

Till Menke

Wettbewerb „TRIOD - Travellers' Realtime Information
mit Open Data“ von rnv und VRN

Mobilitätsfragen

Projektdokumentation

Till Menke

Kosselstr. 4 – 69115 Heidelberg – Tel. 06221/3538228 – studium@tillmenke.de

Veröffentlichungsdatum: 22.11.2017

<http://link.tillmenke.de/20171129-1>

Inhaltsverzeichnis

1	Kurzbeschreibung.....	5
2	Realweltproblem	6
2.1	Erfahrungsbeschreibung.....	6
2.2	Lösung durch dieses Projekt	7
2.2.1	Information an der Haltestelle.....	7
2.2.2	Information mit mobilen oder stationären Endgeräten	7
2.2.3	Einschränkungen der vorliegenden Lösung	7
3	Projekthistorie.....	8
4	Detailumsetzung	8
4.1	PHP-Serverdienst.....	8
4.1.1	Berechnung	8
4.1.2	Eingabe	18
4.1.3	Ausgabe	23
4.2	Konfiguration Raspberry Pi.....	32
4.3	Beispielimplementierung Ampel	33
4.3.1	mobilitaetsampel.py.....	33
4.3.2	Aufrufbefehl	34
4.4	bekannte Unzulänglichkeiten	34
4.4.1	Fahrt erreicht ihr Ziel nicht	34
4.4.2	Linie 721	34
4.4.3	Weiterfahrtalternativen	34

1 Kurzbeschreibung

Der in diesem Projekt entwickelte Dienst richtet sich an ÖPNV-Nutzer, die grundsätzlich wissen, wie sie ihr Ziel erreichen – etwa Pendler.

Anders als herkömmliche Fahrplanauskunftdienste – die sich an diejenigen richten, die überhaupt erst ihren Fahrplan erfahren möchten – informiert dieser Dienst anhand von Echtzeitdaten darüber, wie die Nutzer im konkreten Fall ihr Ziel schnellstmöglich erreichen und scheut dabei auch nicht vor knappen Anschlüssen zurück. Daher kann er insbesondere Nutzer unterstützen, die in einem sehr gut erschlossenen Gebiet mit vielen Fahrtalternativen unterwegs sind.

Die Information geschieht nicht anhand einer klassischen Fahrplanabfrage, sondern anhand von speziell für einzelne Nutzergruppen vorbereiteten Fragen, die präzise nach für auf dem Fahrtweg notwendig werdende Entscheidungen nötigen Informationen fragen. Die Antwort erfolgt als Ampel für eine schnelle Entscheidung und auch in ausführlicherer Textform für diejenigen, die sich lieber selbst ein Bild machen wollen.

Im Hinblick auf den Datenschutz besonders hervorzuheben ist die Datensparsamkeit des Dienstes: Durch die vordefinierten Nutzergruppen ist es durch ein Lesezeichen auf die entsprechende Seite möglich, ohne Registrierung eine spezifisch für den konkreten Nutzer passende Fragensauswahl zu präsentieren, die durch die manuelle Auswahl deutlich treffsicherer ist als die durch Algorithmen generierten Empfehlungen der Art „andere Nutzer fahren auch...“.

2 Realweltproblem

2.1 Erfahrungsbeschreibung

Meine Erfahrung als langjähriger ÖPNV-Nutzer in Heidelberg hat gezeigt, dass es insbesondere in Zentrumsnähe in Heidelberg einige Alternativrouten gibt, die unter Umständen ermöglichen, früher den Zielort zu erreichen, aber erfordern, die Fahrt in die intuitiv falsche Richtung zu beginnen oder eine Fahrt mit richtiger Richtung wieder zu verlassen. Einige Beispiele aus meiner Erfahrung sind:

- Fahrt vom Hauptbahnhof Richtung Bismarckplatz und umgekehrt: Alternative mit Umstieg am Betriebshof
- Fahrt von Hauptbahnhof Süd/Schwetzingen Terasse Richtung Seegarten/Bismarckplatz und umgekehrt: Alternative mit Umstieg an der Rudolf-Diesel-Straße in die Linie 26
- Fahrt von Hauptbahnhof Süd/Schwetzingen Terasse Richtung Bismarckplatz und umgekehrt: Alternative mit Linie 721 und Umstieg am Betriebshof
- Fahrt der Linie 33 Richtung Peterskirche mit Fahrtziel Universität: Umsteigemöglichkeit in die Linie 31/32 am Bismarckplatz

Auch sind an bestimmten Haltestellen wichtige Ziele in beiden Fahrtrichtungen erreichbar. Die Information, von welcher Seite man dieses am schnellsten erreicht, ist aber nicht ohne weiteres zu ermitteln (andere Fahrdauern, notwendiger Vergleich beider DFIs, ...):

- Fahrt vom Betriebshof Richtung Bismarckplatz: Fahrt über Hauptbahnhof oder direkt
- in Zukunft: Fahrt von der Haltestelle Gadamerplatz Richtung Bismarckplatz nach vollständiger Fertigstellung des entsprechenden Abschnitts des Mobilitätsnetzes: Linien 22 und 26 fahren in die entgegengesetzte Richtung, erreichen aber dasselbe Ziel mit derselben Fahrzeit

Diese Listen ließen sich mit anderen Nutzererfahrungen sicherlich noch um einige Punkte erweitern.

Die Alternativrouten zeichnen sich regelmäßig dadurch aus, dass sie nur funktionieren, wenn die Wartezeiten an den Umstiegshaltestellen knapp bemessen sind. Eventuell sind sie auch nur schneller, weil dadurch die Wartezeit an der Starthaltestelle durch frühere Abfahrt verkürzt sind. Zudem sind oft mehr Umstiege nötig als mit der umsteigefreisten Verbindung.

Aus verschiedenen Gründen versagen die klassischen Fahrplanauskunftssysteme bei der Entscheidungshilfe über die Wahl einer solchen Route:

- Fahrplanauskunftssysteme sind typischerweise darauf ausgerichtet, die schnellste Verbindung zu ermitteln. Dabei berücksichtigen sie aber für den Fall, dass der Fahrgast schon an der Haltestelle steht, nicht, dass er bei längerer Fahrtzeit, dafür aber deutlich früherer Abfahrt sein Ziel insgesamt schneller erreicht und sich dabei lästiges Warten an der Starthaltestelle erspart.
- Fahrplanauskunftssysteme sind dem Grunde nach immer noch auf Soll-Daten fixiert, planen also stets ausreichend Umsteigezeit ein, damit die Verbindung auch bei etwas Verspätung nicht gefährdet ist, sodass die knappen Anschlüsse, die die Alternativrouten erst attraktiv machen, dazu führen, dass die Alternativrouten nicht ermittelt werden. Wenn hingegen die Umsteigezeit zu knapp bemessen wird, besteht die Gefahr, dass die Alternativroute nicht wie vorgesehen funktioniert, dies aber erst bemerkt wird, wenn durch das Antreten der Alternativroute eine Umentscheidung nicht mehr möglich ist.

- Fahrplanauskunftssysteme haben, wie die ÖPNV-Planung, den Durchschnittsnutzer vor Augen, der nicht so gerne umsteigt, vermeiden daher tendenziell Umstiege. Auch ohne solche Voreinstellung neigen die Algorithmen dazu, den Weg auszuwählen, der mit weniger Rechenschritten erreichbar ist, also den mit weniger Umstiegen. Es gibt jedoch auch Nutzertypen, die lieber schnell ankommen und dafür Umstiege in Kauf nehmen oder auch sonstige Nutzer, die aus Zeitnot, etwa im Nachlauf eines verspäteten InterCity oder einer verspäteten S-Bahn, lieber umsteigen, um ihren Termin ggf. doch noch pünktlich zu erreichen.

Mit den offiziellen Tools lässt sich eine Information über diese Alternativrouten daher nur dadurch ermitteln, dass man die nach eigener Erfahrung möglichen Routen etwa in rnv Start.Info „ausprobiert“. Diese Art der Ermittlung ist aber aufwendig und führt dazu, dass ein ggf. gerade einfahrender Bus nicht mehr genommen werden kann.

2.2 Lösung durch dieses Projekt

Dieses Projekt versucht, dieses Informationsdefizit in zwei Stufen zu beheben, indem es genau dieses „ausprobieren“ für den Nutzer übernimmt und ihm nur das schnell erfassbare Ergebnis mitteilt.

2.2.1 Information an der Haltestelle

Als aufwendigste Lösung ist die Information über eine Art Ampel, die in der Nähe der Haltestelle angebracht ist, denkbar. Dieses Projekt liefert dafür eine Demonstrationsimplementierung, allerdings keine Hardwareumsetzung.

2.2.2 Information mit mobilen oder stationären Endgeräten

Die durch eine Haltestellenanzeige dargestellten Daten können selbstverständlich auch über das Internet ausgegeben werden, sodass auch mit eigenen mobilen oder stationären Endgeräten der Fahrgäste eine Information möglich ist. Allerdings kostet auch das Aufrufen einer Internetseite über den Browser oder der Abruf der Daten über eine spezielle App, ggf. sogar erst nach dem erforderlichen Start des Endgeräts, Zeit, die zum Verpassen des Busses führen kann, sodass diese Lösung im Vergleich zur Haltestelleninformation nur zweitbeste Wahl ist.

Gleichzeitig erlaubt diese Darstellungsweise aber, umfangreichere Informationen darzustellen. In Betracht kommt etwa eine Beantwortung von Fragen in der Art „Soll ich in die Linie 33 Richtung Emmertsgrund einsteigen, um schnell zum Bismarckplatz zu fahren?“ in natürlicher Sprache. Solche Fragen lassen sich dann auch ohne Aufwand in digitale Sprachassistenten wie Amazon Alexa integrieren. In der Webseitenversion können Details zur vorgeschlagenen Route ausgegeben werden.

Wenn man den Grundsatz der Datensparsamkeit ignorieren würde, ließe sich anhand von IP, Standortdaten oder vorherige Einstellung (per Login) eine weitere Auswahl der richtigen (=anzuzeigenden) Fragen durch das Programm realisieren.

2.2.3 Einschränkungen der vorliegenden Lösung

Da andernfalls eine vollständig neue Fahrplanermittlung erfolgen müsste, die nur auf Basis der Rohdaten, nicht aber auf Basis der vorliegenden Daten ohne nicht darstellbare Netzwerklast, erfolgen kann, basiert dieses Projekt auf einer Liste der Alternativrouten in einem standardisierten Format. Für die begrenzte Zahl an solchen für dieses Projekt geeigneten Routen innerhalb des VRN, die gehäuft nur in den Zentren mit guter ÖPNV-Abdeckung wie z. B. dem Kernbereich Heidelbergs auftreten, wäre ein anderes Vorgehen auch im Hinblick auf die Effizienz unsinnig.

Die vorliegende Lösung beschränkt sich beispielhaft auf die oben genannten Routen.

