anforderungen ein, ist sichergestellt, dass der gewünschte Inhalt auch auf älteren Geräten in einer lesbaren Form dargestellt werden kann. Die Anforderungen beinhalten u.a. eine Einschränkung verwendeter Bilder auf die Formate JPEG & GIF 89a und die Verwendung des Encodings UTF-8.

Aufbauend auf diesem Grundkontext lassen sich mit Hilfe des *progressive enhancement* detailliertere und umfassendere Anforderungen realisieren, die beispielsweise Features für spezifische Geräte bereitstellen und deren technische Fähigkeiten optimal nutzen.

Im Vergleich zu Desktopvarianten sollten Layouts für Smartphones möglichst einfach gestaltet werden. Dadurch lässt sich das auftretende Datenvolumen möglichst gering halten. Während dieser Aspekt im Desktop-Umfeld (vor allem bei Intranets) auf Grund großer verfügbarer Bandbreiten kaum Beachtung findet, ist er für die Nutzung in einem mobilen Netz essentiell. Hier stehen überwiegend geringe Bandbreiten zur Verfügung, weshalb es wichtig ist, das Datenaufkommen und die Anzahl der Requests zu minimieren.

Ein weiterer wichtiger Punkt, der beachtet werden muss, sind unterschiedliche Displaygrößen, also der Platz, der zur Bereitstellung von Informationen zur Verfügung

¹ <http://www.w3.org/TR/mobile-bp/#ddc>

steht. Designer von Layouts und Anwendungen müssen darauf achten, Informationen möglichst effektiv und übersichtlich bereitzustellen.

Von zentraler Bedeutung sind ebenfalls die technischen Fähigkeiten der einzelnen mobilen Geräte. Hierbei ist es wichtig, ob ein Gerät CSS-Fähigkeiten beherrscht und wenn ja, in welchem Umfang. Verfügt es über eine vollständige, nur über eine eingeschränkte, oder unter Umständen über gar keine Interpretationsimplementierung? Besitzt das Gerät die Fähigkeit, JavaScript auszuführen? Diese und weitere Fragen müssen bereits im Vorfeld geklärt werden, um effektiv, zeitnah und erfolgreich Anwendungen und Portale für den mobilen Zugriff umzusetzen.

2.1. iPhone

Das Apple iPhone verfügt über keine mechanischen Bedienelemente wie Tastatur oder Maus. Die Bedienung und Navigation erfolgt ausschließlich über einen Touchscreen in Verbindung mit den Fingern des jeweiligen Benutzers, was einen erheblichen Unterschied zur Bedienung mit Hilfe einer Maus darstellt. Dadurch ergeben sich spezielle Gegebenheiten beispielsweise hinsichtlich Navigationsstrukturen, Eingabefeldern sowie Schaltflächen und Links, da eine punktgenaue Navigation mit Hilfe einer Maus oder die Unterstützung von Tastaturkürzeln nicht gegeben ist.

Der große Touchscreen mit einer Auflösung von 480x320 Pixeln stellt Designern ausreichend Platz zur Verfügung, um alle gewünschten Informationen übersichtlich und mit den Fingern leicht erreichbar bereitzustellen. Die eingesetzte Struktur innerhalb von iPhone-Applikationen zur Darstellung enthält hierbei üblicherweise Listenelemente, woraus die iPhone-typische Navigation mittels einzelner Ebenen (Ebene höher, Ebene tiefer) zu Stande kommt.

Beim Design ist speziell darauf zu achten, dass die Fenster über eine fixe Größe verfügen und nicht - wie an Desktop-PCs - variabel in ihrer Größe sind (Fenster minimieren oder maximieren). Designs und Alignments sollten mit relativen Werten (Prozent) versehen werden, damit sich diese jeweils dem Format des iPhones (Hochformat oder Querformat) anpassen. Darüber hinaus besteht die Möglichkeit, die unterschiedliche Darstellung in Hoch- bzw. Querformat über CSS zu realisieren. Dafür stellt Ihnen Intrexx bereits eine Auswahl an vordefinierten CSS-Stilklassen zur Verfügung, die diese Art der Darstellung umsetzen.

Es existieren zudem keine Scrollbalken, wie man sie von einem Desktop-Browser kennt. Dies bedeutet, dass eine Navigation zu Inhalten, die sich nicht im Anzeigebereich des Displays befinden, über Zoomen und Verschieben der Inhalte mit Hilfe der Finger erfolgen muss. Man sollte daher immer versuchen, die darzustellenden Inhalte an die zur Verfügung stehende Displaybreite anzupassen, um die Benutzernavigation so gering wie möglich zu halten. Ein horizontales Scrollen ist unbedingt zu vermeiden.



Abbildung 5: iPhone Listenlayout

Technische Details iPhone

Feature	Beschreibung
Browser	iPhone Safari, entspricht nahezu vollständig der Desktopvariante von <i>Safari</i> .
CSS 2.1	Nahezu vollständige Kompatibilität
CSS 3	Nahezu vollständige Kompatibilität seit Safari Version 3.2
JavaScript	Vollständige Kompatibilität (Viele iPhone Funktionen wie das Sliding, also das Hineingleiten, von Seiten sind mit Hilfe von JavaScript realisiert)
HTML, XHTML und XHTML MP	Vollständige Kompatibilität

2.2. BlackBerry

BlackBerry Smartphones etwas älterer Generationen verfügen im Gegensatz zu Apples iPhone oder aktuellen BlackBerry-Geräten wie z.B. Storm oder Torch über eine Tastatur und einen Trackball, mit dessen Hilfe innerhalb des Menüs navigiert werden kann. Befindet man sich in Anwendungen wie dem BlackBerry Browser, verfügt man über einen Mauszeiger wie bei einem Desktop-PC, der mit Hilfe des Trackballs gesteuert wird.

Die unterstützten HTML-, CSS- und JavaScript-Möglichkeiten richten sich nach der auf dem jeweiligen Gerät installierten Betriebssystemversion (*BlackBerry Device Operating System*). Zwar wurden bereits mit Version 3.8 wenige grundlegende Funktionen wie eine Unterstützung für CSS und JavaScript bereitgestellt, eine richtige, technisch hochwertigere, Unterstützung erfolgte jedoch erst mit Systemversion 4.6.

Technische Details BlackBerry

Feature	Beschreibung
Browser	BlackBerry-eigener Browser
CSS 2.1	Kompatibilität mit einigen Grundfunktionen wurde mit Version 3.8 eingeführt. Vollständige Kompatibilität seit Betriebssystemversion 4.6
CSS 3	Seit Betriebssystemversion 4.6 Unterstützung für einige, wenige Funktionalitäten (z.B. <i>Color</i>)
JavaScript	Unterstützung seit Version 3.8
HTML, XHTML und XHTML MP	Seit 3.7 besteht eine Unterstützung von grundlegenden Elementen. Diese Unterstützung wurde mit den darauf folgenden Versionen 4.0, 4.2 und 4.6 immer weiter ausgeweitet. Mit aktuellsten Versionen wird ein Großteil von HTML-, XHTML- und XHTML MP Elementen unterstützt. Seit Version 4.6 werden ebenfalls AJAX und xmlhttp unterstützt.

2.3. Geräte „älterer“ Generation

Zusätzlich zu Geräten der neusten Generation wie iPhone oder BlackBerry existiert noch eine Vielzahl an Geräten älterer Bauart, die bei weitem noch nicht über die heute möglichen technischen Fähigkeiten verfügen. Handys wie beispielsweise ein Nokia N70 verfügen weder über einen Touchscreen noch über Trackballs. Die Navigation erfolgte lediglich über Pfeiltasten, mit denen von Link zu Link gesprungen wird oder mit Hilfe der Zifferntasten.



Abbildung 6: Seitenaufrufe über Nokia N70

Weitere fehlende technische Fähigkeiten manifestieren sich in geringeren Displaygrößen und Auflösungen, sowie einer komplett fehlenden oder nur rudimentär vorhandenen Möglichkeit der CSS-Interpretation.

Hinzu kommt, dass diese Geräte nicht über die Fähigkeit verfügen, JavaScript auszuführen und eine sehr geringe Prozessorgeschwindigkeit besitzen. Deshalb ist bei der Entwicklung einer Anwendung für diese und ähnliche Geräte darauf zu achten, Informationen möglichst direkt, einfach, intuitiv bedienbar und übersichtlich bereitzustellen.

3. Entwicklung für mobile Endgeräte

In diesem Abschnitt werden allgemeine Grundsätze erläutert, die beim Entwickeln von mobilen Anwendungen zu beachten sind. Anwendungsdesigner, die bisher noch nicht im Bereich der mobilen Entwicklung tätig waren, müssen bedenken, dass man im mobilen Alltag nahezu vollständig andere Gegebenheiten vorfindet als in einem Desktop-Umfeld.

3.1. Allgemein

Bereits im Vorfeld bzw. zu Beginn der Entwicklung einer Anwendung für mobile Endgeräte sollte man wissen, für welche Gerätearten ein zu implementierender Service zur Verfügung gestellt werden soll. Eine einzige Umgebung „für das mobile Web“ und alle Geräte zu entwickeln ist nicht empfehlenswert. Stattdessen sollten Anwendungen und Layouts darauf ausgelegt werden, mit welchem Gerät zu einem späteren Zeitpunkt darauf zugegriffen werden soll. Diese Unterscheidung lässt sich anhand des sogenannten *Benutzeragenten* definieren. Dabei handelt es sich um Geräte oder Software wie Browser, die bei einem Zugriff einen eindeutigen Namen angeben/versenden, um sich zu identifizieren.

Ein derartiger Name kann beispielsweise wie folgt aussehen:

Für Firefox 3.6

Mozilla/5.0 (Windows; U; Windows NT 6.1; de; rv:1.9.2.13) Gecko/20101203 Firefox/3.6.13

Für Safari 4.0 (iPhone)

Mozilla/5.0 (iPhone; U; CPU iPhone OS 4_0 like Mac OS X; en-us) AppleWebKit/532.9 (KHTML, like Gecko) Version/4.0.5 Mobile/8A293 Safari/6531.22.7

Diese Namen lassen sich nach bestimmten Schlüsselwörtern wie beispielsweise *Firefox*, *Linux* oder *iPhone* durchsuchen und anhand dessen entsprechende Layouts und Anwendungen wählen. Intrexx kann diese Suche für Anwender und Designer übernehmen. Es muss kein eigener Mechanismus zum Auslesen und Auswählen implementiert werden. Es müssen lediglich für Layouts, wie oben erwähnt, Schlüsselwörter hinterlegt werden.

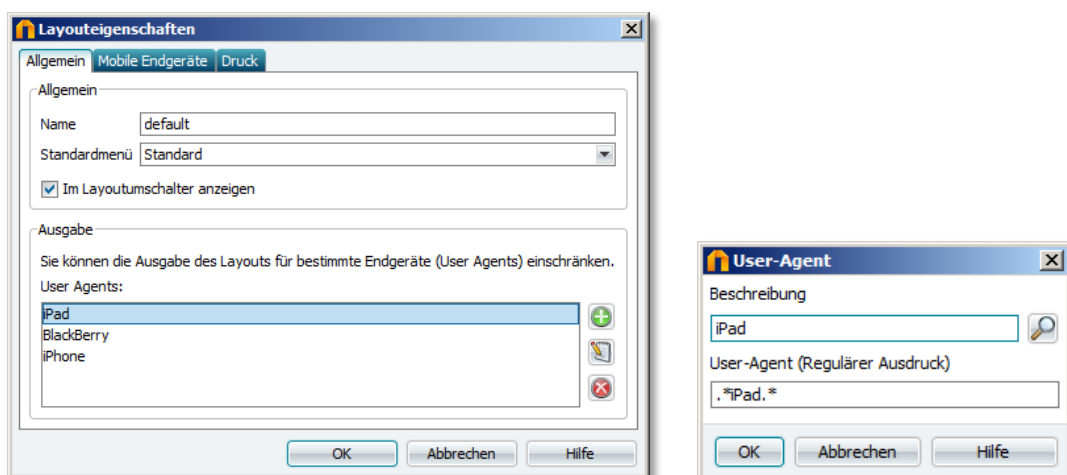
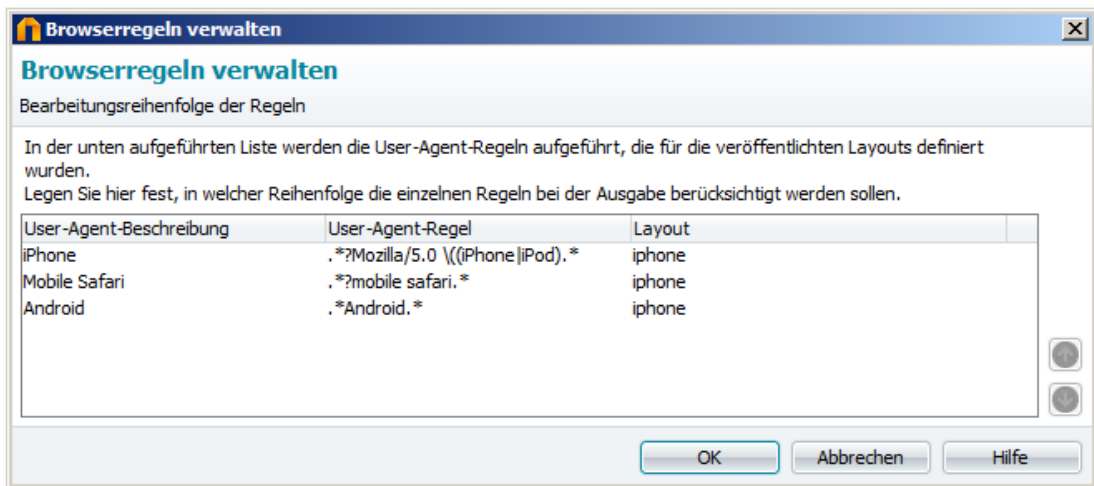


Abbildung 7: Benutzeragent zuweisen

- ⓘ Auf Grund der sehr großen Anzahl verwendeter Benutzeragenten-Strings und mehrfach vorkommenden Schlüsselwörter sind unter Umständen gewisse Reihenfolgen zu beachten, nach denen diese Namen untersucht werden. Damit wird

die korrekte Zuweisung von Layouts zu einem Agenten gewährleistet.

Ein Beispiel hierfür ist, dass das Schlüsselwort *Mobile* in einem Benutzeragenten-String für iPhone und Blackberry vorkommen kann. Sie können daher innerhalb des Moduls *Design* unter *Layoutverwaltung* angeben, in welcher Reihenfolge angelegte Regeln zu Benutzeragenten abgearbeitet werden. So kann eine korrekte Zuweisung eines gerätespezifischen Layouts sichergestellt werden.



Anwendungen und Layouts, die von Designern umgesetzt werden, sollten entsprechend dieses Agenten eine für das jeweilige Gerät angepasste Umgebung liefern, die die Eigenheiten dieses Geräts berücksichtigt und dessen Vorteile nutzt. Dazu zählen u.a. eigene Layouts und CSS-Stilklassen sowie die Berücksichtigung verschiedener Displaygrößen.

Im Zusammenhang diverser Bildschirmgrößen seien an dieser Stelle die Stichworte *elastisches Layout* und *liquides Layout* erwähnt. Unter einem *elastischen Layout* versteht man, dass sich eine Webseite nicht an einem Browserfenster ausrichtet, sondern nach der verwendeten Schriftgröße. Das Layout „wächst“ und „schrumpft“ mit der eingestellten Schriftgröße. Bei *liquiden* Layouts handelt es sich um Layouts, die sich jeweils an die Gegebenheiten (Bildschirmgrößen) des zugreifenden Gerätes anpassen. Dies bedeutet, dass man beispielsweise bei Größenangaben mit Prozentwerten statt mit absoluten Einheiten wie Pixel (px) oder Point (pt) arbeiten sollte.

Weiterhin ist das normalerweise unterschiedliche Verhalten zwischen Desktop-Benutzern und Benutzern mobiler Geräte zu beachten. Oftmals werden im Desktopbereich Anwendungen oder Webseiten geöffnet, um sie erst zu einem späteren Zeitpunkt zu verwenden oder es werden mehr Informationen dargestellt, als zum aktuellen Zeitpunkt gebraucht werden. Als Beispiel sei an dieser Stelle ein CRM-System genannt, das nach der Eingabe einer Kundennummer alle zu dieser Nummer gespeicherten Daten in einem Formular angezeigt. Hierbei kann es sich um Kontaktdaten, Ansprechpartner, Umsätze, Aufträge und weitere Daten handeln. Im Desktop-Umfeld erfolgt eine Entwicklung meistens dahingehend, dass die vorhandenen Daten übersichtlich dargestellt werden, um alle Informationen schnell „auf einen Blick“ bereit zu haben.

Das mobile Verhalten verändert sich normalerweise dahingehend, dass eine bestimmte Information nur zum Zeitpunkt des Aufrufs direkt benötigt wird und verfügbar sein muss.

Eine Speicherung weiterer Daten auf Vorrat oder eine Anzeige zu vieler Informationen ist im Normalfall nicht gewünscht.

3.2. Display

Mittlerweile verfügen Monitore von Desktop-PCs sowie Notebook-Displays über eine ausreichende Größe und Auflösung, um viele Informationen auf einmal übersichtlich darstellen zu können. Designer, allen voran Webdesigner, können zum heutigen Zeitpunkt mit einer Mindestauflösung von 800x600 Pixeln rechnen. Man verfügt über genügend Bildschirmplatz, um einen großen Informationsgehalt auf einmal darzustellen. Hierbei werden oft mehr Informationen angezeigt, als tatsächlich zum Zeitpunkt des Aufrufs benötigt werden.

Mobile Endgeräte verfügen jedoch über weitaus kleinere Displays (z.B.: iPhone Display: 3,5 Zoll bei einer Auflösung von 480x320 Pixeln). Man muss daher darauf achten, einen Kompromiss zu finden, um einerseits genügend Informationen bereitstellen zu können, andererseits jedoch die Lesbarkeit zu gewährleisten. Man sollte mit möglichst wenig Navigation und Benutzeraktivität wie Zoomen und Scrollen auf angezeigte Informationen zugreifen können. Das Zoomen auf Inhalte sollte vermieden werden, zumal zum aktuellen Zeitpunkt nur wenige Geräte über diese technische Fähigkeit verfügen. Beim Thema Scrollen sollte vor allem das horizontale Scrollen unbedingt vermieden werden.

Es ist ebenfalls zu berücksichtigen, dass mobile Endgeräte meistens unter anderen Lichtverhältnissen eingesetzt werden, als man diese in einem gut ausgeleuchteten Büro vorfindet. Benutzer arbeiten unterwegs beispielsweise unter direkter Sonneneinstrahlung oder in dämmerigen Lichtverhältnissen ohne zusätzliche Lichtquellen. Eine zu kleine oder nicht ausreichend mit Kontrast versehene Darstellung von Daten führt zu Leseschwierigkeiten und erschwert Endanwendern die Verwendung im mobilen Einsatz.

3.3. Bedienung

Auch bei der Art der Bedienung und der täglichen Anwendung unterscheiden sich die beiden Gerätearten Desktop-PC und mobiles Endgerät grundlegend, und das, obwohl beide im selben Umfeld (z.B. Internet & Intranet) eingesetzt werden. Bei Desktop-PCs verfügen Benutzer über Eingabegeräte wie Tastatur und Maus. Dadurch lassen sich sehr viel mehr Ideen und Konzepte im Bezug auf Design, Interaktivität mit dem Benutzer und Bereitstellung von Informationen realisieren. Es können Rückmeldungen an und von Nutzern über Popup-Fenster realisiert werden und Formulare bereitgestellt werden, die auszufüllen sind.

Da auf mobilen Endgeräten Benutzereingaben entweder über eine sehr kleine mechanische Tastatur oder über einen Touchscreen erfolgen und diese für aufwändige Eingaben nur bedingt geeignet sind, sollte eine derartige Interaktion möglichst gering gehalten werden. Es soll Anwendern möglich sein, mit wenigen Klicks und minimaler Interaktion an die gewünschten Informationen zu gelangen. Anstatt also von einem Benutzer Eingaben dahingehend zu verlangen, was er als nächstes tun möchte oder welche Daten er benötigt, sollte der Benutzer einfach aus Listen mit Vorschlägen und Themen auswählen können.

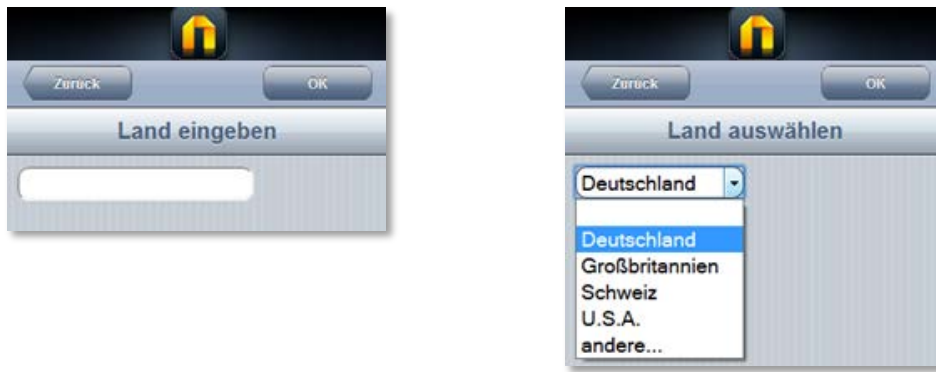


Abbildung 8: Vergleich Eingabe vs. Auswahlliste

3.4. Navigation

Als Folge der im vorherigen Abschnitt besprochenen Besonderheiten bei der Bedienung mobiler Endgeräte ist auch die Navigation bei Anwendungen und Webseiten entsprechend zu gestalten. Einem Benutzer stehen keine Eingabegeräte wie z.B. eine Maus zur Verfügung, die für die Navigation in und die Steuerung von Software und Webinhalten genutzt werden könnten. Elemente lassen sich also nicht punktgenau lokalisieren und nicht in der Art verwenden, wie dies im Desktopumfeld möglich ist. Auch die schnelle Überbrückung größerer Bildschirmabstände mit Hilfe einer Maus ist bei einem mobilen Endgerät nicht möglich. Um das Fehlen derartiger Steuerungsoptionen zu kompensieren, müssen äquivalente Navigationselemente bereitgestellt werden, die dieselben Funktionen mit geringerem Aufwand bereitstellen.

Beispielsweise sollten sich im Darstellungsbereich immer Links befinden, die es ermöglichen, eine Seite bzw. einen Schritt zurück zu springen. Auch zum Anfang einer Seite sollte man springen können, für den Fall, dass andernfalls ein Scrollen notwendig wäre. Ein Home-Button zum Ausgangspunkt sollte ebenfalls vorhanden sein.

Im mobilen Umfeld wollen Benutzer möglichst schnell an gewünschte Informationen gelangen. Aus diesem Grund sollte ein Mittelweg in Bezug auf die Anzahl an vorhandenen Links gefunden werden: Einerseits sollten sich möglichst wenige Links im Darstellungsbereich einer Seite befinden, damit der Benutzer den von ihm gewünschten Link sofort findet. Ist es realisierbar, alle notwendigen Navigationslinks auf den aktuellen Anzeigebereich des Gerätes zu beschränken, entfällt eine weitere Navigation über Scrollen. Hierdurch können auch ungewollte Nebenwirkungen wie ein versehentliches Drücken eines Links während dem Scrollen verhindert werden. Andererseits sollten sich jedoch ausreichend Navigationsmöglichkeiten und Links im Anzeigebereich befinden, um einem Anwender umständliche Wege zu ersparen und ihn schnell an seine Ziele zu bringen.