3 Projekthistorie

Eine erste Version eines Programmes, das solche Informationen im Frage-Antwort-Stil liefert, wurde im Rahmen meines nicht öffentlichen Informationsportals „HUVUG HD“ als „Mobilitätsfragen“ bereitgestellt. Das Portal HUVUG HD enthielt vorher schon eine der rnv-Start.Info-App nachgebildete Abfahrtstafelanzeige, um mir eine Nutzung auch vom PC und jedem anderen internetfähigen Gerät aus zu ermöglichen. Auf deren Implementierung basierte auch die obengenannte Version.

Für diesen Wettbewerb wurde auf dieser Idee aufbauend und auf die Datenbasis der VRN-API umgestellt eine neue Lösung entwickelt, die anders als die vorhergehende Implementierung durch ein objektorientiertes Einstellen der Routen verallgemeinerungsfähig ist und das vergleichsweise einfache Hinzufügen von neuen Routen erlaubt. Auch die Visualisierung durch die Ampel und die Ausgabe der Fahrtdetails wurde erstmals implementiert.

4 Detailumsetzung

Basis des Projekts ist ein PHP-Serverdienst, der HTML-Seiten zur Anzeige der Antworten und eine API für eine Hardwareampel bereitstellt.

Dieser Serverdienst wurde für die Demonstration auf einem Raspberry Pi installiert.

Darauf aufbauend wurde eine Beispielimplementierung für eine Hardwareampel entwickelt, für dessen Vervollständigung lediglich die entsprechenden Leuchtdioden an die bezeichneten GPIO-Pins des Rasperrys angeschlossen werden müssen.

Auf eine spezifische Anwendung für mobile Endgeräte wurde verzichtet, da eine solche keinen Vorteil im Vergleich zu einer einfachen Verknüpfung auf die nutzergruppenspezifische Fragenseite oder eine Antwort bietet. Das Layout der HTML-Seiten ist aber auch für mobile Endgeräte optimiert.

4.1 PHP-Serverdienst

In diesem Abschnitt werden die Dateien vorgestellt, die im Webseitenverzeichnis des Serverdienstes liegen müssen. Jeder Unterabschnitt beginnt mit einer kurzen Beschreibung der Funktionen der Dateien. Anschließend werden die Dateien in der Reihenfolge ihrer Abhängigkeiten, d. h. die grundlegendste Datei zuerst, im (größtenteils kommentierten) Volltext dargestellt.

4.1.1 Berechnung

Die Berechnung der Antworten erfolgt über das in `typen.php` angelegte Typensystem, welches dafür auf die von `trias.php` bereitgestellten Funktionen zur Übersetzung der TRIAS-API zurückgreift.

4.1.1.1 *trias.php*

```
1. <?php
2. /*Mittelfunktionen zur Übersetzung in Trias (dadurch andere API durch Austausch (nur
   ) dieser Datei möglich)*/
3. function HaltestelleIdToName($id){
4.     $triasmittel='         <LocationInformationRequest>
5.         <LocationRef>
6.         <StopPointRef>'. $id. '</StopPointRef>
7.         </LocationRef>
8.         </LocationInformationRequest>';
9.     $triasresult=@simplexml_load_string(gettrias($triasmittel));
```



```

10.     return $triasresult->ServiceDelivery->DeliveryPayload-
    >LocationInformationResponse->Location->Location->StopPoint->StopPointName->Text-
    >__toString();
11. }
12. function HaltestelleNameToId($name){
13.     $triasmittel='         <LocationInformationRequest>
14.         <InitialInput>
15.         <LocationName>' . $name . '</LocationName>
16.         </InitialInput>
17.         </LocationInformationRequest>';
18.     $triasresult=@simplexml_load_string(gettrias($triasmittel));
19.     return $triasresult->ServiceDelivery->DeliveryPayload-
    >LocationInformationResponse->Location->Location->StopPoint->StopPointRef-
    >__toString();
20. }
21. function HaltestelleAbfahrtafel($StopPointRef,$zeitpunkt="jetzt",$anab="ab",$folgeh
    alte="false"){
22.     if($anab=="ab"){ $anabxml="departure"; }
23.     if($anab=="an"){ $anabxml="arrival"; }
24.     if($zeitpunkt=="jetzt"){ $zeitpunkt=time(); }
25.     $zeitpunktxml=date(DateTime::RFC3339,$zeitpunkt);
26.     $triasmittel='         <StopEventRequest>
27.         <Location><LocationRef><StopPointRef>' . $StopPointRef . '</StopPointRef
    ></LocationRef><DepArrTime>' . $zeitpunktxml . '</DepArrTime></Location>
28.         <Params>
29.         <StopEventPolicy><TimeWindow>15H</TimeWindow></StopEventPolicy>
30.         <StopEventContentFilter><IncludeOnwardCalls>' . $folgehalte . '</Inc
    ludeOnwardCalls><IncludeRealtimeData>true</IncludeRealtimeData></StopEventContentFil
    ter>
31.         <StopEventPolicy><StopEventType>' . $anabxml . '</StopEventType></St
    opEventPolicy>
32.         </Params><!--
    Linienfilter funktioniert derzeit offensichtlich nicht--
    >keine Implementation des Linienfilters-->
33.         </StopEventRequest>';
34.     $triasresult=@simplexml_load_string(gettrias($triasmittel));
35.     $abfahrten=$triasresult->ServiceDelivery->DeliveryPayload-
    >StopEventResponse->StopEventResult;
36.     $abfahrtenarray=array();
37.     foreach($abfahrten as $abfahrt){
38.         $abfahrtevent=$abfahrt->StopEvent;
39.         $LineRef=$abfahrtevent->Service->LineRef->__toString();
40.         $JourneyRef=$abfahrtevent->Service->JourneyRef->__toString();
41.         $JourneyOperatingDay=$abfahrtevent->Service->OperatingDayRef-
    >__toString();
42.         $LineDirection=$abfahrtevent->Service->DirectionRef->__toString();
43.         $LineName=$abfahrtevent->Service->PublishedLineName->Text->__toString();
44.         $LineVon=$abfahrtevent->Service->OriginText->Text->__toString();
45.         $LineNach=$abfahrtevent->Service->DestinationText->Text->__toString();
46.         $LineTyp=$abfahrtevent->Service->Mode->Name->Text->__toString();
47.         $hasrealtime=true;
48.         if($anab=="ab"){
49.             $AbfahrtTimePlan=$abfahrtevent->ThisCall->CallAtStop->ServiceDeparture-
    >TimetabledTime->__toString();
50.             try{$AbfahrtTimeReal=$abfahrtevent->ThisCall->CallAtStop-
    >ServiceDeparture->EstimatedTime-
    >__toString();}catch(Exception $e){$AbfahrtTimeReal=""; $hasrealtime=false;}
51.         }
52.         if($anab=="an"){
53.             $AbfahrtTimePlan=$abfahrtevent->ThisCall->CallAtStop->ServiceArrival-
    >TimetabledTime->__toString();
54.             try{$AbfahrtTimeReal=$abfahrtevent->ThisCall->CallAtStop-
    >ServiceArrival->EstimatedTime-
    >__toString();}catch(Exception $e){$AbfahrtTimeReal=""; $hasrealtime=false;}
55.         }
56.         if($AbfahrtTimeReal==""){ $hasrealtime=false;}

```

```

57.     $diefolgehalte="nicht ermittelt";
58.     if($folgehalte=="true"){ $diefolgehalte=$abfahrtevent->OnwardCall;}
59.     $abfahrtenarray[]=array("LineRef"=>$LineRef, "JourneyRef"=>$JourneyRef, "JourneyOperatingDay"=>$JourneyOperatingDay, "LineDirection"=>$LineDirection, "LineName"=>$LineName, "LineVon"=>$LineVon, "LineNach"=>$LineNach, "LineTyp"=>$LineTyp, "hasrealtime"=>$hasrealtime, "AbfahrtTimePlan"=>$AbfahrtTimePlan, "AbfahrtTimeReal"=>$AbfahrtTimeReal, "Folgehalte"=>$diefolgehalte);
60.     }
61.     /*Filterung nach Abfahrten erst nach der angefragten Zeit; da Zeitfilter der API wieder funktioniert unnötig, aber unschädlich und daher als zusätzliche Kontrolle sinnvoll; funktioniert auch ohne API-Filter solange, wie genügend Antworten*/
62.     $abfahrtenarrayTEMP=$abfahrtenarray;
63.     for($i=0;$i<count($abfahrtenarray);$i++){
64.         if($abfahrtenarray[$i]["hasrealtime"]){if(strtotime($abfahrtenarray[$i]["AbfahrtTimeReal"])<$zeitpunkt){unset($abfahrtenarrayTEMP[$i]);}}
65.         else{if(strtotime($abfahrtenarray[$i]["AbfahrtTimePlan"])<$zeitpunkt){unset($abfahrtenarrayTEMP[$i]);}}
66.     }
67.     $abfahrtenarray=$abfahrtenarrayTEMP;
68.
69.     /*weiter regulär*/
70.     return $abfahrtenarray;
71. }
72. function Fahrtverlauf($JourneyRef,$OperatingDayRef){
73.     $triasmittel='          <TripInfoRequest>
74.         <JourneyRef>'. $JourneyRef. '</JourneyRef>
75.         <OperatingDayRef>'. $OperatingDayRef. '</OperatingDayRef>
76.     </TripInfoRequest>';
77.     echo $triasmittel;
78.     $triasresult=@simplexml_load_string(gettrias($triasmittel));
79.     /*TODO: Antwort, derzeit nicht benötigt, da API mit der dokumentierten Anfrage ohnehin nicht funktioniert und daher Ermittlung über Abfahrts-
80.     API (Folgehalte)*/
81. }
82. function checkMyWay($starttime,$starthalt,$myway){
83.     if($starttime=="jetzt"){ $starttime=time();}
84.     $result=array("status"=>"+", "abfahrtszeit"=>0, "ankunftszeit"=>0, "fahrtminuten"=>0, "gesamtminuten"=>"fahrt+warte", "warteminuten"=>array(), "fahrtweg"=>array()); /*Status: ?=unbekannt, da mindestens eine Fahrt ohne aktuelle Abfahrtszeit; +=passt, !=knapper Anschluss (1min), !=sehr knapper Anschluss (0min), -=Anschluss wegen fehlendem Halt an Umsteigestation nicht möglich*/
85.     $ende=false;
86.     $nextway=$myway;
87.     $nextstarthalt=$starthalt;
88.     $nextstarttime=$starttime;
89.     $i=1;/*Nummer der Haltestelle in Schlange (wichtig für Ermittlung 1. Haltestelle)*/
90.     while($ende!=true){
91.         if($nextway->linie=="Fuß"){
92.             $result["fahrtweg"][]="Fußweg ". $nextway->fußminuten. " Minuten";
93.             $result["fahrtminuten"]+=$nextway->fußminuten;
94.             $result["ankunftszeit"]+=$nextway->fußminuten*60;
95.         }
96.         else{
97.             $abfahrtszeit=$nextstarttime;
98.             $nextabfahrten = $nextstarthalt->getAbfahrtsTafel($nextway->linie,$nextstarttime,"ab","false");/*letzte Option auf "false" wenn Ermittlung über Fahrtverlauf-API, auf True wenn dies weiterhin nicht funktioniert aber dafür Haltestellen-API auch Echtzeitdaten für Folgedaten herausrückt (dann unten entsprechende Anpassung der Alternative 3 nötig)*/
99.             if($i==1){$result["fahrtweg"][]="Fahrtbeginn: ". $nextstarthalt->getName();}
100.            $myabfahrt=$nextabfahrten[0];/*nächster Bus ab der nächsten Haltestelle*/
101.            /