Als Empfehlung des W3C existieren folgende zwei Strategien zum Umsetzen einer mobilen Navigation:

Um mehrere Anwendungen bereitzustellen, sollte als Einstiegspunkt (Startseite) ein Inhaltsverzeichnis verwendet werden, das die zur Verfügung stehenden Anwendungen auflistet und über einen Link die Möglichkeit bietet, direkt in diese Anwendung zu gelangen. Innerhalb einer Anwendung sollte sich im Kopf der Seite ein flaches Menü befinden, das über 2-3 Links verfügt, wie einen Zurück-Button oder die Möglichkeit zum Sprung auf eine Seite für einen neuen Datensatz. Sollten weitere

Navigationsmöglichkeiten notwendig sein, sind diese am Ende der Seite unterhalb des Informationsbereichs zu platzieren.

Somit wird auch bei Geräten, die den Inhalt in reinen HTML-Strukturen ohne CSS darstellen, gewährleistet, dass der Anzeigebereich nicht mit einer Vielzahl von Links gefüllt ist, die den Platz für den eigentlichen, informativen Seiteninhalt einschränken. Die Umsetzung und Darstellung sollte mit Hilfe eines flachen Menüs erfolgen. In diesem Menü können sich Menüpunkte oder direkte Links zu Startseiten von einzelnen Anwendungen befinden. Sollten weitere Elemente benötigt werden (z.B. weitere Menüordner), sind diese am Ende der Seite unterhalb des Informationsbereichs zu platzieren.

3.5. Layout

Als Folge immer weiter steigender, zur Verfügung stehender Bandbreiten stehen Designern alle Tore offen, um auch aufwändige Designs in einem Desktopumfeld umzusetzen. Layouts können mit einer Vielzahl an Grafiken, Videos, Animationen beispielsweise mit Flash und anderen Hilfsmitteln wie JavaScript realisiert werden. Das dabei entstehende Datenaufkommen spielt meist nur eine untergeordnete Rolle, da vor allem in Intranets zu jeder Zeit ausreichend Bandbreite zur Verfügung steht.

Im Umfeld mobiler Geräte ergibt sich jedoch eine völlig andere Ausgangssituation. Zum Einen unterscheiden sich die Geräte hinsichtlich ihrer technischen Möglichkeiten deutlich. Nicht jedes Gerät ist für die Ausführung von JavaScript oder Flash geeignet. Zum Anderen stehen bei mobilen Zugriffen ins Internet nur geringe Bandbreiten zur Verfügung. Nicht zu vernachlässigen sind beim Zugriff auf das mobile Internet außerdem die anfallenden Kosten. Vielerorts existieren immer noch sehr hohe Preise für die im Internet eingewählte Zeit und das dabei aufkommende Datenvolumen.

Daher muss bereits beim Design darauf geachtet werden, das Volumen der zu übertragenden Daten möglichst gering zu halten.

Sollten Bilder verwendet werden, ist darauf zu achten, dass diese auf dem zugreifenden Gerät darstellbar sind. Die verwendeten Bildformate sollten JPEG oder GIF89a sein. Große und hochauflösende Bilder sind zu vermeiden. Stattdessen sollten skalierte Bilder mit einer geringeren Auflösung oder nur Ausschnitte des ursprünglichen Originals verwendet werden, wenn dadurch keine wichtigen Informationen verloren gehen.



Abbildung 9: Originalbild vs. Ausschnitt mit geringerer Auflösung, jedoch mit gleichem informativem Inhalt

Ein weiterer zu beachtender Punkt ist die Aufteilung des Darstellungsbereichs auf die zu zeigenden Inhalte. Im Desktop-Umfeld finden sich praktisch immer mehrspaltige Layouts. Seien es mehrere Spalten mit den anzuzeigenden Informationen oder jeweils ein Spalte für Navigation und Inhalt. Im mobilen Umfeld sollten Inhalte immer als einspaltige Liste dargestellt werden, um eine optimale Navigation, Bedienung und Darstellung zu gewährleisten.

3.6. JavaScript

Die Verwendung von JavaScript sollte auf ein notwendiges Minimum beschränkt werden. Bei der Verwendung von JavaScript besteht das Risiko, dass das zugreifende Gerät dies nicht unterstützt oder manuell abgeschaltet wurde und es so einem Benutzer nicht oder nur eingeschränkt möglich ist, den zur Verfügung gestellten Service zu nutzen. Zusätzlich erhöht sich durch das notwendige Laden des JavaScript-Codes das Datenaufkommen und somit auch die gesamte Ladezeit der damit verbundenen Anwendung. Denn für jede JavaScript-Datei entstehen auf Grund der notwendigen Anfrage beim Server Latenzzeiten, die je nach Standort und Bandbreite einige Zeit in Anspruch nehmen können.

Ein weiterer Punkt der hierbei zu beachten ist, ist der "Aufwand", der betrieben werden muss, um JavaScript verarbeiten zu können. Hierbei entstehen Belastung für Prozessor und Speicher, die auf Grund teilweise geringer Leistung der entsprechenden Geräte zu langen Verarbeitungszeiten führen können. Dieser Rechenaufwand führt zusätzlich zu einer höheren Belastung der Batterien bzw. Akkus und resultiert in einer geringeren Lauf- und Nutzungszeit des mobilen Gerätes.

3.7. Zu vermeidende Elemente

Folgende Elemente sind im Umfeld mobiler Anwendungen bzw. Geräte auf alle Fälle zu vermeiden:

Frames

In HTML werden Frames verwendet, um mehrere HTML-Dokumente in einer Ansicht zusammenzufassen und so eine anwendungsähnliche Oberfläche zu schaffen.² Abgesehen davon, dass viele Geräte Frames nicht unterstützen, führt diese Art der Aufbereitung einer Ansicht im mobilen Umfeld zu unzähligen weiteren Problemen. Hierzu zählen z.B. eine nicht intuitive oder sogar falsche Funktionsweise des Zurück-Buttons oder die fehlende Möglichkeit, Lesezeichen anzulegen. Zudem wird die Suche nach Inhalten erschwert.

Popups

Das Öffnen von Seiten in einem weiteren Fenster sollte in Anwendungen vermieden werden. Viele Geräte unterstützen jeweils nur ein Anwendungs- bzw. Browserfenster. Daher kann das Öffnen anderer Seiten auf solchen Geräten zu nicht vorhersehbaren Ergebnissen führen. Zudem führen Popups zu längeren Verarbeitungszeiten und damit zu höheren Verbindungskosten. Zusätzlich verschlechtern sich der Bedienkomfort und die Übersichtlichkeit für Benutzer, da die Verwendung und Navigation über mehrere Seiten/Fenster erfolgen muss.

Layouttabellen und Spacer-GIFs

Zur Darstellung von Seiteninhalten sollten keine Tabellen (Layouttabellen, verschachtelte Tabellen...) und 1px-Grafiken verwendet werden. Auf mobilen Geräten führt die Verwendung von Tabellen zu unerwünschten Seiteneffekten. Wird eine Tabelle vollständig dargestellt und überschreitet sie eine gewisse Breite, müssen Benutzer horizontal und vertikal scrollen, um Inhalte anzusehen. Diese Art der Navigation ist gerade bei einem mobilen Endgerät äußerst umständlich und sollte vermieden werden.

Ebenfalls kann es passieren, dass der verwendete Browser die Tabelle beim Überschreiten der Displaybreite umbricht und die fehlenden Spalten in einer neuen Zeile einfügt. Dies geht zu Lasten der Übersichtlichkeit. Außerdem vergrößert sich durch die Verwendung von Tabellen der zu Grunde liegende HTML-Code erheblich. Dies hat einen aufgeblähten, unsauberen Code zur Folge und führt darüber hinaus zu einem erhöhten Datenvolumen.

Imagemaps

Auf Imagemaps sollte gänzlich verzichtet werden, bzw. sollten diese laut Empfehlung des W3C nur dann eingesetzt werden, wenn sichergestellt ist, dass das auf den Inhalt zugreifende Geräte Imagemaps technisch unterstützt. Anstelle von Imagemaps sollten Textlinks eingesetzt werden.

² http://www.w3.org/TR/xframes/#s_intro

4. HTML-Strukturen

Beim Umsetzen von Seiten und Layouts sollte man sich nicht nur auf das rein optische Ergebnis konzentrieren. Man sollte im Gesamten darauf achten, dass die drei für ein gutes Ergebnis notwendigen "Säulen" *Inhalt*, *Design* und *Struktur* strikt voneinander getrennt bleiben und auf alle drei Punkte in gleichem Maße Wert gelegt wird. In Hinblick auf die *Struktur* sollte man darauf achten, dass das zugrunde liegende Konstrukt an HTML-Elementen eine gesunde und möglichst semantische Struktur besitzt. Doch was genau ist darunter zu verstehen?

Das World Wide Web (WWW) stellt eine mittlerweile nicht mehr überschaubare Masse an Inhalten und Information für Benutzer weltweit bereit. Diese Daten stehen innerhalb des WWW normalerweise jedoch in keinerlei Beziehung zueinander. Das WWW bietet lediglich eine Art Präsentationsplattform, die die erfassten Daten darstellt.

Um Inhalte tatsächlich als solche zu erkennen und erfasste Daten entsprechend deuten zu können, entwickelte Tim Berners-Lee (der "Erfinder" des WWW) das Konzept des *Semantic Web*. Dessen Idee beruht darauf, dass das Netz aus Daten besteht, die direkt oder indirekt von Maschinen interpretiert und verarbeitet werden können.

Ein Punkt, der als Baustein der Vision vom semantischen Web angesehen werden kann, ist *semantisches HTML*. Die Idee des semantischen HTML lässt sich an einem einfachen Beispiel erläutern:



Zu sehen ist hier zweimal der Firmenname „United Planet“. Beide Darstellungen unterscheiden sich innerhalb eines Browsers optisch nicht voneinander – von ihrer Erzeugung in HTML sind sie jedoch grundlegend verschieden. So wurde die Formatierung einer Überschrift manuell vorgenommen. Es wurde eine eigene Schriftgröße und die Formatierung *fett* für den Text angegeben. Bei dieser Art der Textformatierung ist es nicht möglich, die Überschrift als solche zu interpretieren. Suchmaschinen erkennen nicht die Relevanz und Struktur des Inhalts. Auch Screenreadern bietet sich nicht die Möglichkeit, den Inhalt als solchen zu identifizieren und Anwendern entsprechende Rückmeldung zu geben, da über HTML nur eine optische, aber keine inhaltliche Struktur erzeugt wurde.

Die zweite Überschrift wurde mit dem HTML-Tag `<h1>` erzeugt. Zwar ergibt sich hierdurch das gleiche optische Ergebnis. Jedoch kennzeichnet dieses Tag den im folgenden Element enthaltenen Text eindeutig als Überschrift. D.h. Suchmaschinen und Screenreader können den Inhalt deuten, zuordnen und entsprechend interpretieren. Zudem besitzen praktisch alle mobilen Endgeräte ein Standard-Rendering, um beispielsweise bei abgeschaltetem CSS oder fehlenden technischen Möglichkeiten ein Mindestmaß an HTML-Darstellung von Standardtags wie `<h1>` zu gewährleisten.

Ein weiterer Vorteil einer gesunden und semantischen Struktur sind die besseren Stylemöglichkeiten mit CSS. Gestaltungsprobleme, die zwangsweise durch fehlerhaften, unstrukturierten oder aufgeblähten Code entstehen, können so vermieden werden.

5. Layouts und Applikationen mit Intrexx

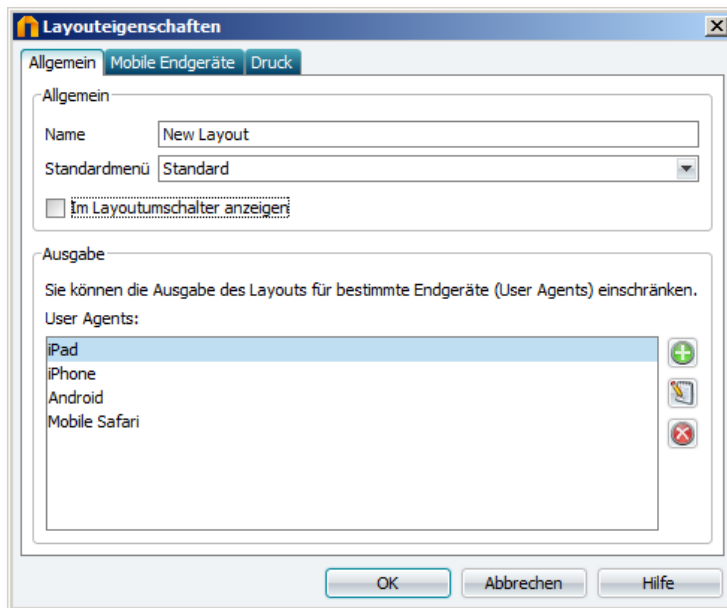
5.1. Layouts

Beim Erstellen von Layouts für mobile Endgeräte haben Sie die Möglichkeit, mit Hilfe des Layoutvorlagen-Assistenten eine Layout-Vorlage für mobile Endgeräte auszuwählen oder auf Basis des leeren Layouts ein komplett eigenes Layout zu entwickeln.

Haben Sie sich für die Vorlage entschieden, gehen Sie bitte wie folgt vor: Öffnen Sie im Modul *Design* eine Vorlage über *Datei / Layout Neu* oder mit Hilfe der Tastenkombination *STRG+N*. Im sich öffnenden Layoutmanager-Dialog können Sie in der Vorlagenkategorie *Mobil* aus Vorlagen für iPhone und BlackBerry wählen. Treffen Sie eine Auswahl durch einen Klick auf das gewünschte Symbol.

Die für Mobilelayouts notwendigen Einstellungen (wie bestimmte Kontrollen und Stilklassen) wurden in den bereitgestellten Vorlagen bereits vorgenommen. Durch die Auswahl einer Vorlage und dem Veröffentlichen des Layouts auf dem Portal Server haben Sie in kürzester Zeit ein voll funktionsfähiges und optisch ansprechendes Layout - sowohl für den Einsatz im Desktop-Bereich als auch für mobile Endgeräte.

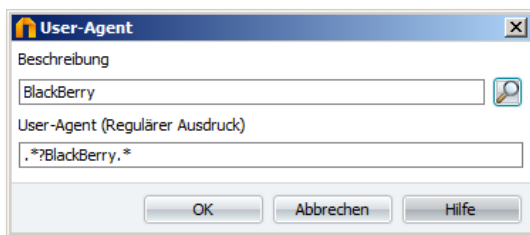
Sollten Sie sich jedoch dazu entschieden haben, ein eigenes Layout ohne Vorlage zu erstellen, müssen Sie einige Grundeinstellungen manuell vornehmen. Im ersten Schritt öffnen Sie dazu über *Datei / Layouteigenschaften* den Eigenschaftendialog des aktuellen Layouts.




In diesem Dialog können Sie auf dem Reiter *Allgemein* einen Namen für das Layout vergeben und das zu verwendende Standardmenü angeben. Wählen Sie hier den Menüpunkt *Mobile* bzw. jenen Menüpunkt, der beim Aufruf des Portals über ein mobiles Endgerät aufgerufen werden soll. Im Kapitel *Applikationen* wird nochmals genauer auf die Standardmenüs eingegangen.

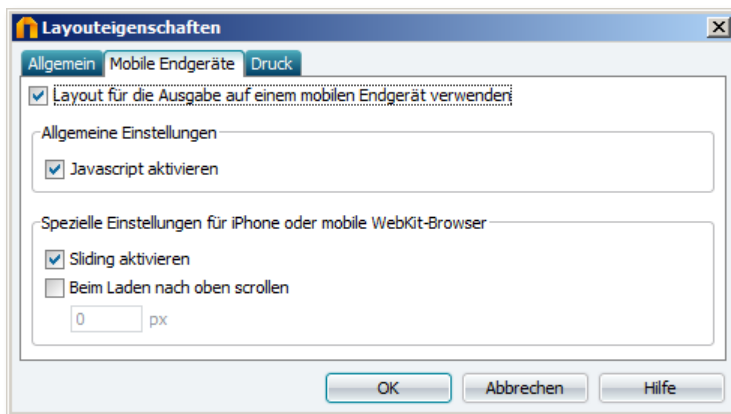
- Sie können in einem Layout z.B. ein flaches Menü anlegen und dieses über die Einstellung *Menü* einem Mobil-Menüpunkt zuweisen. Somit würde im mobilen Layout zu jeder Zeit ein Menü angezeigt werden. Da dies aus Layout- und Usabilitygründen nicht immer gewünscht ist, können die gleichen Funktionen mit Hilfe von speziellen Zusatzkontrollen für mobile Endgeräte (z.B. *Inhaltsverzeichnis (mobile Endgeräte)*) realisiert werden.

Im Bereich *Ausgabe* können Sie festlegen, welche Endgeräte dieses Layout beim Zugriff verwenden sollen. Hierbei definieren Sie sogenannte *Benutzeragenten (User Agents)*. Greift ein Endgerät auf ein Portal zu, sendet dieses Gerät einen Request-Header, in dem sich Informationen über das zugreifende Gerät befinden. Damit ist es möglich, das Gerät zu identifizieren und somit Layouts entsprechend des Zugriffsgerätes zuzuweisen.



Es sind bereits einige Benutzeragenten vordefiniert. Sie können jedoch mit einem Klick auf  *User Agents auswählen* beliebige weitere Agenten anlegen. Hierzu müssen Sie eine Beschreibung angeben sowie einen regulären Ausdruck. Nach dem hier angegebenen regulären Ausdruck wird im Benutzeragenten-String gesucht. Sollte der Ausdruck gefunden werden, wird das damit assoziierte Layout verwendet.

Wechseln Sie anschließend im Dialog *Layouteigenschaften* auf den Reiter *Mobile Endgeräte*.



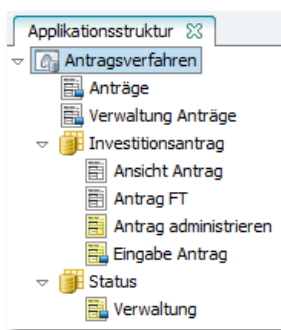
Hier kann definiert werden, ob das Layout für mobile Endgeräte verwendet werden soll. Ist diese Option markiert, können Sie zusätzlich angeben, ob Javascript und im Falle eines iPhone-Layouts das *Sliding* aktiviert werden soll. In diesem Modus wird die Navigation innerhalb einer Anwendung mit dem iPhone-typischen Ein- und Ausblenden von links und rechts visualisiert.

5.2. Applikationen

Um mit mobilen Endgeräten auf Applikationen eines Portals zugreifen zu können, sind wie beim Layout einige Besonderheiten zu beachten, was verwendete Kontrollen, Styles und HTML-Strukturen anbelangt.

- Es wird dazu geraten, Anwendungsfälle für Desktop- und mobile Applikationen innerhalb einer Applikation abzubilden, statt für jeden Bereich eine separate Applikation zu erstellen.

Im Folgenden sollen diese Besonderheiten anhand einer Beispielanwendung aufgezeigt werden, mit der Investitionsanträge gestellt und von einem Verantwortlichen genehmigt oder abgelehnt werden können. Die in diesem Beispiel verwendete Applikation, die bisher noch rein auf eine Verwendung im Desktop-Umfeld ausgelegt ist, verfügt über folgende Struktur an Seiten und Datengruppen:



Folgende Datenfelder (zusätzlich zu den Systemdatenfeldern) befinden sich in den beiden Datengruppen:

Datengruppe Investitionsantrag

Datenfeld	Datentyp
Antragsnummer	Integer
Name	String
Betreff	String
Betrag	Double
Begründung	Text
Status	Referenz auf Datengruppe Status (Status String)

Datengruppe Status

Datenfeld	Datentyp
Status	String

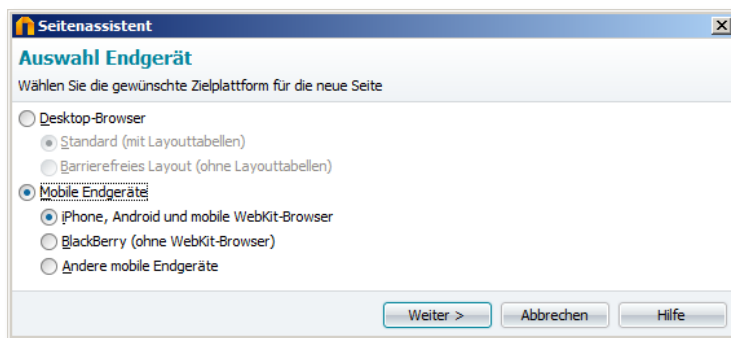
Die Applikation soll nun so erweitert werden, dass ein Antragsprüfer mit Hilfe seines mobilen Endgerätes direkt seine zu bearbeitenden Anträge abrufen und die Entscheidung über die zu tätigende Investition eingeben kann. Auf diese Weise können Entscheidungswege und -zeiten enorm verkürzt werden.


5.2.1. Seitenassistent

Um das Erstellen von Seiten für mobile Endgeräte und die Definition von sich wiederholenden Elementaktionen (z.B. Speichern über Schaltflächen) zu erleichtern und zu beschleunigen, bietet Intrexx in Version 5.0 einen Seitenassistenten, mit dessen Hilfe sich in kürzester Zeit vollständig funktionsfähige Seiten sowohl für Desktopapplikationen als auch für das mobile Umfeld erstellen lassen.

Befinden Sie sich im Modul Applikationen, können Sie bei einer geöffneten Applikation über das Menü *Neu / Seitenassistent* oder durch einen Klick mit der rechten Maustaste auf den Applikationsnamen oder eine Datengruppe in der Applikationsstruktur den Assistenten aufrufen.

Im ersten Dialog ist auszuwählen, für welches Endgerät eine Seite erstellt werden soll. In diesem Beispiel wird das iPhone als Zielplattform ausgesucht.



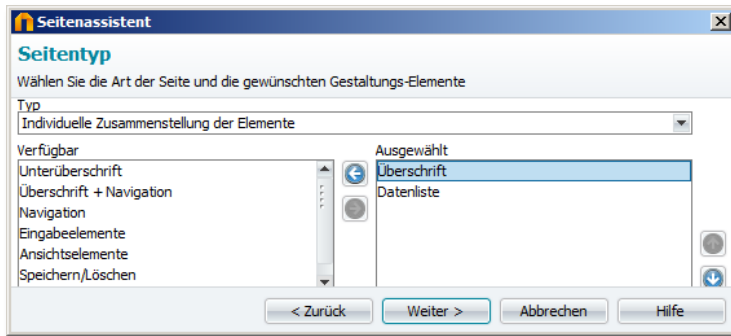
Klicken Sie auf  *Weiter*.

Sie können nun den Typ der Seite festlegen. Dies bedeutet, dass definiert werden kann, welche Elemente bzw. Bereiche auf dieser Seite platziert werden sollen.