```

```

100.         if($myabfahrt["hasrealtime"]){$warteminuten=(strtotime($myabfahrt["Abfahr
rtTimeReal"])-
$nextstarttime)/60;$abfahrtszeit=strtotime($myabfahrt["AbfahrtTimeReal"]);}
101.         else{$warteminuten=(strtotime($myabfahrt["AbfahrtTimePlan"])-
$nextstarttime)/60; $abfahrtszeit=strtotime($myabfahrt["AbfahrtTimePlan"]); $result[
"status"]="?";/*keine Echtzeitdaten*/}
102.         $result["warteminuten"][]=$warteminuten;
103.         if($i==1){/*Wartezeitbehandlung für erstes Verkehrsmittel*/
104.             $result["abfahrtszeit"]=$abfahrtszeit;
105.             //if($warteminuten>5&&$i>0){$result["status"]="lwz";}/*Wartezeit für
ersten Bus wird nicht gezählt*/
106.         }
107.         else{/*Wartezeitbehandlung für Umstiege*/
108.             if($warteminuten==1){$result["status"]="!";}
109.             if($warteminuten==0){$result["status"]="!!";}
110.             $result["fahrtweg"][]="Umsteigezeit ".$warteminuten." Minuten";
111.         }
112.         /*Ankunftszeit ermitteln*/
113.         /* Alternative 1: Wenn Fahrtverlauf-
API funktioniert: Ermittlung des Fahrtverlaufs durch Fahrtverlauf($myabfahrt["Journe
yRef"],$myabfahrt["JourneyOperatingDay"]); dann TODO: Schreiben der Ankunftszeit an
Zielhaltestelle in $result["ankunftszeit"] (spart Netzwerktraffic und wohl auch Rech
ezeit der API, dafür sollte oben bei Abfrage der Abfahrtszeit die Option $folgehalt
e auf "false" gesetzt werden)*/
114.         if($myabfahrt["Folgehalte"]=="nicht ermittelt"){
115.             /* Alternative 2: HOTFIX: Ermittle aus Abfahrtszeit der Zielhaltestelle
(funktioniert nicht bei langen Fahrtzeiten, da Ankunftszeit ja noch nicht bekannt)*
/
116.             $zielankunft=$nextway->zielhalt-
>getAnkunftszeit(array(),strtotime($myabfahrt["AbfahrtTimePlan"]));
117.             $result["ankunftszeit"]=0;
118.             foreach($zielankunft as $ankunft){
119.                 if($ankunft["JourneyRef"]===$myabfahrt["JourneyRef"]){/*Fahrt sti
mmt überein*/
120.                     if($ankunft["hasrealtime"]){$result["ankunftszeit"]=strtotim
e($ankunft["AbfahrtTimeReal"]);/*Ankunftszeit wird mit Zwischenzeiten belegt und hat
daher am Ende den Wert der Zielankunft*/}
121.                     else{$result["ankunftszeit"]=strtotime($ankunft["AbfahrtTime
Plan"]);/*keine Echtzeitdaten $result["status"]="?";*/}
122.                 }
123.             }
124.             /* Alternative 2 Ende */
125.         }
126.         else{
127.             /* Alternative 3: Ermittlung aus Folgehaltestellen (derzeit nicht in Bet
rieb (vgl. Parameter false bei Abfahrtszeit), da dort keine Echtzeitdaten) */
128.             foreach($myabfahrt["Folgehalte"] as $myfolgehalt){
129.                 $haltref=$myfolgehalt->CallAtSop->StopPointRef-
>__toString(); /*CallAtSop nach Dokumentation Tippfehler für CallAtStop --
> wenn es nicht mehr funktioniert liegt es wohl daran*/
130.                 if(str_replace($nextway->zielhalt-
>getId(),"",$haltref)!=$haltref){/*$haltref enthält Zielhaltestellen-
ID*/$result["ankunftszeit"]=strtotime($myfolgehalt->CallAtSop->ServiceArrival-
>TimetabledTime);$result["status"]="?";/*keine Echtzeitdaten */}
131.             }
132.         }
133.         /* Alternative 3 Ende */
134.     }
135.     /* Ende Ermittlung Ankunftszeit */
136.     $result["fahrtweg"][]=date("H:i",$abfahrtszeit). " Uhr: Einsteigen in ".$
myabfahrt["LineTyp"]. " ".$myabfahrt["LineName"]. " Richtung ".$myabfahrt["LineNach"];

137.     if($result["ankunftszeit"]==0){
138.         $result["status"]="-" /*kein Halt an Umsteigestation*/
139.         $result["ankunftszeit"]="nie";
140.         $result["fahrtweg"][]="Ziel wird nicht in angemessener Zeit erreicht
.";

```

```

141.         $ende=true; /*Schleife beenden*/
142.     }
143.     else{
144.         $result["fahrtweg"][]=date("H:i", $result["ankunftszeit"]). " Uhr: Aus
steigen an ".$nextway->zielhalt->getName();
145.         $result["fahrtminuten"+($result["ankunftszeit"]-
$abfahrtszeit)/60;
146.     }
147. }
148. if($result["status"]=="?"){ $result["fahrtweg"][]="Die Aussagen zum unmittelb
ar vorstehenden Verkehrsmittel beruhen nicht auf Echtzeitinformationen.";}
149. $nextstarttime=$result["ankunftszeit"];
150. $nextstarthalt=$nextway->zielhalt;
151. $nextway=$nextway->weiter;
152. if($nextway==array()){ $ende=true;}
153. else{ $nextway=$nextway[0]; /*in derzeitiger Umsetzung prüft CheckMyWay nur er
sten Eintrag (andere Prüfung komplizierter, weil durch Multiweiterfahrtmöglichkeiten
Prüfung nur rekursiv möglich und deutlich steigende API-Abfragen --
> langsamer)*/}
154. $i++;
155. }
156. $warteminutengesamt=0; foreach($result["warteminuten"] as $minuten){ $warteminute
ngesamt+=$minuten;}
157. $result["gesamtminuten"]=$result["fahrtminuten"]+$warteminutengesamt;
158. return $result;
159.}
160./*eigentliche Abfrage*/
161.function gettrias($triasmittel){
162.$triasbeginn='<?xml version="1.0" encoding="iso-8859-1" ?>
163.<Trias xmlns="trias" xmlns:siri="http://www.siri.org.uk/siri" version="1.0">
164.    <ServiceRequest>
165.        <siri:RequestTimestamp>'.date("Y-m-d\TH:i:s").'</siri:RequestTimestamp>
166.        <siri:RequestorRef>TRIOD</siri:RequestorRef>
167.        <RequestPayload>
168.';
169.$triasende='
170.        </RequestPayload>
171.    </ServiceRequest>
172.</Trias>';
173.
174.$url = 'https://triod.vrn.de/Middleware/Data/trias';
175.
176.$ch = curl_init($url);
177	curl_setopt($ch, CURLOPT_POST, 1);
178	curl_setopt($ch, CURLOPT_VERBOSE, 1);
179	curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 0);
180	curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, 0);
181	curl_setopt($ch, CURLOPT_POSTFIELDS, $triasbeginn.$triasmittel.$triasende);
182	curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type: application/xml'));
183	curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
184.$response = curl_exec($ch);
185.if(curl_errno($ch)){ die(curl_errno($ch).curl_error($ch));}
186	curl_close($ch);
187.return $response;
188.}
189.??>

```

4.1.1.2 typen.php

```

1. <?php
2. /*Frage und Antwort*/
3. class Mobilitaetsfrage{
4.     public $title=""; /*angezeigte Frage*/
5.     public $category=""; /*Nutzergruppe*/
6.     public $art=""; /*Art der Ampel: Ja/Nein (verbindungswahl) oder Richtung A/B (ric
htungswahl) oder einfacher Text (nurtext) oder Wartezeitampel (wartezeit) oder Frage
, ob sich Umstieg lohnt (umstieg)*/
7.     public $starthalt=null; /*Klasse vom Typ Haltestelle*/

```

```

8.     public $fahrtA=null;/*Weiterfahrtmöglichkeit -
       Alternative A (bei Verbindungswahl: normale Fahrt)*/
9.     public $fahrtB=null;/*Weiterfahrtmöglichkeit -
       Alternative B (bei Verbindungswahl: Fahrtalternative, nach der gefragt wird)*/
10.    public $wartezeitinklStarthalt=false; /*Gibt an, ob bei Art wartezeit die Wartezeit an der Starthaltestelle berücksichtigt wird */
11.    public $prognosegrenze=600; /*Prognosegrenze (für Art umstieg) in Sekunden -
       > 600=60*10=10 Minuten*/
12.    private $titleFahrtA=null; /*Ampellichtbeschriftung, wird über entsprechende Methoden get/setFahrtenTitles (s. u.) abgerufen/festgelegt, da Standardwerte abhängig von $art*/
13.    private $titleFahrtB=null; /*Ampellichtbeschriftung, wird über entsprechende Methoden get/setFahrtenTitles (s. u.) abgerufen/festgelegt, da Standardwerte abhängig von $art*/
14.    private $titleFahrtUnentschieden=null; /*Ampellichtbeschriftung, wird über entsprechende Methoden get/setFahrtenTitles (s. u.) abgerufen/festgelegt, da Standardwerte abhängig von $art*/
15.    private $antworttextFahrtA=null; /*Antworttext, wird über entsprechende Methoden get/setAntwortText (s. u.) abgerufen/festgelegt, da Standardwerte abhängig von $art*/
16.    private $antworttextFahrtB=null; /*Antworttext, wird über entsprechende Methoden get/setAntwortText (s. u.) abgerufen/festgelegt, da Standardwerte abhängig von $art*/
17.    private $antworttextFahrtUnentschieden=null; /*Antworttext, wird über entsprechende Methoden get/setAntwortText (s. u.) abgerufen/festgelegt, da Standardwerte abhängig von $art*/
18.    public $antworttextadd=null;
19.    function __construct($name,$category,$art,$starthaltid) {
20.        $this->title=$name;
21.        $this->category=$category;
22.        $this->art=$art;
23.        $this->starthalt=new Haltestelle($starthaltid);
24.        $this->fahrtA=new Weiterfahrtmoeglichkeit();
25.        $this->fahrtB=new Weiterfahrtmoeglichkeit();
26.    }
27.    function getAntwort() {
28.        $thereturn = new Mobilitaetsantwort();
29.        $thereturn->ampellicht=2;
30.        $thereturn->text="textbasierte Antwort";
31.        $thereturn->isEchtzeit=false;
32.        $thereturn->fahrtADetails=array();
33.        $thereturn->fahrtBDetails=array();
34.        if($this->art=="nurtext"){
35.            $thereturn->text=$this->antworttextadd;
36.        }
37.        elseif($this->art=="umsteigezeit"){
38.            $fahrtA=checkMyWay("jetzt",$this->starthalt,$this->fahrtA);
39.            $minstowait=0;
40.            $istart=1;
41.            if($this->wartezeitinklStarthalt){$istart=0;}
42.            for($i=$istart;$i<count($fahrtA["warteminuten"]);$i++){
43.                $minstowait+=$fahrtA["warteminuten"][$i];
44.            }
45.            if($fahrtA["status"]=="-"){
46.                $thereturn->ampellicht=0;$thereturn->text="Die Fahrt erreicht ihr Ziel nicht in angemessener Zeit. Warten lohnt sich nicht.";}
47.            else{
48.                if($minstowait<5){$thereturn->ampellicht=3;}
49.                if($minstowait>=5&&$minstowait<=10){$thereturn->ampellicht=2;}
50.                if($minstowait>10){$thereturn->ampellicht=1;}
51.                $wartezeitstarttext="";
52.                if($this->wartezeitinklStarthalt){$wartezeitstarttext=" (inkl. der Wartezeit an der Starthaltestelle).";}
53.                $thereturn->text="Sie müssen insgesamt ".$wartezeitstarttext." ".(int)$minstowait." Minuten warten.";}

```

```

53.     }
54.     $thereturn->fahrtADetails=$fahrtA["fahrtweg"];
55.     $thereturn->fahrtBDetails=$fahrtA["fahrtweg"];
56.     if($fahrtA["status"]!="?"){ $thereturn->isEchtzeit=true;}
57.     if(isset($this->antworttextadd)){ $thereturn->text.="\\n".$this-
>antworttextadd;}
58.     }
59.     else{
60.         $fahrtA=checkMyWay("jetzt",$this->starthalt,$this->fahrtA);
61.         if($this->art=="umstieg"){
62.             $ankunft=$this->starthalt->getAnkunftszeit($this->fahrtA->linie);
63.             $ankunftszeitFahrtA=strtotime($ankunft[0]["AbfahrtTimePlan"]);
64.             if($hasrealtime){ $ankunftszeitFahrtA=strtotime($ankunft[0]["Abfahrt
imeReal"]);}
65.             if($ankunftszeitFahrtA-time()>$this->prognosegrenze){
66.                 $thereturn->ampellicht=0;
67.                 $thereturn-
>text="Weiß ich nicht.\\nIhre Fahrt kommt erst um ".date("H:i",$ankunftszeitFahrtA)."
Uhr an der Umsteigehaltestelle an.\\nIch kann erst eine solide Prognose erstellen, w
enn Ihre Fahrt dort in weniger als ".($this-
>prognosegrenze/60)." Minuten ankommt.";
68.                 $thereturn->fahrtADetails=$fahrtA["fahrtweg"];
69.                 $thereturn-
>fahrtBDetails=array("Details zur Fahrtalternative wurden aus oben genanntem Grund n
icht ermittelt.");
70.                 if($fahrtA["status"]!="?"){ $thereturn-
>isEchtzeit=true; /*da nur eine Verbindung geprüft, muss auch nur diese Echtzeit sein
, um Warnmeldung auszublenden, dass einer geprüfte Verbindung keine Echtzeitdaten zu
grunde liegen*/}
71.                 if(isset($this->antworttextadd)){ $thereturn->text.="\\n".$this-
>antworttextadd;}
72.                 return $thereturn; /*keine Ausführung der folgenden Ampellicht-
und Textermittlungen, sondern sofortiger Abbruch*/
73.             }
74.             else{
75.                 $fahrtB=checkMyWay($ankunftszeitFahrtA,$this->starthalt,$this-
>fahrtB); /* Suche Verbindung B nicht vor Ankunft von Verbindung A (dort sitzt der F
ahrgast)*/
76.             }
77.         }
78.         else{
79.             $fahrtB=checkMyWay("jetzt",$this->starthalt,$this->fahrtB);
80.         }
81.         $differenzminuten=( $fahrtA["ankunftszeit"]-
$fahrtB["ankunftszeit"])/60;
82.         $schnellereFahrtA=true;
83.         $thereturn->ampellicht=1; /*schnellere Fahrt A*/
84.         $anschlusskennzeichen=$fahrtA["status"]; /*schnellere Fahrt A maßgeblich
für Anschlusskennzeichen*/
85.         if($differenzminuten>0){ /*Subtrahend (=Ankunftszeit FahrtB) kleiner (=fr
üher) als Minuend (=Ankunftszeit FahrtA) --> schnellereFahrtB*/
86.             $schnellereFahrtA=false;
87.             $thereturn->ampellicht=3; /*schnellere Fahrt B*/
88.             $anschlusskennzeichen=$fahrtB["status"]; /*schnellere Fahrt B maßgebli
ch für Anschlusskennzeichen*/
89.         }
90.         if($differenzminuten==0){ $thereturn->ampellicht=2; /*unentschieden*/}
91.         if($fahrtA["status"]=="-"){ $thereturn-
>ampellicht=3; $schnellereFahrtA=false; $differenzminuten=0; $anschlusskennzeichen="-
";}
92.         if($fahrtB["status"]=="-"){ $thereturn-
>ampellicht=1; $schnellereFahrtA=true; $differenzminuten=0; $anschlusskennzeichen="-
";}
93.         if($fahrtA["status"]=="-"&&$fahrtB["status"]=="-"){
94.             $thereturn-
>ampellicht=0; /*Ampel aus, da keine Aussage (keine Fahrt erreicht ihr Ziel)*/

```

```

95.         if($this->art=="richtungswahl"){$thereturn-
>ampellicht=2;/*bei Richtungswahl existiert Licht "unklar", welches auch genutzt wer-
den soll*/}
96.     }
97.     if($anschlusskennzeichen=="!"||$anschlusskennzeichen=="!!"){$thereturn-
>ampellicht=2;}
98.     $thereturn->text=$this->getAntwortText($thereturn-
>ampellicht,$schnellereFahrtA,$differenzminuten,$anschlusskennzeichen);
99.     if($fahrtA["status"]!="?"&&$fahrtB["status"]!="?"){ $thereturn-
>isEchtzeit=true;}
100.     $thereturn->fahrtADetails=$fahrtA["fahrtweg"];
101.     $thereturn->fahrtBDetails=$fahrtB["fahrtweg"];
102.     }
103.     return $thereturn;
104. }
105. function getFahrtenTitles(){
106.     $thereturn = array("fahrtA"=>"nein","fahrtB"=>"ja","fahrtUnentschieden"=>"vi-
elleicht");
107.     if($this-
>art=="verbindungswahl"){ $thereturn = array("fahrtA"=>"nein","fahrtB"=>"ja","fahrtUn-
entschieden"=>"vielleicht");}
108.     if($this-
>art=="richtungswahl"){ $thereturn = array("fahrtA"=>"Richtung A","fahrtB"=>"Richtung
B","fahrtUnentschieden"=>"unklar");}
109.     if($this-
>art=="umsteigezeit"){ $thereturn = array("fahrtA"=>"mehr als 10 Minuten","fahrtB"=>"
weniger als 5 Minuten","fahrtUnentschieden"=>"5-10 Minuten");}
110.     if(isset($this->titleFahrtA)&&isset($this->titleFahrtB)&&isset($this-
>titleFahrtUnentschieden)){ $thereturn = array("fahrtA"=>$this-
>titleFahrtA,"fahrtB"=>$this->titleFahrtB,"fahrtUnentschieden"=>$this-
>titleFahrtUnentschieden);}
111.     return $thereturn;
112. }
113. function setFahrtenTitles($fahrtA,$fahrtB,$fahrtUnentschieden){
114.     $this->titleFahrtA=$fahrtA;
115.     $this->titleFahrtB=$fahrtB;
116.     $this->titleFahrtUnentschieden=$fahrtUnentschieden;
117. }
118. function getAntwortText($ampellicht,$schnellereFahrtA,$differenzminuten,$anschl-
sskennzeichen){
119.     $thereturn = "Leider ist keine textbasierte Antwort verfügbar.";
120.     $fahrtenTitles = $this->getFahrtenTitles();
121.     if($this->art=="verbindungswahl"||$this->art=="umstieg"){
122.         if($ampellicht==1){ $thereturn = "Nein, diese Alternative ist ".-
$differenzminuten." Minuten langsamer.";}
123.         if($ampellicht==2){
124.             $thereturn = "Die Alternativen unterscheiden sich nicht.";
125.             if(!$schnellereFahrtA){/*Fahrtalternative ist schneller*/
126.                 if($anschlusskennzeichen=="!"){ $thereturn="Die erfragte Fahrtal-
ternative enthält einen knappen Anschluss (1 Minute).\nDas Erreichen dieses Anschluss-
es kann bei weiteren Verspätungen nicht garantiert werden.\nIm Erfolgsfall wäre die
Fahrtalternative ".abs($differenzminuten)." Minuten schneller.";}
127.                 if($anschlusskennzeichen=="!!"){ $thereturn="Die erfragte Fahrtal-
ternative enthält einen sehr knappen Anschluss (0 Minuten).\nDas Erreichen dieses An-
schlusses kann nicht garantiert werden.\nSie sollten sich gegen die Fahrtalternative
entscheiden.\nIm Erfolgsfall wäre die Fahrtalternative ".abs($differenzminuten)." M-
inuten schneller.";}
128.             }
129.             else{/*normale Fahrt ist schneller*/
130.                 if($anschlusskennzeichen=="!"){ $thereturn="Die normale Fahrt ent-
hält einen knappen Anschluss (1 Minute).\nDas Erreichen dieses Anschlusses kann bei
weiteren Verspätungen nicht garantiert werden.\nIm Erfolgsfall wäre die normale Fahr-
t ".abs($differenzminuten)." Minuten schneller.";}
131.                 if($anschlusskennzeichen=="!!"){ $thereturn="Die normale Fahrt en-
thält einen sehr knappen Anschluss (0 Minuten).\nDas Erreichen dieses Anschlusses ka-
nn nicht garantiert werden.\nSie sollten sich für die Fahrtalternative entscheiden.\n