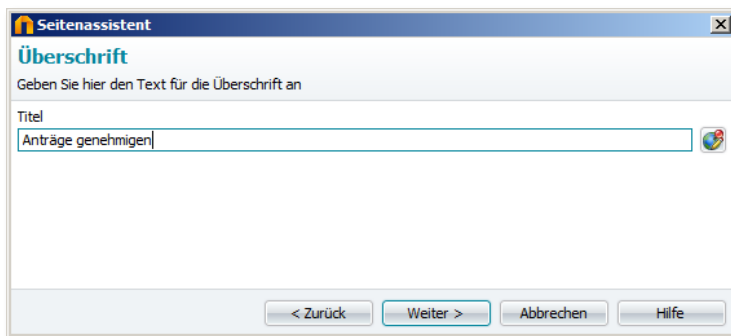
Es stehen hierbei folgende Elemente zur Auswahl:

Element	Beschreibung
Überschrift	Überschrift (Titel) zur Seite
Unterüberschrift	Text, der unterhalb der Überschrift angezeigt wird, z.B. als detailliertere Beschreibung der Seite
Navigation	Bereich für Schaltflächen/Links zur Navigation
Datenliste	Liste zur Anzeige vorhandener Daten einer Datengruppe
Suche	Suchelement

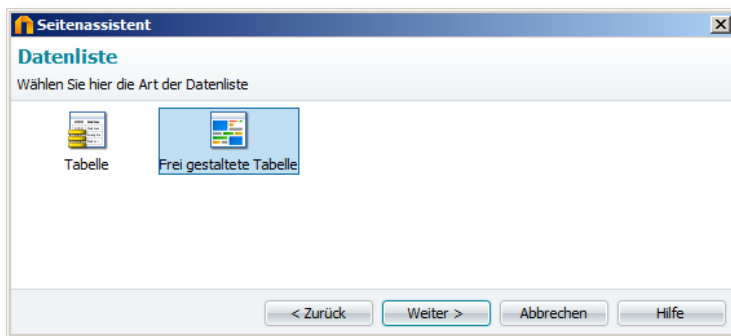
Da im aktuellen Beispiel zu bearbeitende Einträge für einen Prüfer angezeigt werden sollen, wählen Sie innerhalb dieses Dialogs die Elemente *Überschrift* und *Datenliste*.



Klicken Sie auf *Weiter*. Geben Sie den Text an, der als Überschrift auf der erstellten Seite angezeigt werden soll und bestätigen Sie mit *Weiter*.



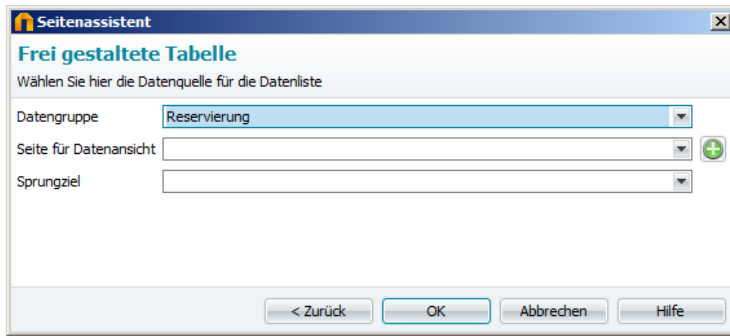
Nun kann ausgewählt werden, auf welche Art und Weise die vorhandenen Daten dargestellt werden sollen.



Wählen Sie den Typ *Frei gestaltete Tabelle*.

Abschließend muss festgelegt werden, welche Daten angezeigt werden sollen (bzw. im Falle der frei gestalteten Tabelle: welche Seite). Sollte noch keine Ansichtseite vorhanden sein, die in der freien Tabelle angezeigt werden kann, können Sie diese direkt innerhalb dieses Dialogs durch einen Klick auf *Ansichtseite erstellen* gestalten.

Das Sprungziel zur Detailansicht eines Antrags wird zu einem späteren Zeitpunkt definiert und erhält aktuell noch keinen Wert.



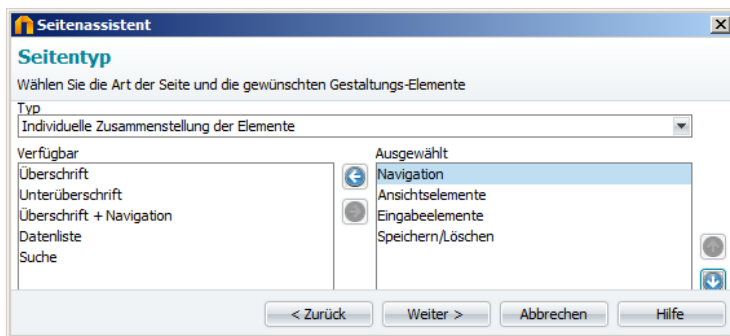
Bestätigen Sie abschließend die Einstellungen mit einem Klick auf **OK**.

Im letzten Schritt muss die freie Tabelle auf der soeben angelegten Seite auf jene Datensätze gefiltert werden, die noch zu bearbeiten sind. Bereits genehmigte oder abgelehnte Datensätze sollen nicht angezeigt werden. Definieren Sie dafür im Eigenschaftendialog der Tabelle einen Filter entsprechend der Status der Anträge (z.B. *Status = Neu*).

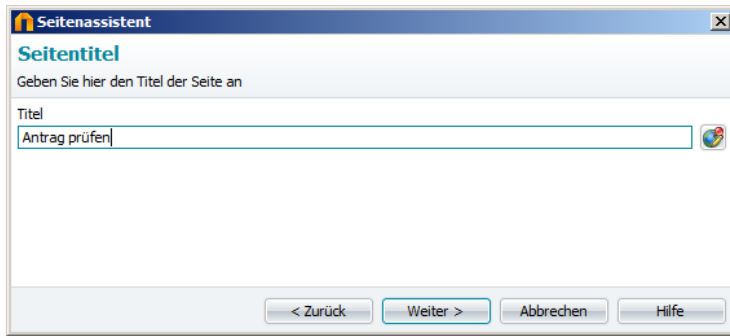
Zusätzlich benötigt wird eine Seite, die die Details eines Antrags auflistet und die Möglichkeit bietet, diesem angezeigten Antrag einen Status (z.B. *genehmigt* oder *abgelehnt*) zuzuordnen.

Öffnen Sie erneut den Seitenassistenten mit einem Rechtsklick auf die Datengruppe *Investitionsantrag* und wählen Sie erneut das iPhone als gewünschte Zielplattform.

Da für einen gestellten Antrag ein Status vergeben werden soll, werden Ansichtsfelder für den aktuellen Antrag sowie ein Eingabefeld für den vergebenen Status benötigt. Wählen Sie daher die im folgenden Screenshot gezeigten Gestaltungselemente (*Navigation, Ansichtselemente, Eingabelemente* und *Speichern/Löschen*):

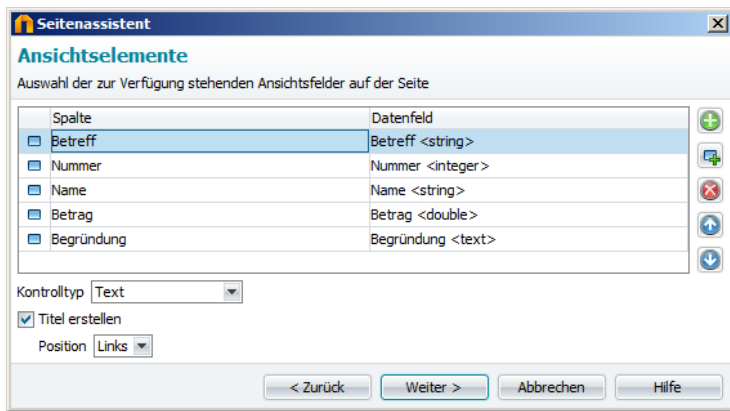


Nach einem Klick auf **Weiter** geben Sie der Seite einen Titel, z.B. *Antrag prüfen*.



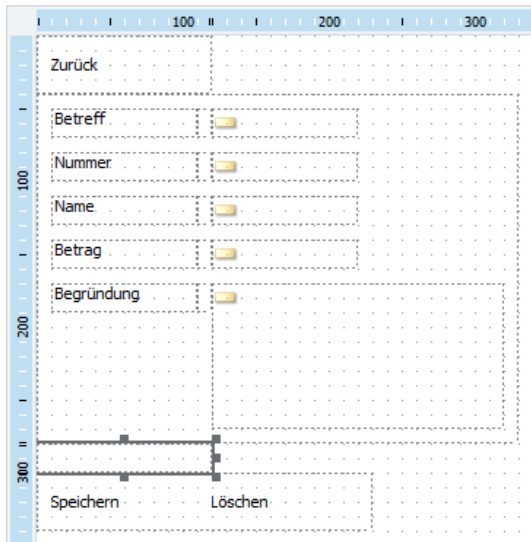
Zur Navigation wird eine Schaltfläche *Zurück* angelegt, die zurück auf die vorher angelegte Ansichtseite *Anträge zu genehmigen* springt.

Anschließend geben Sie die Ansichtsfelder an, die die Daten des zu überprüfenden Antrags anzeigen sollen. Wählen Sie die für Sie zur Prüfung relevanten bzw. verfügbaren Felder.



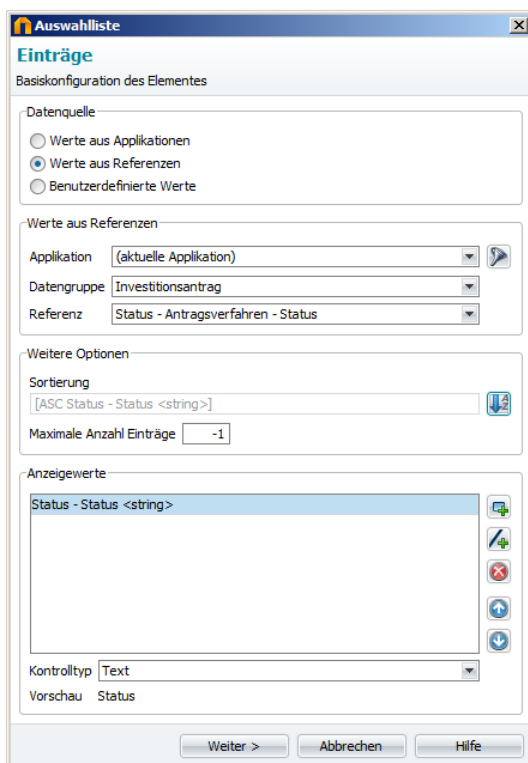
Im letzten Schritt sind die gewünschten Eingabelemente zu definieren. Da der Status eines Antrags über eine Referenz-Auswahlliste angegeben werden soll, wählen Sie in diesem Dialog kein Element aus und schließen den Assistenten mit OK.

Die vom Assistenten erstellte Seite sollte nun in etwa so aussehen:



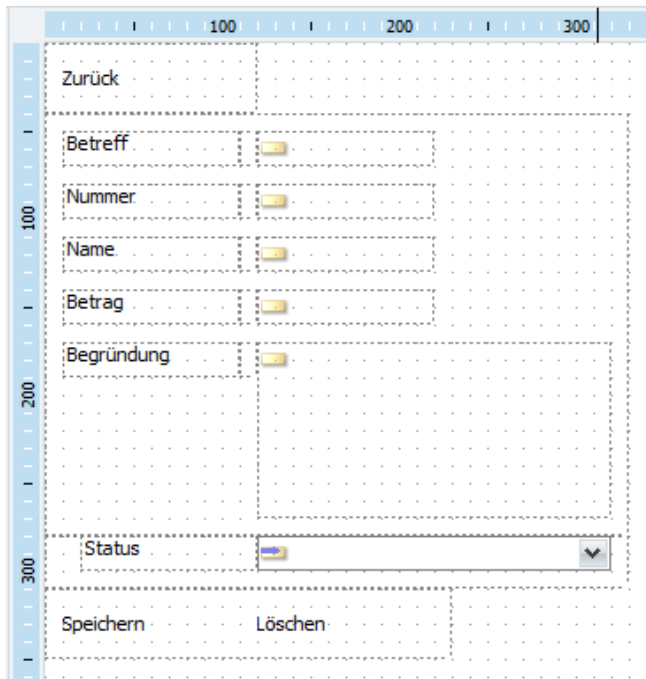
Die in diesem Screenshot markierte Gruppierung ist jene, die nun noch mit einer Referenzauswahlliste versehen werden soll. In unserer Beispiel-Applikation existiert in der Datengruppe *Investitionsanträge* eine Referenz auf die Datengruppe *Status*.


Fügen Sie das Eingabeelement *Auswahlliste* mit *Werte aus Referenzen* in die oben markierte Gruppierung ein.



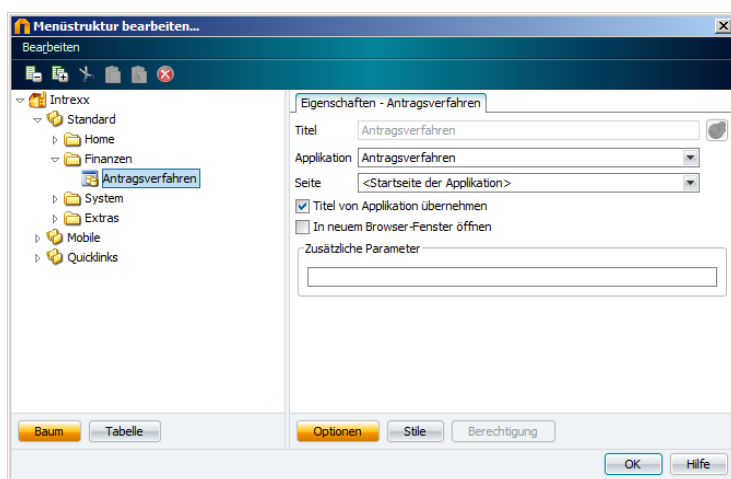
Anschließend wählen Sie die Option *Verknüpfung mit bestehendem Datenfeld* und verlassen die Einstellungsdialoge über *Fertigstellen*.


Die fertige Seite sollte zum aktuellen Zeitpunkt in etwa wie folgt aussehen:



 Hierbei ist folgendes zu beachten: Die Platzierung und das Layout von angelegten Elementen ergibt sich durch die zu Grunde liegende HTML-Struktur und durch den Einsatz von CSS. Die Höhe und Breite von Gruppierungen sowie von Eingabe- und Ansichtsfeldern hat keine Auswirkung auf die endgültige Darstellung auf einem mobilen Endgerät.

Öffnen Sie nun über das Menü *Extras / Menüstruktur bearbeiten* den Menüdesigner und fügen Sie die Applikation *Antragsverfahren* zu dem Menüpunkt hinzu, dem ein mobiles Layout zugeordnet wurde und der somit beim Aufruf über ein mobiles Endgerät ausgewählt wird.



Schließen Sie den Dialog mit einem Klick auf  **OK**.

Beim Zugriff mit einem mobilen Gerät erhalten Sie anschließend folgende Übersichtsseiten, ohne dass weitere manuelle Anpassungen an Funktionalität oder Layout notwendig sind:



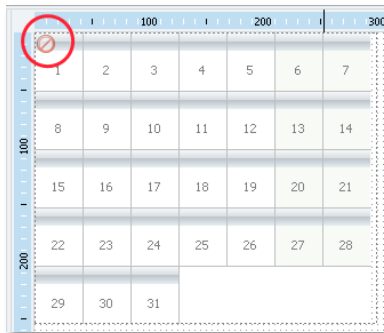
5.2.2. Seitenassistent Fazit

Im hier durchgespielten Beispiel wurde gezeigt, wie alleine durch das zweimalige Ausführen des Seitenassistenten innerhalb weniger Minuten eine Anwendung für den mobilen Einsatz erweitert wurde. Anwender benötigen hierbei keinerlei Kenntnisse von der Entwicklung für mobile Endgeräte. Der Intrexx Seitenassistent übernimmt die Zusammenstellung der gewünschten Elemente sowie deren Positionierung und die Zuweisung typischer Stilklassen für mobile Endgeräte.

5.2.3. Kontrollen

Wird eine Seite für mobile Endgeräte transformiert, gibt es einige Kontrollen, die anschließend nicht verwendet werden können, da sie keine oder nur eine eingeschränkte Unterstützung für mobile Plattformen bieten. Beispielsweise können Schaltflächen, die als Aktionskontrolle für Mobiles angelegt werden, nur über Funktionen wie Speichern, Löschen und Sprünge verfügen.

Existiert eine Kontrolle, die nicht über eine Mobileunterstützung verfügt, auf einer Applikationsseite, wird die entsprechende Kontrolle mit einem Hinweis-Icon versehen.



Verfügt eine Seite über solch eine Kontrolle, wird diese bei einem Aufruf über ein mobiles Endgerät nicht angezeigt. Die Seite an sich kann trotzdem aufgerufen werden.

Im Folgenden finden Sie eine Übersicht mit Kontrollen und Funktionen, die in Verbindung mit mobilen Anwendungen verwendet werden können:

Kontrolle	iPhone/Android BlackBerry mit WebKit	Mobile Geräte mit Javascript	Mobile Geräte ohne Javascript
Schaltflächen Speichern, Löschen, Sprünge	+*	+*	+*
Statische Filter	+	+	+
VTL-Include	+	+	+
Bilder	+	+	+
PDF/DOC ansehen	+	evtl.	evtl.
Applikationssuche	+	+	+
"einfache" Eingabekontrollen	+	+	+
Auswahllisten	+	+	+
Radiobutton	+	+	+
Checkbox	+	+	+
Ansicht Datei-/Bildauswahl	+	+	+
Longtext	ohne FCK-Editor	ohne FCK-Editor	ohne FCK-Editor
"einfache" Ansichtskontrollen	+	+	+
Ansichtstabelle (ohne Exportmöglichkeit und Spaltensortierung)	+	+	+
Freie Tabelle	+	+	+
Statischer Text	+	+	+
Benutzerinfos	nur Benutzername	nur Benutzername	nur Benutzername
Benutzerdefiniertes Javascript	+	+	-
Webservice-Aufruf beim Laden einer Seite	+	+	+
DataPicker	+	+	-
"Zurück" zur letzten Seite	+	+	+
Toggle-Button	+	-	-
Spinning-Wheel zur Datumseingabe	+	-	-
Datensatznavigation per Swipe	+	+	
Diagramme/Charts	+	+	+
Neuer Datensatz aus freier Tabelle	+	+	+
Einblenden/Ausblenden flexibel	+	-	-

* Pro Seite können beliebig viele Schaltflächen existieren, die Sprünge auf andere Seiten realisieren, da für diese Art jeweils ein <a> -Tag generiert wird. Für Aktionen wie Speichern und Löschen wird eine Schaltflächenkontrolle generiert. Dies hat zur Folge,

dass bei nicht JavaScript-fähigen Geräten oder bei abgeschaltetem JavaScript nur eine Schaltflächenkontrolle pro Seite möglich ist. Sollten Sie also z.B. Speichern- und Löschenkontrollen benötigen, sind hierfür zwei unterschiedliche Seiten anzulegen.

5.2.4. Abhängigkeiten

Auf mobilen Endgeräten kann ein Vielzahl an Abhängigkeiten definiert und ausgeführt werden, um dynamische Suchen und Filterung von Datensätzen zu realisieren. Unterstützt werden Abhängigkeiten von Ansichtstabellen und freien Tabellen. Filterungen können selbsterklärend sowohl auf Textfelder, als auch auf Datums- oder Zahlenfelder erfolgen.

Die definierten Abhängigkeiten unterstützen die JavaScript-Events *onload* beim Laden einer Seite, *onchange*, *onfocus* und *onblur* bei Eingabekontrollen und *onclick* bei Aktionskontrollen (z.B. Schaltflächen).

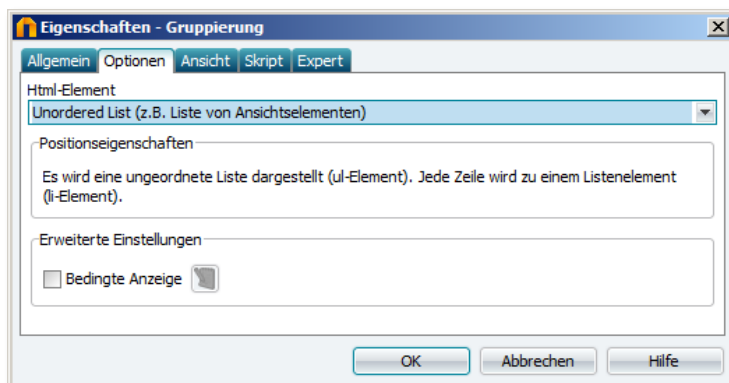
- Bei der Ausführung einer Filterung wird nicht nur der Tabellencontainer, sondern die gesamte Seite neu geladen.

Dahingehend zu beachten ist ebenfalls, dass beim Triggern von Abhängigkeiten über *onclick* einer Schaltfläche dieselbe Seite wiederum als Zielseite eingestellt werden muss. Beim Triggern über *onclick* bei Links muss hier zusätzlich im Eigenschaftendialog das Attribut *submit=true* gesetzt werden.

5.2.5. HTML-Elemente und Styles

Bei Seiten, die mit Unterstützung des Seitenassistenten erstellt wurden, werden Einstellungen wie die verwendeten HTML-Elemente und Styles automatisch vorgenommen. Sollten benutzerdefinierte Einstellungen gewünscht sein, können diese über die Eigenschaftendialoge der einzelnen Kontrollen getätigt werden.

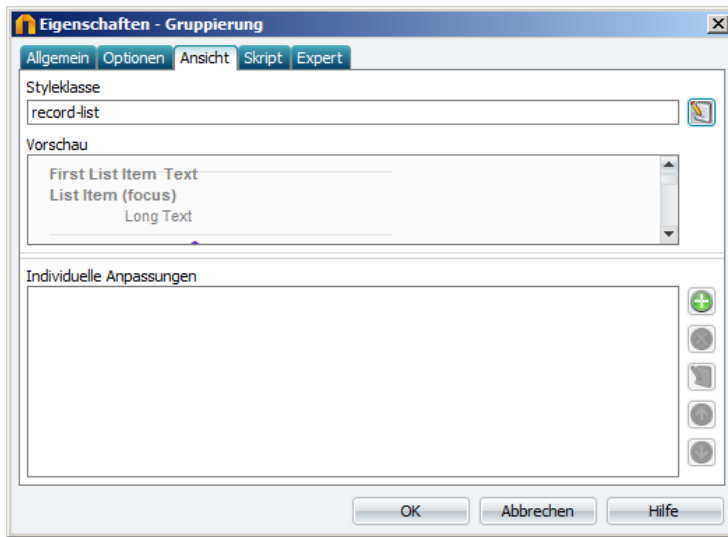
Öffnen Sie hierzu den Eigenschaftendialog eines Elements und wählen Sie den Reiter *Optionen*.



Über die Auswahlliste *HTML-Element* kann das HTML-Element gewählt werden, mit dem die Gruppierung und die in ihr enthaltenen Kontrollen angezeigt und aufgelistet werden. Wählen Sie hier das HTML-Element aus, das Ihren Ansprüchen an Funktionalität und Layout genügt.

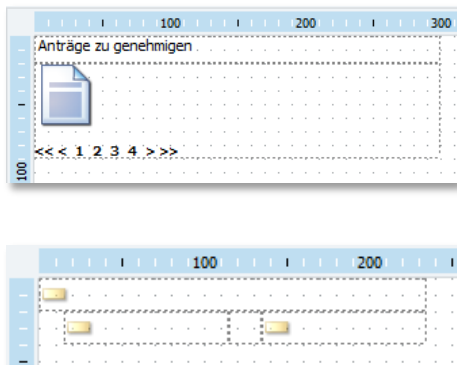
- Speziell bei der Entwicklung von Applikationen für das iPhone sollten vorwiegend Listen-Elemente verwendet werden, da andere iPhone-Layouts ebenfalls größtenteils auf Listen basieren.

Wechseln Sie nun auf den Reiter *Ansicht*.



Unter *Styleklasse* kann hier der für das momentan angewählte Element zu verwendende Style ausgewählt werden. Beispielsweise sind für ein iPhone-typisches Verhalten in der Layoutvorlage *iPhone* bereits einige vordefinierte Styleklassen hinterlegt. Der Style für Listen *Liste Datensätze – Edge to Edge (record-list)* beinhaltet eine Listendarstellung von Elementen inklusive Symbolen wie z.B. einem Pfeil zur Navigation.

Der folgende Screenshot zeigt die Startseite der Applikation. Auf dieser Seite befindet sich eine freie Tabelle, die die darunter abgebildete Ansichtseite anzeigt. Der freien Tabelle wird hierbei der Style *Liste Datensätze – Edge to Edge (record-list)* zugewiesen.