```

```

nIm Erfolgsfall wäre die normale Fahrt ".abs($differenzminuten)." Minuten schneller.
";}
132.         }
133.     }
134.     if($ampellicht==3){$thereturn = "Ja, diese Alternative ist ".abs($differenzminuten)." Minuten schneller.";}
135.     if($anschlusskennzeichen=="-
"){/*eine Fahrt (wenn schnellereFahrtA=false Fahrt A, sonst Fahrt B) erreicht ihr Ziel nicht*/
136.         if($schnellereFahrtA){$thereturn="Nein, die Alternative erreicht ihr Ziel nicht in angemessener Zeit.";}
137.         else{$thereturn="Ja, nur die Alternative erreicht ihr Ziel.";}
138.         if($ampellicht==0){/*keine Fahrt erreicht ihr Ziel*/$thereturn="In nächster Zeit fahren keine passenden Verkehrsmittel auf den gewünschten Strecken.";}
139.     }
140. }
141. if($this->art=="richtungswahl"){
142.     if($ampellicht==1){$thereturn = "Die Fahrt ".$fahrtenTitles["fahrtA"]." ist ".abs($differenzminuten)." Minuten schneller.";}
143.     if($ampellicht==2){
144.         $thereturn = "Die Alternativen unterscheiden sich nicht.";
145.         $schnellerefahrtTitle=$fahrtenTitles["fahrtA"];
146.         if(!$schnellereFahrtA){$schnellerefahrtTitle=$fahrtenTitles["fahrtB"]};
147.         if($anschlusskennzeichen=="!"){$thereturn="Die Fahrt ".$schnellerefahrtTitle." enthält einen knappen Anschluss (1 Minute).\nDas Erreichen dieses Anschlusses kann bei weiteren Verspätungen nicht garantiert werden.\nIm Erfolgsfall wäre die Fahrtalternative ".abs($differenzminuten)." Minuten schneller.";}
148.         if($anschlusskennzeichen=="!!"){$thereturn="Die Fahrt ".$schnellerefahrtTitle." enthält einen sehr knappen Anschluss (0 Minuten).\nDas Erreichen dieses Anschlusses kann nicht garantiert werden.\nSie sollten sich gegen die Fahrtalternative entscheiden.\nIm Erfolgsfall wäre die Fahrtalternative ".abs($differenzminuten)." Minuten schneller.";}
149.     }
150.     if($ampellicht==3){$thereturn = "Die Fahrt ".$fahrtenTitles["fahrtB"]." ist ".abs($differenzminuten)." Minuten schneller.";}
151.     if($anschlusskennzeichen=="-
"){/*eine Fahrt (wenn schnellereFahrtA=false Fahrt A, sonst Fahrt B) erreicht ihr Ziel nicht*/
152.         if($schnellereFahrtA){$thereturn="Die Fahrt ".$fahrtenTitles["fahrtB"]." erreicht ihr Ziel nicht in angemessener Zeit.";}
153.         else{$thereturn="Die Fahrt ".$fahrtenTitles["fahrtA"]." erreicht ihr Ziel nicht.";}
154.         if($ampellicht==2){/*keine Fahrt erreicht ihr Ziel*/$thereturn="In nächster Zeit fahren keine passenden Verkehrsmittel auf den gewünschten Strecken.";}
155.     }
156. }
157. if(isset($this->antworttextadd)){$thereturn.="\\n".$this->antworttextadd;}
158. return $thereturn;
159. }
160.}
161.
162.class Weiterfahrtmoeglichkeit{
163.     public $linie=null;/*erlaubte Linien als array (wenn leer: alle)*/
164.     public $zielhalt=null;/*Klasse vom Typ Haltestelle*/
165.     public $weiter=null;/*alle Weiterfahrtmöglichkeiten als array (wenn leer: Ziel erreicht) [ACHTUNG: in derzeitiger Umsetzung prüft CheckMyWay nur ersten Eintrag (andere Prüfung komplizierter, weil durch Multiweiterfahrtmöglichkeiten Prüfung nur rekursiv möglich und deutlich steigende API-Abfragen --> langsamer)*/
166.     public $fußminuten=0;/*wenn Linie=Fuß steht hier zu veranschlagende Zeit in Minuten*/
167.     function __construct($zielhaltid="", $linie=array(), $weiter=array()) {
168.         $this->zielhalt=new Haltestelle($zielhaltid);
169.         $this->weiter=$weiter;
170.         $this->linie=$linie;

```



```

171.     }
172. }
173.
174. class Mobilitaetsantwort{
175.     public $ampellicht=0;
176.     public $text="Es ist keine textbasierte Antwort verfügbar.";
177.     public $isEchtzeit=false;
178.     public $fahrtADetails=array();
179.     public $fahrtBDetails=array();
180. }
181.
182. /*Klassen, die Verkehrsobjekt speichern (Vorbereitung für API-Abfrage)*/
183. require_once("trias.php");
184. class Haltestelle{
185.     private $name=null;
186.     private $StopPointRef="";
187.     function __construct($id="", $name=null) {
188.         $this->StopPointRef=$id;
189.         $this->name=$name;
190.     }
191.     function getName(){
192.         if(isset($name)){/*wenn Name schon ermittelt wurde, braucht Trias nicht erne
193. ut abgefragt zu werden*/return $this->name;}
194.         else{/*Name wird erst on demand ermittelt, um unnötige Abfragen zu sparen*/
195.
196.             $name=HaltestelleIdToName($this->StopPointRef);
197.             $this->name=$name;
198.             return $name;
199.         }
200.     }
201.     function getAbfahrtafel($filter=array(), $starttime="jetzt", $anab="ab", $folgehalte="false"){
202.         $tafelresult=HaltestelleAbfahrtafel($this->StopPointRef, $starttime, $anab, $folgehalte);
203.         $filteredtafel=array();
204.         if($filter!=array()){
205.             foreach($tafelresult as $result){
206.                 $aufnahme=false;
207.                 foreach($filter as $thefilter){
208.                     if($result["LineRef"]===$thefilter->id){/*nur angegebene Linien*/
209.                         if($thefilter->directionart=="haltestellevon"&&str_replace($thefilter->direction, "", $result["LineVon"])!=$result["LineVon"]){$aufnahme=true;}
210.                         if($thefilter->directionart=="haltestellenach"&&str_replace($thefilter->direction, "", $result["LineNach"])!=$result["LineNach"]){$aufnahme=true;}
211.                         if($thefilter->directionart=="richtung"&&$result["LineDirection"]===$thefilter->direction){$aufnahme=true;}
212.                         if($thefilter->direction==""){$aufnahme=true;}
213.                     }
214.                 }
215.                 if($aufnahme){$filteredtafel[]=$result;}
216.             }
217.         }
218.         else{$filteredtafel=$tafelresult;}
219.         return $filteredtafel;
220.     }
221.     function getAnkunfthtafel($filter=array(), $starttime="jetzt"){
222.         return $this->getAbfahrtafel($filter, $starttime, "an", "false");
223.     }
224.     function getId(){
225.         return $this->StopPointRef;
226.     }
227.     function NewFromName($name="") {/*einfacher durchzuführen über direkten Aufruf H
228. altestelleNameToId direkt in Argument bei neuer Klasse*/

```

```

226.     $id=HaltestelleNameToId($name);
227.     $this->name=$name;
228.     $this->StopPointRef=$id;
229. }
230.     function NewFromId($id="") {/*einfacher durchzuführen über direkten Aufruf Halte-
stelleNameToId direkt in Argument bei neuer Klasse*/
231.         $this->name=null;
232.         $this->StopPointRef=$id;
233.     }
234.}
235.
236.class Linie{
237.     public $title="";/*wird nicht festgelegt, da Abfrage Liniennamen-
>ID nicht in Trias vorgesehen*/
238.     public $id="";
239.     public $direction=""; /*leerer String, wenn alle*/
240.     public $directionart="";/*$directionart: haltestellenach (Stringteil aus Endhalte-
stelle), haltestellevon (Stringteil aus Endhaltestelle) oder richtung (wie in TRIAS
-Rückgabe)*/
241.     function __construct($id,$direction="", $directionart="haltestellenach") {
242.         $this->id=$id;
243.         $this->direction=$direction;
244.         $this->directionart=$directionart;
245.     }
246.}
247.??>

```

4.1.2 Eingabe

Das Anlegen der Objekte zu den einzelnen Fragen erfolgt in der Datei routen.php unter Rückgriff auf das oben dargestellte Typensystem. Diese Datei ist daher das inhaltliche Herzstück des Projekts.

Die Datei triastest.php dient als Hilfsmittel für die Eingabe der Ermittlung der nötigen Richtungs- und Linienidentifizierer, indem er die vollständige Ausgabe der Abfahrtstafel als XML ermöglicht.

4.1.2.1 routen.php

```

1. <?php
2. require_once("typen.php");
3.
4. /*IDs häufig genutzter Haltestellen zum Einsparen von API-
Anfragen und Linien zur einfacheren Verwendungen*/
5. $haltestellen=array("Bismarckplatz"=>"de:08221:1146", "Rudolf-Diesel-
Straße"=>"de:08221:1149", "Schwetzinger Terasse"=>"de:08221:4256", "HD Hbf"=>"de:08221:
1160", "HD Hbf Süd"=>"de:08221:4272", "Uniplatz"=>"de:08221:1198", "Peterskirche"=>"de:
08221:1181", "Seergarten"=>"de:08221:1168", "Betriebshof"=>"de:08221:1144", "Marstallst
r."=>"de:08221:1174", "HD-
Altstadt"=>"de:08221:1166", "Mannheim Schloss"=>"de:08222:2462", "MA Hbf"=>"de:08222:2
417");
6. $linien=array("31"=>"de:vrn:19031", "32"=>"de:vrn:19032", "33"=>"de:vrn:19033", "34"=>"
de:vrn:19034", "35"=>"de:vrn:19035", "21"=>"de:vrn:11021", "22"=>"de:vrn:11022", "23"=>"
de:vrn:11022", "24"=>"de:vrn:11024", "26"=>"de:vrn:11026", "721"=>"de:vrn:33721", "M3"=>"
de:vrn:19013", "5"=>"de:vrn:13065", "9"=>"de:vrn:08009", "5A"=>"de:vrn:13005", "15"=>"d
e:vrn:13015", "1"=>"de:vrn:08001");
7. $WeiterfahrtBetriebshofNachBismarckplatz=new Weiterfahrtmoeglichkeit($haltestellen["
Bismarckplatz"],array(new Linie($linien["22"],"return","richtung"),new Linie($linien
["35"],"outward","richtung"));
8. $LinienHauptbahnhofNachBismarckplatz=array(new Linie($linien["5"],"outward","richtun
g"),new Linie($linien["9"],"return","richtung"),new Linie($linien["21"],"return","ri
chtung"),new Linie($linien["32"],"outward","richtung"),new Linie($linien["33"],"outw
ard","richtung"),new Linie($linien["34"],"outward","richtung"));
9. $LinienHauptbahnhofNachBetriebshof=array(new Linie($linien["21"],"outward","richtung
"),new Linie($linien["24"],"return","richtung"),new Linie($linien["32"],"return","ri
chtung"),new Linie($linien["34"],"return","richtung"));
10. $LinienBetriebshofNachHauptbahnhof=array(new Linie($linien["21"],"return","richtung"
),new Linie($linien["24"],"outward","richtung"),new Linie($linien["32"],"outward","r
ichtung"));

```

```

11.
12. /*Fragendefinition*/
13. $fragen=array(); /*Liste aller verfügbaren Fragen*/
14.
15. $myfrage=new Mobilitaetsfrage("In welcher Richtung gelange ich vom Heidelberger Hauptbahnhof am schnellsten zum Bismarckplatz?", "Bahnpendler", "richtungswahl", $haltestellen["HD Hbf"]); /*statt direkter Eingaber der LocationRef möglich ( dauert dann aber länger, da erst API-Abfrage nötig): HaltestelleNameToId("Heidelberg Hauptbahnhof")*/
16. $myfrage->fahrtB->zielhalt->NewFromId($haltestellen["Bismarckplatz"]); /*statt direkter Eingaber der LocationRef möglich ( dauert dann aber länger, da erst API-Abfrage nötig): ->NewFromId($haltestellen["Bismarckplatz"]);*/
17. $myfrage->fahrtB->linie=$LinienHauptbahnhofNachBismarckplatz;
18. $myfrage->fahrtA->zielhalt->NewFromId($haltestellen["Betriebshof"]);
19. $myfrage->fahrtA->linie=$LinienHauptbahnhofNachBetriebshof;
20. $myfrage->fahrtA->weiter[]=$WeiterfahrtBetriebshofNachBismarckplatz;
21. $myfrage->setFahrtenTitles("über den Betriebshof (Gleis 2)", "direkte Fahrt (Gleis 1)", "unklar");
22. $fragen[]=$myfrage;
23.
24. $myfrage=new Mobilitaetsfrage("Kann ich vom Heidelberger Hauptbahnhof über den Betriebshof schneller zum Bismarckplatz gelangen?", "Bahnpendler", "verbindungswahl", $haltestellen["HD Hbf"]);
25. $myfrage->fahrtA->zielhalt->NewFromId($haltestellen["Bismarckplatz"]);
26. $myfrage->fahrtA->linie=$LinienHauptbahnhofNachBismarckplatz;
27. $myfrage->fahrtB->zielhalt->NewFromId($haltestellen["Betriebshof"]);
28. $myfrage->fahrtB->linie=$LinienHauptbahnhofNachBetriebshof;
29. $myfrage->fahrtB->weiter[]=$WeiterfahrtBetriebshofNachBismarckplatz;
30. $fragen[]=$myfrage;
31.
32. $myfrage=new Mobilitaetsfrage("In welcher Richtung gelange ich vom Betriebshof am schnellsten zum Bismarckplatz?", "Bahnpendler", "richtungswahl", $haltestellen["Betriebshof"]);
33. $myfrage->fahrtB->zielhalt->NewFromId($haltestellen["Bismarckplatz"]);
34. $myfrage->fahrtB->linie=$WeiterfahrtBetriebshofNachBismarckplatz->linie;
35. $myfrage->fahrtA->zielhalt->NewFromId($haltestellen["HD Hbf"]);
36. $myfrage->fahrtA->linie=$LinienBetriebshofNachHauptbahnhof;
37. $myfrage->fahrtA->weiter[]=$new Weiterfahrtmoeglichkeit($haltestellen["Bismarckplatz"], $LinienHauptbahnhofNachBismarckplatz);
38. $myfrage->setFahrtenTitles("über den Hauptbahnhof (Gleis 1)", "direkte Fahrt (Gleis 2)", "unklar");
39. $fragen[]=$myfrage;
40.
41. $myfrage=new Mobilitaetsfrage("Lohnt es sich, zur Universität am Bismarckplatz aus der Linie 33 umzusteigen?", "Studenten Bahnstadt", "umstieg", $haltestellen["Bismarckplatz"]);
42. $myfrage->fahrtA->zielhalt->NewFromId($haltestellen["Peterskirche"]);
43. $myfrage->fahrtA->linie=array(new Linie($linien["33"], "outward", "richtung")); /*33 Richtung Ziegelhausen*/
44. $myfrage->fahrtA->weiter[0]=new Weiterfahrtmoeglichkeit($haltestellen["Uniplatz"], "Fuß", array());
45. $myfrage->fahrtA->weiter[0]->fußminuten=2;
46. $myfrage->fahrtB->zielhalt->NewFromId($haltestellen["Uniplatz"]);
47. $myfrage->fahrtB->linie=array(new Linie($linien["31"], "outward", "richtung"), new Linie($linien["32"], "outward", "richtung")); /*31/32 Richtung Uniplatz*/
48. $myfrage->antworttextadd="Diese Antwort basiert auf der Hypothese eines um zwei Minuten längeren Fußwegs von der Peterskirche aus.";
49. $myfrage->prognosegrenze=60*10; /*10 Minuten*/
50. $fragen[]=$myfrage;
51.

```

```

52. $myfrage=new Mobilitaetsfrage("Soll ich mit der Linie 33 ab Schwetzingen Terasse Richtung Bismarckplatz oder Richtung Rudolf-Diesel-Straße fahren?","Studenten Bahnstadt","richtungswahl",$haltestellen["Schwetzingen Terasse"]);
53. $myfrage->fahrtA->zielhalt->NewFromId($haltestellen["Bismarckplatz"]);
54. $myfrage->fahrtA->linie=array(new Linie($linien["33"],"outward","richtung"));/*33 Richtung Ziegelhausen*/
55. $myfrage->fahrtB->zielhalt->NewFromId($haltestellen["Rudolf-Diesel-Straße"]);
56. $myfrage->fahrtB->linie=array(new Linie($linien["33"],"return","richtung"));/*33 Richtung Emmertsgrund*/
57. $myfrage->fahrtB->weiter[]=new Weiterfahrtmoeglichkeit($haltestellen["Bismarckplatz"],array(new Linie($linien["26"],"return","richtung")));
58. $myfrage->setFahrtenTitles("Richtung Bismarckplatz","Richtung Rudolf-Diesel-Straße","unklar");
59. $fragen[]=$myfrage;
60.
61. $myfrage=new Mobilitaetsfrage("Soll ich ab Schwetzingen Terasse mit der Linie 33 oder mit der Linie 721 fahren?","Studenten Bahnstadt","richtungswahl",$haltestellen["Schwetzingen Terasse"]);/*keine Richtungsangabe, da Wahl nur Richtung Bismarckplatz Sinn macht*/
62. $myfrage->fahrtA->zielhalt->NewFromId($haltestellen["Bismarckplatz"]);
63. $myfrage->fahrtA->linie=array(new Linie($linien["33"],"outward","richtung"));/*33 Richtung Ziegelhausen*/
64. $myfrage->fahrtB->zielhalt->NewFromId($haltestellen["Betriebshof"]);
65. $myfrage->fahrtB->linie=array(new Linie($linien["721"],"return","richtung"));/*721 Richtung Neuenheimer Feld*/
66. $myfrage->fahrtB->weiter[]=$WeiterfahrtBetriebshofNachBismarckplatz;
67. $myfrage->setFahrtenTitles("mit der Linie 33","mit der Linie 721","unklar");
68. $fragen[]=$myfrage;
69.
70. $myfrage=new Mobilitaetsfrage("Soll ich mit Ziel Schwetzingen Terasse am Bismarckplatz in die Linie 26 einsteigen?","Studenten Bahnstadt","verbindungswahl",$haltestellen["Bismarckplatz"]);
71. $myfrage->fahrtA->zielhalt->NewFromId($haltestellen["Schwetzingen Terasse"]);
72. $myfrage->fahrtA->linie=array(new Linie($linien["33"],"return","richtung"));/*33 Richtung Emmertsgrund*/
73. $myfrage->fahrtB->zielhalt->NewFromId($haltestellen["Rudolf-Diesel-Straße"]);
74. $myfrage->fahrtB->linie=array(new Linie($linien["26"],"outward","richtung"));/*26 Richtung Kirchheim*/
75. $myfrage->fahrtB->weiter[]=new Weiterfahrtmoeglichkeit($haltestellen["Schwetzingen Terasse"],array(new Linie($linien["33"],"outward","richtung")));
76. $fragen[]=$myfrage;
77.
78. $myfrage=new Mobilitaetsfrage("Soll ich mit Ziel Schwetzingen Terasse am Seegarten in die Linie 26 einsteigen?","Studenten Bahnstadt","verbindungswahl",$haltestellen["Seegarten"]);
79. $myfrage->fahrtA->zielhalt->NewFromId($haltestellen["Schwetzingen Terasse"]);
80. $myfrage->fahrtA->linie=array(new Linie($linien["33"],"return","richtung"));/*33 Richtung Emmertsgrund*/
81. $myfrage->fahrtB->zielhalt->NewFromId($haltestellen["Rudolf-Diesel-Straße"]);
82. $myfrage->fahrtB->linie=array(new Linie($linien["26"],"outward","richtung"));/*26 Richtung Kirchheim*/
83. $myfrage->fahrtB->weiter[]=new Weiterfahrtmoeglichkeit($haltestellen["Schwetzingen Terasse"],array(new Linie($linien["33"],"outward","richtung")));
84. $fragen[]=$myfrage;
85.

```

```

86. $myfrage=new Mobilitaetsfrage("Wenn ich jetzt am Uniplatz einsteige, wie lange muss
ich dann am Bismarckplatz auf die Linie 33 warten?","Studenten Bahnstadt","umsteigezeit",
$haltestellen["Uniplatz"]);/*keine Richtungsangabe, da Frage nur sinnvoll, wenn
Richtung Emmertsgrund (Ziegelhausen an Peterskirche sinnvoller)*/
87. $myfrage->fahrtA->zielhalt->NewFromId($haltestellen["Bismarckplatz"]);
88. $myfrage->fahrtA->
>linie=array(new Linie($linien["31"],"return","richtung"),new Linie($linien["32"],"return",
"richtung"));/*31/32 Richtung Neuenheimer Feld*/
89. $myfrage->fahrtA->
>weiter[]=new Weiterfahrtmoeglichkeit($haltestellen["HD Hbf"],array(new Linie($linien["33"],
"return","richtung")));
90. $myfrage->wartezeitinklStarthalt=false;
91. $fragen[]=$myfrage;
92.
93. $myfrage=new Mobilitaetsfrage("An welchen Bahnsteig muss ich mich am Gadamerplatz stellen,
um schnellstmöglich zum Bismarckplatz zu gelangen?","Studenten Bahnstadt","nurtxt","");
94. $myfrage->
>antworttextadd="Bis 17.12.2017:\nBesser an eine andere Haltestelle.\nDie Haltestelle
Gadamerplatz befindet sich noch im Bau.\n\nAb 17.12.2017:\nDerzeit fährt nur an einem
Bahnsteig eine Straßenbahn Richtung Bismarckplatz.\nStellen Sie sich an diesen
Bahnsteig.";
95. $fragen[]=$myfrage;
96.
97. $myfrage=new Mobilitaetsfrage("Bismarckplatz zum Hauptbahnhof: Bus oder Bahn?","Bahnpendler",
"richtungswahl",$haltestellen["Bismarckplatz"]);/*natürlich mit Bus zum S-Bahnhof*/
98. $myfrage->fahrtA->zielhalt->NewFromId($haltestellen["HD Hbf"]);
99. $myfrage->fahrtA->
>linie=array(new Linie($linien["5"],"return","richtung"),new Linie($linien["9"],"outward",
"richtung"),new Linie($linien["21"],"outward","richtung"));
100. $myfrage->fahrtB->zielhalt->NewFromId($haltestellen["HD Hbf"]);
101. $myfrage->fahrtB->
>linie=array(new Linie($linien["32"],"return","richtung"),new Linie($linien["33"],"return",
"richtung"),new Linie($linien["34"],"return","richtung"));
102. $myfrage->setFahrtenTitles("mit der Straßenbahn","mit dem Bus","unklar");
103. $fragen[]=$myfrage;
104.
105. $myfrage=new Mobilitaetsfrage("Wie lange muss ich, wenn ich von der Marstallstraße zum
Hauptbahnhof die S-Bahn benutze, insgesamt warten?","Bahnpendler","umsteigezeit",
$haltestellen["Marstallstr."]);/*natürlich mit Bus zum S-Bahnhof*/
106. $myfrage->fahrtA->zielhalt->NewFromId($haltestellen["HD-Altstadt"]);
107. $myfrage->fahrtA->
>linie=array(new Linie($linien["35"],"outward","richtung"),new Linie($linien["32"],"return",
"richtung"));
108. $myfrage->fahrtA->
>weiter[]=new Weiterfahrtmoeglichkeit($haltestellen["HD Hbf"],array(new Linie("de:ddb:92V01",
"return","richtung"),new Linie("de:ddb:92V02","return","richtung"),new Linie("de:ddb:92V05",
"return","richtung"),new Linie("de:ddb:92V06","return","richtung")));
109. $myfrage->wartezeitinklStarthalt=true;
110. $fragen[]=$myfrage;
111.
112. $myfrage=new Mobilitaetsfrage("Wie lange muss ich in Mannheim am Schloss warten, um mit
der Straßenbahn zum Hauptbahnhof zu gelangen?","Bahnpendlerstudenten Mannheim",
"umsteigezeit",$haltestellen["Mannheim Schloss"]);/*natürlich mit Bus zum S-Bahnhof*/
113. $myfrage->fahrtA->zielhalt->NewFromId($haltestellen["MA Hbf"]);
114. $myfrage->fahrtA->
>linie=array(new Linie($linien["1"],"return","richtung"),new Linie($linien["5"],"return",
"richtung"),new Linie($linien["5A"],"return","richtung"),new Linie($linien["15"],
"outward","richtung"));
115. $myfrage->wartezeitinklStarthalt=true;
116. $fragen[]=$myfrage;
117.