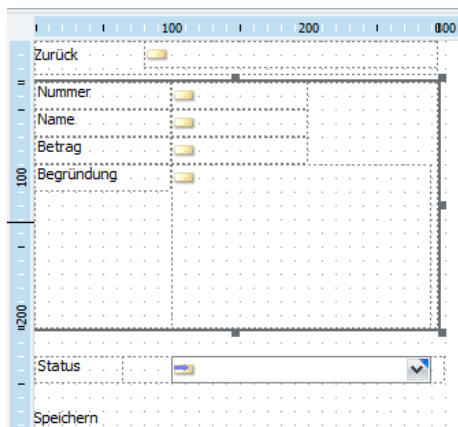


Daraus ergibt sich folgende Ansicht beim Zugriff mit einem iPhone:




Als weiterer Standard-Style existiert der Style *Liste Ansichtsfelder – Rounded Rectangle*. Hierbei werden umgebende Gruppierungen im iPhone-Standard-Style mit abgerundeten Ecken dargestellt.

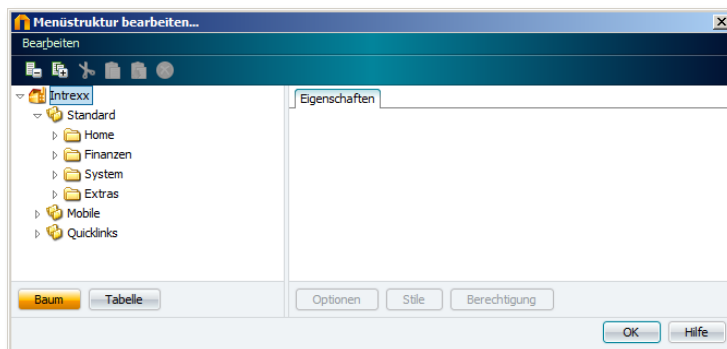
Folgendes Bild zeigt die Ansichtsseite für die Kundendetails. Der markierten Gruppierung wurde der Style *Liste Ansichtsfelder – Rounded Rectangle* zugewiesen.








5.2.6. Menüstruktur bearbeiten

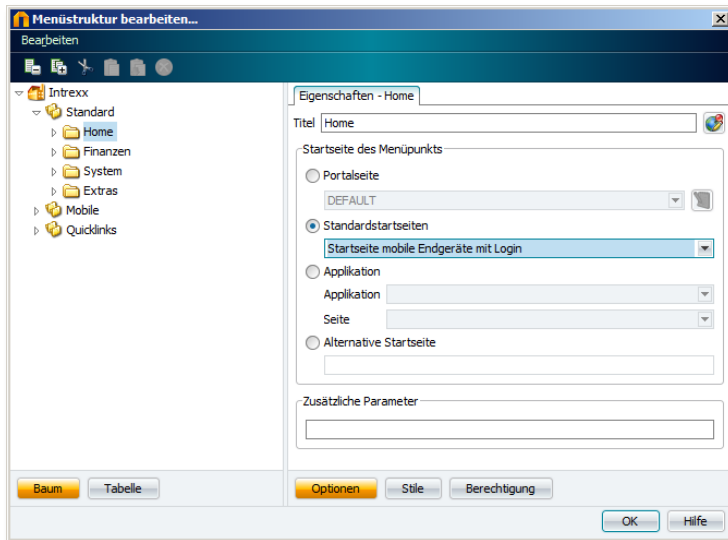
Im Intrexx Portal Manager können Sie über  *Extras / Menüstruktur bearbeiten* den Dialog zur Bearbeitung der Menüstruktur öffnen.



Im linken Fenster sehen Sie den Namen des aktuellen Portals. Als Untermenüpunkte gibt es *Standard*, *Mobile* und *Quicklinks*.

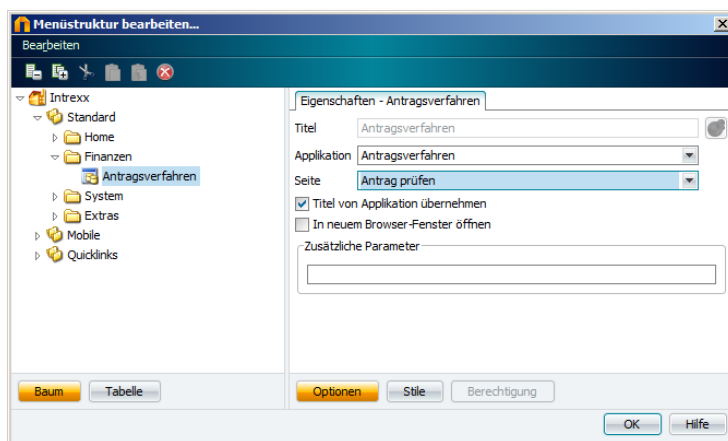
-  Die hier vorhandenen Menüs sind im Modul *Design* bei einem geöffneten Layout unter  *Layout / Layouteigenschaften* in der Auswahlliste *Standardmenü* aufgelistet.

Im Menüpunkt *Mobile* kann definiert werden, welche Menüpunkte und Applikationen bei einem Zugriff mit einem mobilen Endgerät verfügbar sein sollen. Mit einem Rechtsklick auf den Ordner *Mobile* können Sie über  *Neu* neue Menüordner, Applikationen und Links hinzufügen. Im Menü für Mobiles befindet sich bereits ein Ordner *Home*, dem als Standardstartseite die *Startseite mobile Endgeräte mit Login* zugewiesen ist.



Anschließend können Sie mit einem Rechtsklick auf *Home* diesem Menüpunkt eine Applikation hinzufügen. Weitere Informationen zum Menüdesigner finden Sie im Handbuch *Design*.

- Sie können für Applikationen verschiedene Startseiten anlegen, um unterschiedliche Startpunkte für Mobiles und für den Desktop-Zugriff zu erhalten. Die Startseite für den Zugriff über Desktop-PCs sollten Sie im Modul *Applikation* festlegen. Die Einstellung der Startseite für mobile Endgeräte sollte innerhalb des Menüdesigners über die Eigenschaften der Applikation erfolgen (Auswahlliste *Seite*).



Beachten Sie beim Design einer Anwendung ebenfalls, dass mobile Layouts normalerweise nicht über ein Applikationsmenü verfügen und so die Navigation mit mobilen Endgeräten über Links in der Applikation geregelt werden sollte.

5.3. Benutzerdefiniertes JavaScript

In Verbindung mit mobilen Applikationen kann auch von einem Benutzer definiertes JavaScript ausgeführt werden. Es besteht hierbei die Möglichkeit, die gewünschten JavaScript-Funktionen global oder gezielt eingeschränkt im jeweiligen Anwendungsfall einzubinden.

Beim Einsatz auf mobilen Endgeräten sind hierbei einige Besonderheiten zu beachten:

- JavaScript, das in Verbindung mit mobilen Endgeräten verwendet werden soll, kann nicht über den Intrexx JavaScript-Editor erfasst und gespeichert werden, da es sich um eine andere Art der Speicherung und der Verarbeitung handelt. Skripte sind manuell innerhalb von `<intrexx>/org/<portal>/external/htmlroot` abzulegen und zu referenzieren.
- Bei der Definition eines auslösenden Ereignisses und des damit verbundenen Skripts werden die in der referenzierten Datei verfügbaren Methoden nicht aufgelistet. Daher muss auf die korrekte Angabe des Methodenaufrufs geachtet werden.
- Innerhalb von JavaScript-Skripten stehen die aus dem Desktopumfeld bekannten Intrexx-Zugriffsobjekte nicht zur Verfügung (Ausnahme: `getElement("<GUID>")` und `Browser.setValueById(p_strId, p_value)`). Es können jedoch alle im ECMAScript-Standard definierten Zugriffsmethoden verwendet werden.

Beispielsweise sollte statt

```
Browser.getValue(getElement ("<GUID>"))
```

der Aufruf

```
getElement ("<GUID>").value
```

verwendet werden.

- Werden innerhalb einer Funktion die Events `onclick()` oder `onsubmit()` aufgerufen, muss die aufrufende Funktion über `return` einen booleschen Wert `true` oder `false` zurückgeben.

5.3.1. Globale Definition


Um benutzerdefinierte JavaScript-Methoden global in einem mobilen Portal einzubinden, müssen Sie das benutzerdefinierte Skript in die Datei `custom_head.vm` einbinden. Diese befindet sich unter `<intrexx>/org/<portal>/internal/system/vm/custom/`

Hier ist folgender Eintrag hinzuzufügen:

```
<script src="<pfad>/<javascript-Datei>.js" type="text/javascript"></script>
```

Es ist zu beachten, dass der Pfad zur gewünschten Datei hierbei relativ zum Verzeichnis `<intrexx>/org/<portal>/external/htmlroot` anzugeben ist, also z.B.

```
<script src="include/custom/myMobile.js" type="text/javascript"></script>
```

-  Wird eine Datei auf diese Weise eingebunden, wird sie beim Aufruf des Portals bereits geladen. Dies ist unabhängig davon, ob (und wenn ja, zu welchem Zeitpunkt) deren Methoden benötigt werden. Diese Art der Einbindung sollte daher nur erfolgen, wenn Methoden im Portal mehrfach verwendet werden.

Wird eine JavaScript-Funktion beispielsweise nur in einer Applikation verwendet, sollte sie auch nur in dieser eingebunden werden. Diese Vorgehensweise wird im folgenden Abschnitt genauer beschrieben.

Diese Art der Einbindung lädt die Datei bei jedem Aufruf des Portals, unabhängig vom zugreifenden Benutzeragenten. Um das Inkludieren auf mobile Endgeräte zu beschränken, kann eine Browserweiche eingebaut werden, die das Skript nur beim Zugriff eines mobilen Browsers lädt.

Beispiel:

```
#if ($Browser.getIPhone())
<script src="include/custom/iPhoneJS.js"
type="text/javascript"></script>


#elseif ($Browser.getMobile())
<script src="include/custom/mobileJS.js"
type="text/javascript"></script>

#end
```

Hiermit lassen sich gerätespezifische Skripte laden. Die Methode `$Browser.getMobile()` überprüft hierbei auf Benutzeragenten wie BlackBerry, PDA, Palm, Opera Mini und weitere.

5.3.2. Definition innerhalb einer Applikation

Um eine Funktion nicht global, sondern nur innerhalb einer Seite einer Applikation verfügbar zu machen, kann der Pfad zum benutzerdefinierten Skript auf dem Expertreiter der Seite angegeben werden.

-  Damit der Reiter Expert angezeigt wird, muss im Menüpunkt *Extras / Optionen* auf dem Reiter *Applikationen* die Option *Option für Experten aktivieren* markiert sein.