```

```

118.$myfrage=new Mobilitaetsfrage("Nutzergruppe Emmertsgrund: Erreiche ich meinen Anschl
uss in Rohrbach Süd?","Fahrgastbeirat","umsteigezeit","de:08221:1130");/*Mombertplat
z als zentrale Haltestelle ausgewählt, Verfeinerung möglich*/
119.$myfrage->fahrtA->zielhalt->NewFromId("de:08221:1278");
120.$myfrage->fahrtA-
>linie=array(new Linie($linien["33"],"outward","richtung"),new Linie("de:vrn:19029",
"outward","richtung"),new Linie("de:vrn:19027","outward","richtung"));
121.$myfrage->fahrtA-
>weiter[]=new Weiterfahrtmoeglichkeit($haltestellen["Bismarckplatz"],array(new Linie
($linien["23"],"return","richtung"),new Linie($linien["24"],"return","richtung")));
122.$myfrage-
>antworttextadd="Wenn Sie länger warten müssen als der Takt auf der gewünschten Stre
cke ist, können Sie zu Hause warten, da Sie den Anschluss nicht erreichen.";
123.$fragen[]=$myfrage;
124.
125.$myfrage=new Mobilitaetsfrage("Nutzergruppe Schwetzingen: Erreiche ich mit der Linie
22 in Eppelheim meinen Anschluss an die Linie 713?","Fahrgastbeirat","umsteigezeit"
,$haltestellen["Bismarckplatz"]);/*nötig, da Mobinetz nicht umgesetzt*/
126.$myfrage->fahrtA->zielhalt->NewFromId("de:08226:696");
127.$myfrage->fahrtA->linie=array(new Linie($linien["22"],"outward","richtung"));
128.$myfrage->fahrtA-
>weiter[]=new Weiterfahrtmoeglichkeit("de:08226:7123",array(new Linie("de:vrn:33713"
,"outward","richtung")));
129.$myfrage-
>antworttextadd="Wenn Sie länger warten müssen als der Takt der Linie 22 ist, können
Sie die nächste Linie 22 nehmen, da Sie den Anschluss nicht erreichen.\nBedanken Si
e sich bei denjenigen, die das Teilprojekt einer durchgehenden Straßenbahn nach Schw
etzingen aus dem Mobilitätsnetz verhindert haben.";
130.$fragen[]=$myfrage;
131.
132.$myfrage=new Mobilitaetsfrage("Wie komme ich zu meinem Ziel?","FAQ","nurtext","");
133.$myfrage-
>antworttextadd="Weiß ich nicht. Bitte verwenden Sie die <a href=\"http://reiseausku
nft.bahn.de/bin/query.exe/dn?tariffClass=1\">Fahrplanauskunft</a>.\nDieser Dienst di
ent dazu, Fragen zu beantworten, die durch die Fahrplanauskunft nicht so einfach erm
ittelt werden können.";
134.$fragen[]=$myfrage;
135.
136.$myfrage=new Mobilitaetsfrage("Ist dieser Dienst etwas für mich?","FAQ","nurtext",""
);
137.$myfrage->antworttextadd="Dieser Dienst richtet sich an Sie, wenn Sie -
etwa als Pendler - grundsätzlich wissen, wie Sie Ihr Ziel erreichen.
138.Anders als herkömmliche Fahrplanauskunftdienste informiert Sie dieser Dienst anhand
von Echtzeitdaten darüber, wie sie nach dem Ihnen bekannten Fahrplan <b>heute</b> ih
r Ziel schnellstmöglich erreichen. Daher kann er Sie insbesondere dann unterstützen
, wenn Sie in einem sehr gut erschlossenen Gebiet mit vielen Fahrtalternativen unter
wegs sind.
139.Diese Information geschieht nicht anhand einer klassischen Fahrplanabfrage, sondern
anhand von speziell für Ihre Nutzergruppe vorbereiteten Fragen, die präzise nach für
auf dem Fahrtweg notwendig werdende Entscheidungen nötigen Informationen fragen. Di
e Antwort erhalten Sie als Ampel für eine schnelle Entscheidung und auch in ausführli
cherer Textform für diejenigen, die sich lieber selbst ein Bild machen wollen.";
140.$fragen[]=$myfrage;
141.
142.$myfrage=new Mobilitaetsfrage("Wo kommen die Daten her?","FAQ","nurtext","");
143.$myfrage->antworttextadd="Die Daten werden über die TRIAS-
API des Verkehrsverbunds Rhein-Neckar (VRN) ermittelt.";
144.$fragen[]=$myfrage;
145.>>

```

4.1.2.2 triastest.php

```

1. <?php
2. require_once("trias.php");
3. $zeitpunktxml=date(DateTime::RFC3339,time());
4. $StopPointRef="de:08221:4256";
5. $triasmittel=' <StopEventRequest>

```

```

6.         <Location><LocationRef><StopPointRef>de:08221:4256</StopPointRef></L
ocationRef><DepArrTime>'.$zeitpunktxml.'</DepArrTime></Location>
7.         <Params>
8.             <StopEventPolicy><StopEventType>departure</StopEventType><TimeWi
ndow>15H</TimeWindow><NumberOfResults>700</NumberOfResults></StopEventPolicy>
9.             <StopEventContentFilter><IncludeOnwardCalls>true</IncludeOnwardC
alls><IncludeRealtimeData>true</IncludeRealtimeData></StopEventContentFilter>
10.            <StopEventDataFilter><LineFilter><Line><LineRef>de:vrn:19033</Li
neRef></Line><Exclude>true</Exclude></LineFilter></StopEventDataFilter><!--
sollte laut Dokumentation funktionieren, filtert real aber nicht -->
11.        </Params>
12.    </StopEventRequest>';//<DirectionRef>return</DirectionRef>
13.    /*entgegen Dokumentation nicht funktionierende Abfrage zum Fahrtverlauf:
    $triasmittel='         <TripInfoRequest>
14.            <JourneyRef>de:vrn:19035:174</JourneyRef>
15.            <OperatingDayRef>2017-11-13</OperatingDayRef>
16.        </TripInfoRequest>';*/
17. echo '<?xml version="1.0" encoding="iso-8859-1" ?>'.gettrias($triasmittel);
18. ?>

```

4.1.3 Ausgabe

Die Ausgabe erfolgt über Dateien nach dem Muster `antwort{form}.php`, und zwar `antwortapi.php` für eine API, `antwortampel.php` für die Ampelgrafik und `antwortdetails.php` für die Fahrtdetails.

Der Einstieg in die Ausgabe erfolgt über die Datei `index.php`, welche die Auswahl der Nutzergruppe ermöglicht. Die Auswahl der konkreten Frage ermöglicht im Anschluss daran die Datei `fragen.php`.

Die Ausgabeseiten nutzen das Stylesheet `style.css`, welche das Stylesheet des einheitlichen Erscheinungsbilds importiert und Spezialanpassungen für diese Anwendung definiert.

Die Ampelgrafik wird über die Datei `ampel.svg.php` generiert.

4.1.3.1 style.css

```

1. <!--
2. @import "https://www.tillmenke.de/style.css";
3. a.frage{display:block;width:100%;font-size:1.5em;line-height:1.0em;
padding:0.5em;padding-right:0;}
4. div.sidebar{float:none;}
5. div.sidebar_text{margin-top:10px;margin-bottom:10px;}
6. -->

```

4.1.3.2 ampel.svg.php

```

1. <?php header('Content-type: image/svg+xml');
2. echo '<?xml version="1.0" encoding="UTF-
8" standalone="no"?>'; /*einige Webserver erlauben bei PHP-
Datei kein unmittelbares <?xml*/
3. ?>
4. <svg xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:rdf="http://www.w3.org/1999/0
2/22-rdf-syntax-
ns#" xmlns:svg="http://www.w3.org/2000/svg" xmlns="http://www.w3.org/2000/svg" xml:s
pace="preserve"
5.     width="699.97137"
6.     height="294.1123"
7.     style="fill-rule:evenodd"
8.     viewBox="0 0 197.54748 83.00503"
9.     id="ampel"
10.    version="1.1">
11. <defs>
12. <style type="text/css">
13.     .ampelbox {fill:black; stroke:white;stroke-width:0.7056}
14.     .ampellicht {stroke:#4C4C4C;stroke-width:0.7056}
15.     .ampellichtsymbol_modell12 {<?php if($_GET["modell"]!=2){echo 'display:none';} ?>
}