Soll beispielsweise ein Funktionsaufruf beim Laden einer Seite erfolgen, öffnen Sie den Eigenschaftendialog der gewünschten Seite und wechseln Sie auf den Reiter *Expert*. Hier können Sie in neues Expertattribut definieren, das Sie zum Einbinden von JavaScript bei mobilen Endgeräten benötigen:

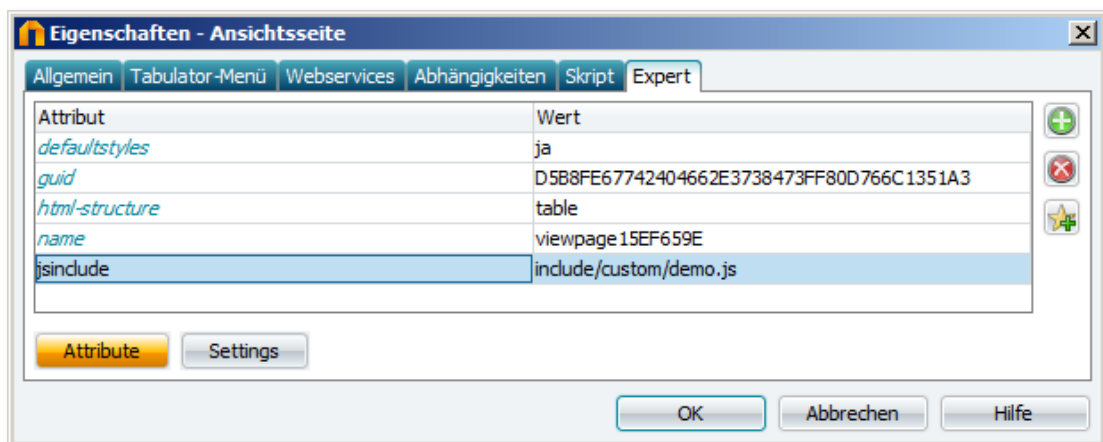
Name:

jsinclude

Wert:

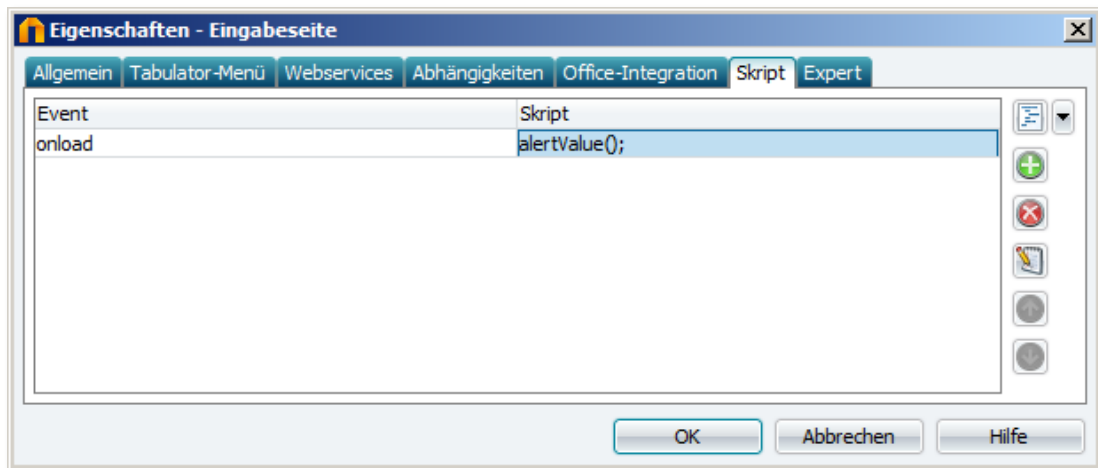
Pfad zur JavaScript-Datei. Angabe des Pfades relativ zu

 `<intrexx>/org/<portal>/external/htmlroot`



Die angegebene Datei kann nun innerhalb dieser Seite verwendet werden. Dies bedeutet, dass deren Funktionen sowohl bei Seitenereignissen wie *onload()* und *onunload()* sowie bei Kontrollen, die sich auf dieser Seite befinden, zur Verfügung stehen.

Im Eigenschaftendialog der Seite oder einer sich darauf befindlichen Kontrolle kann anschließend auf dem Reiter Skript ein gewünschter Funktionsaufruf hinterlegt werden.



6. Links (Stand 26.10.2009)

6.1. Einführung in mobiles Design

Einführung und zu beachtende Grundsätze

http://www.w3.org/2007/02/mwbp_flip_cards

Einführung, Definition, Geschichte

http://www.w3.org/2006/07/Mobile_Web_Design.pdf

Design Trends

<http://www.smashingmagazine.com/2009/01/13/mobile-web-design-trends-2009/>

6.2. Best Practices

Extended Guidelines for Mobile Web Best Practices

<http://www.w3.org/TR/mwbp-guidelines/>

Best Practices für das mobile Web: Empfehlung des W3C

<http://www.w3.org/TR/2008/REC-mobile-bp-20080729/>

Dokument von Luca Passani: Global authoring practices for the mobile web

<http://www.passani.it/gap/>

Mobile Web Best Practice in Kurzform

<http://www.utexas.edu/web/guidelines/mobile.html>

6.3. Tests

Übersicht von Testergebnissen der Mobile Web Test Suites Working Group zu verschiedenen mobilen Benutzeragenten

<http://www.w3.org/2007/03/mth/results?ts=cssmp>

Bereich für Mobile-Tests der Seite quirksmode von Peter-Paul Koch

<http://www.quirksmode.org/blog/archives/mobile/index.html>

6.4. Emulatoren

Online-Emulator, der den mobilen Zugriff auf Seiten und Anwendungen mit den Handys Sony Ericsson K750 und Nokia N70 simuliert

<http://mtld.mobi/emulator.php>

Online-Emulator für Opera Mini, die mobile Variante des Browsers Opera

<http://www.opera.com/mini/demo/>

Blackbaud iPhone Browser Simulator für Windows

<http://labs.blackbaud.com/NetCommunity/article?artid=662>

6.5. Apple

Richtlinien für die Entwicklung von Web-Applikationen

http://developer.apple.com/safari/library/documentation/InternetWeb/Conceptual/iPhoneWebAppHIG/Introduction/Introduction.html#//apple_ref/doc/uid/TP40007900-CH1-SW1

Webinhalte für das iPhone optimieren

http://developer.apple.com/safari/library/documentation/AppleApplications/Reference/SafariWebContent/OptimizingforSafariiPhone/OptimizingforSafariiPhone.html#//apple_ref/doc/uid/TP40006517-SW1

6.6. BlackBerry

Übersicht von Anleitungen und Handbüchern rund um BlackBerry

<http://docs.blackberry.com/en/>

7. Über Intrexx

Intrexx ist eine **integrierte plattformunabhängige Entwicklungsumgebung** zur schnellen und einfachen Erstellung und Verwaltung von z.B. multilingualen **Enterprise-, Kundenportalen oder Webapplikationen**. Intrexx ist **einfach erlernbar** und bedarf keiner Programmierkenntnisse. Das Erstellen des Portals erfolgt nach dem Drag & Drop Prinzip. Wer also eine Excel-Tabelle erstellen kann, der kann auch Anwendungen und Formulare wie z.B. ein Urlaubsantragsverfahren erstellen.



Neben dem Erstellen und Betreiben eines Portals lassen sich mit Intrexx ganz einfach vorhandene **Daten integrieren** und **Prozessabläufe** per Mausklick automatisieren. Durch die nahtlose Unterstützung von **mobilen Endgeräten** können alle Daten sehr schnell und verblüffend einfach für Smartphones wie iPhone oder BlackBerry zur Verfügung gestellt werden. So sind alle Mitarbeiter auch außer Haus ohne Medienbruch in alle Geschäftsprozesse eingebunden und ein optimaler Informationsfluss ist gewährleistet.

Intrexx ist dank unzähliger vorbereiteter Templates **schnell eingerichtet** und erlaubt den Aufbau eines Portals oder Intranets innerhalb kürzester Zeit.

Intrexx ist **komplett**. Es verfügt über alles, was man benötigt, um ein leistungsfähiges Portal zu entwickeln und erfolgreich zu betreiben.

Intrexx enthält (Auszug):

- Modul *Applikationen* zur Anwendungs- und Formularerstellung
- Modul *Design* zur Layouterstellung und Bearbeitung der Menüstruktur
- Modul *Prozesse* für das Abbilden von Workflows und das Erstellen mobiler Anwendungen
- Web Service Orchestrierung
- Komplette Benutzerverwaltung mit LDAP-Anbindung
- Datenintegration mit Einbindung externer Datenquellen (z.B. ERP-Daten)
- Volltext-Suchmaschine
- Link-Integration zur Einbindung von externen Webseiten
- Diverse Werkzeuge zur komfortablen Administration des Portals

Intrexx ist **Standardsoftware**, sehr oft installiert und somit auch sehr **preiswert**. Das System besteht im Wesentlichen aus zwei Teilen:

Intrexx Portal Manager: Er wird auf einem beliebigen Client oder auf dem Server installiert und verfügt über alle Komponenten um Layout, Menü oder Applikationen zu entwickeln und zu verwalten. Auch die Einrichtung der Benutzer mit den Rechten an den jeweiligen Anwendungen eines Portals erfolgt im Portal Manager von Intrexx.

Intrexx Portal Server: Er wird auf einem Server installiert und steuert alle Transaktionen der angelegten Webapplikationen und Portale. Er überwacht die Rechte der Benutzer innerhalb der Transaktionsvorgänge, steuert die gesamten Businesslogiken und regelt den Zugriff auf die Datenquellen.

Intrexx Professional ist grundsätzlich kostenlos per Download verfügbar. Die Testphase ist auf 30 Tage beschränkt, kann jedoch bei Bedarf entsprechend verlängert werden. Es gibt während dieser Zeit keinerlei Einschränkungen im Programm, alle Module, Anwendungen oder Applikationen können uneingeschränkt betrieben, getestet und verändert werden. Ebenfalls ist die Anzahl der User nicht beschränkt. Die verbleibende Testzeit wird immer in der rechten unteren Ecke des Portal Managers angezeigt. Am Ende der Testphase wird das System automatisch deaktiviert. Über eine spezielle Info-Seite kann bei Bedarf die Testdauer verlängert oder eine gültige Lizenz erworben werden. Alle während der Testphase angelegten Applikationen und erfassten Daten bleiben selbstverständlich erhalten und können weiterhin eingesetzt werden.

Für den Einsatz in einer Produktivumgebung werden für den Betrieb der Applikation oder des Portals entsprechende Benutzerlizenzen für den Intrexx Portal Server erforderlich.

Es stehen verschiedene Lizenzgrößen einschließlich einer Runtime-Lizenz zur Verfügung. Intrexx passt sich durch die **transparente Lizenzierung** jeder Unternehmensgröße an.

Im Intrexx Application Store befinden sich hunderte von freien und kostenpflichtigen Unternehmensanwendungen, die mit wenigen Klicks dem eigenen Portal hinzugefügt werden. Weiterhin gibt es auch fertige Lösungen zum Management von Unternehmensprozessen. Diese komplexen Webapplikationen werden Studio-Lösungen genannt. Sie sind in der Regel bereits im Intrexx Portal Manager als Applikationsvorlage enthalten, bedürfen aber im Falle einer produktiven Verwendung einer gesonderten Lizenzierung. Diese erfolgt durch eine einmalige Serverlizenz unabhängig von der Benutzeranzahl.

Der Abschluss eines (optionalen) Service- und Wartungsvertrags gewährleistet immer neueste Technologien und Releases und bietet effizienten Support durch United Planet.

Intrexx Professional steht unter www.intrexx.com/de/intrexx-5-professional kostenlos zum Download bereit.

8. Über United Planet

United Planet gehört mit über 4.000 Installationen und mehr als 500.000 Nutzern seiner Portalsoftware Intrexx allein im deutschsprachigen Raum zu den Marktführern im Segment der mittelständischen Wirtschaft, der öffentlichen Verwaltung und bei Organisationen (z.B. Kliniken). Geführt wird das Unternehmen von Lexware-Gründer Axel Wessendorf.

Mit der plattformunabhängigen Standardsoftware Intrexx lassen sich webbasierte Applikationen bis hin zu kompletten Intranets/Enterprise Portalen mit modernsten Funktionalitäten deutlich schneller erstellen als mit vergleichbaren Programmen wie z.B. Microsoft SharePoint.

Intrexx erlaubt die Einbindung vorhandener Daten aus ERP-Systemen, Microsoft Exchange, Lotus Notes etc., die Erstellung produktiver Workflows und die Generierung von mobilen Apps für Smartphones und Tablet PCs aller Hersteller. Im Intrexx Application Store stehen hunderte von fertigen Apps und komplette Portale zum Download bereit.

Kontakt

Postanschrift

United Planet GmbH
Postfach 1731
79017 Freiburg
Deutschland

Hausanschrift

United Planet GmbH
Schnewlinstr. 2
79098 Freiburg
Deutschland

Kommunikation

Telefon: +49-(0)761-20703-0
Telefax: +49-(0)761-20703-570
E-Mail: info@unitedplanet.com
Internet: www.unitedplanet.com

© Februar 2011. United Planet, Freiburg. Alle Rechte vorbehalten.