```

```

16. .aus {fill:#191919}
17. .gruen {fill:lime}
18. .rot {fill:red}
19. .gelb {fill:yellow}
20. .weiss {fill:white}
21. text {font-weight:normal;font-family:'Calibri',sans-serif}
22. .achtungbox {<?php if(!$_GET["achtungbox"]){echo 'display:none';} ?>}
23. rect.achtungbox {fill:#663333}
24. text.achtungbox, path.achtungbox {font-size:4.5166px;fill:red}
25. text.ampeltext {fill:black;font-size:8.4667px;}
26. </style>
27. </defs>
28. <g
29.   id="Ebene_x0020_1"
30.   transform="translate(-5.0133204,-31.71697)">
31. <!-- Beschreibung Ampelbox
32. <text
33.   x="5.3660998"
34.   y="37.583302"
35.   class="ampeltext"
36.   id="ampelrot_text">ja</text>
37.
38. <text
39.   x="82.514"
40.   y="37.583302"
41.   class="ampeltext"
42.   id="ampelgelb_text">unklar (s. u.)</text>
43.
44. <text
45.   x="187.526"
46.   y="37.583302"
47.   class="ampeltext"
48.   id="ampelgruen_text">nein</text> -->
49.
50. <!-- Ampelbox -->
51. <rect
52.   class="ampelbox"
53.   x="5.3661199"
54.   y="40.009102"
55.   width="65.613998"
56.   height="65.613998"
57.   id="ampelbox1" />
58. <rect
59.   class="ampelbox"
60.   x="70.980103"
61.   y="40.009102"
62.   width="65.613998"
63.   height="65.613998"
64.   id="ampelbox2" />
65. <rect
66.   class="ampelbox"
67.   x="136.59399"
68.   y="40.009102"
69.   width="65.613998"
70.   height="65.613998"
71.   id="ampelbox3" />
72. <!-- Ampellichter -->
73. <ellipse
74.   class="ampellicht <?php if($_GET["licht"]=="rot"&&$_GET["modell"]!=2){echo 'rot';
75.   }else{echo 'aus';} ?>"
76.   cx="38.173"
77.   cy="72.816101"
78.   rx="29.2969"
79.   ry="29.2969"
80.   id="ampelrot" />

```



```

81.     class="ampellicht <?php if($_GET["licht"]=="gelb"){if($_GET["modell"]==2){echo 'w
eiss';}else{echo 'gelb';}}else{echo 'aus';} ?>"
82.     cx="103.787"
83.     cy="72.816101"
84.     rx="29.2969"
85.     ry="29.2969"
86.     id="ampelgelb"/>
87.     <ellipse
88.     class="ampellicht <?php if($_GET["licht"]=="gruen"&&$_GET["modell"]!=2){echo 'gru
en';}else{echo 'aus';} ?>"
89.     cx="169.401"
90.     cy="72.816101"
91.     rx="29.2969"
92.     ry="29.2969"
93.     id="ampelgruen" />
94.     <!-- Ampellichtsymbole -->
95.     <path
96.     class="ampellichtsymbol_modell12 <?php if($_GET["licht"]=="rot"&&$_GET["modell"]==
2){echo 'weiss';}else{echo 'aus';} ?>"
97.     d="M 24.901451,74.682746 C 27.085851,76.748646 29.321051,78.873746 31.615551,81.0
66646 33.901551,83.259546 36.212951,85.520146 38.549751,87.848446 38.659851,87.96694
6 38.744451,88.085546 38.795251,88.212546 38.846051,88.331046 38.846051,88.474946 38
.803751,88.644346 38.769851,88.813646 38.668251,88.999946 38.507451,89.203146 38.355
051,89.406346 38.134851,89.651846 37.855451,89.931246 37.584551,90.202146 37.355951,
90.405346 37.144251,90.540846 36.941051,90.684746 36.763251,90.769446 36.610851,90.7
86346 36.466951,90.811746 36.331451,90.803346 36.212951,90.769446 36.102851,90.72714
6 35.992851,90.650946 35.891251,90.557746 L 19.618251,74.310246 C 19.423551,74.09014
6 19.262651,73.878446 19.135651,73.675246 19.008651,73.472046 18.949451,73.184146 18
.949451,72.820146 18.949451,72.540746 19.000251,72.286746 19.101851,72.066546 19.194
951,71.837946 19.372751,71.592446 19.618251,71.329946 L 35.891251,55.082446 C 35.992
851,54.980846 36.102851,54.913146 36.212951,54.870746 36.331451,54.828446 36.466951,
54.819946 36.610851,54.845346 36.763251,54.870746 36.941051,54.955446 37.144251,55.0
99346 37.355951,55.234846 37.584551,55.438046 37.855451,55.700546 38.134851,55.98834
6 38.355051,56.225446 38.507451,56.437146 38.668251,56.640346 38.769851,56.826546 38
.803751,56.995946 38.846051,57.156746 38.846051,57.309146 38.795251,57.436146 38.744
451,57.571646 38.659851,57.698646 38.549751,57.808746 36.212951,60.128546 33.901551,
62.380746 31.615551,64.573546 29.321051,66.766446 27.085851,68.891546 24.901451,70.9
57446 L 56.58375,70.957446 C 56.693851,70.957446 56.80385,70.991346 56.90545,71.0590
46 56.998651,71.126746 57.09175,71.228346 57.15945,71.363846 57.23565,71.507746 57.2
9495,71.702546 57.33725,71.939546 57.379651,72.176646 57.39655,72.472946 57.39655,72
.820146 57.39655,73.167246 57.379651,73.463546 57.33725,73.700646 57.29495,73.937746
57.23565,74.132446 57.15945,74.267946 57.09175,74.411846 56.998651,74.513446 56.905
45,74.581146 56.80385,74.648946 56.693851,74.682746 56.58375,74.682746 L 24.901451,7
4.682746 z"
98.     id="Pfeillinks"/>
99.     <path
100.    class="ampellichtsymbol_modell12 <?php if($_GET["licht"]=="gruen"&&$_GET["modell"]
==2){echo 'weiss';}else{echo 'aus';} ?>"
101.    d="M 182.67245,70.957446 C 180.47965,68.891546 178.23595,66.766446 175.94995,64.5
73546 173.67245,62.380746 171.36105,60.128546 169.03265,57.808746 168.91415,57.69864
6 168.83795,57.571646 168.78715,57.436146 168.73635,57.309146 168.72785,57.156746 16
8.77025,56.995946 168.81255,56.826546 168.91415,56.640346 169.06655,56.437146 169.22
745,56.225446 169.44755,55.988346 169.72695,55.700546 169.98945,55.438046 170.22645,
55.234846 170.43815,55.099346 170.64135,54.955446 170.81915,54.870746 170.97155,54.8
45346 171.11545,54.819946 171.25095,54.828446 171.36105,54.870746 171.47955,54.91314
6 171.58965,54.980846 171.69125,55.082446 L 187.95565,71.329946 C 188.15885,71.54164
6 188.31975,71.753346 188.44675,71.964946 188.56525,72.168146 188.62455,72.456046 18
8.62455,72.820146 188.62455,73.099546 188.58225,73.353546 188.48065,73.573646 188.37
905,73.802246 188.20965,74.039346 187.95565,74.310246 L 171.69125,90.557746 C 171.58
965,90.650946 171.47955,90.727146 171.36105,90.769446 171.25095,90.803346 171.11545,
90.811746 170.97155,90.786346 170.81915,90.769446 170.64135,90.684746 170.43815,90.5
40846 170.22645,90.405346 169.98945,90.202146 169.72695,89.931246 169.44755,89.65184
6 169.22745,89.406346 169.06655,89.203146 168.91415,88.999946 168.81255,88.813646 16
8.77025,88.644346 168.72785,88.474946 168.73635,88.331046 168.78715,88.212546 168.83
795,88.085546 168.91415,87.966946 169.03265,87.848446 171.36105,85.520146 173.67245,
83.259546 175.94995,81.066646 178.23595,78.873746 180.47965,76.748646 182.67245,74.6
82746 L 150.99865,74.682746 C 150.88015,74.682746 150.77855,74.648946 150.67695,74.5

```

```

81146 150.57535,74.513446 150.49065,74.411846 150.41445,74.267946 150.34675,74.13244
6 150.28745,73.937746 150.24515,73.700646 150.20285,73.463546 150.17745,73.167246 15
0.17745,72.820146 150.17745,72.472946 150.20285,72.176646 150.24515,71.939546 150.28
745,71.702546 150.34675,71.507746 150.41445,71.363846 150.49065,71.228346 150.57535,
71.126746 150.67695,71.059046 150.77855,70.991346 150.88015,70.957446 150.99865,70.9
57446 L 182.67245,70.957446 z"
102.   id="PfeilRechts" />
103. <!-- Achtungbox -->
104. <rect
105.   class="achtungbox"
106.   x="5.3660898"
107.   y="105.623"
108.   width="196.842"
109.   height="9.099"
110.   id="achtungbox" />
111. <text
112.   x="17.2477"
113.   y="111.665"
114.   class="achtungbox"
115.   id="achtungstext">Achtung: Mindestens eine der geprüften Verbindungen beruht nicht
auf Echtzeitinformationen.</text>
116. <path
117.   class="achtungbox"
118.   d="M 10.862,112.111 C 10.862,112.1725 10.8573,112.2226 10.8488,112.2623 10.8412,1
12.302 10.8251,112.3342 10.8024,112.3597 10.7788,112.3843 10.7485,112.4013 10.7117,1
12.4117 10.6748,112.4221 10.6275,112.4268 10.5708,112.4268 10.5131,112.4268 10.4658,
112.4221 10.428,112.4117 10.3902,112.4013 10.3599,112.3843 10.3382,112.3597 10.3155,
112.3342 10.3003,112.302 10.2918,112.2623 10.2843,112.2226 10.2795,112.1725 10.2795,
112.111 10.2795,112.0505 10.2843,111.9994 10.2918,111.9578 10.3003,111.9162 10.3155,
111.8841 10.3382,111.8604 10.3599,111.8377 10.3902,111.8207 10.428,111.8103 10.4658,
111.8009 10.5131,111.7952 10.5708,111.7952 10.6275,111.7952 10.6748,111.8009 10.7117
,111.8103 10.7485,111.8207 10.7788,111.8377 10.8024,111.8604 10.8251,111.8841 10.841
2,111.9162 10.8488,111.9578 10.8573,111.9994 10.862,112.0505 10.862,112.111 z M 10.7
76,111.3659 C 10.776,111.381 10.7722,111.3943 10.7656,111.4047 10.7599,111.416 10.74
76,111.4255 10.7315,111.433 10.7145,111.4397 10.6937,111.4453 10.6672,111.4491 10.64
17,111.4529 10.6095,111.4548 10.5708,111.4548 10.532,111.4548 10.4999,111.4529 10.47
34,111.4491 10.4479,111.4453 10.427,111.4397 10.41,111.433 10.393,111.4255 10.3817,1
11.416 10.375,111.4047 10.3694,111.3943 10.3656,111.381 10.3656,111.3659 L 10.3183,1
08.7231 C 10.3164,108.7013 10.3193,108.6815 10.3268,108.6654 10.3344,108.6484 10.347
6,108.6351 10.3675,108.6238 10.3864,108.6124 10.4119,108.6049 10.445,108.6001 10.477
2,108.5954 10.5188,108.5935 10.5708,108.5935 10.6209,108.5935 10.6616,108.5964 10.69
37,108.6011 10.7258,108.6068 10.7523,108.6153 10.7712,108.6266 10.7911,108.637 10.80
43,108.6512 10.8119,108.6663 10.8195,108.6824 10.8232,108.7013 10.8232,108.7231 L 10
.776,111.3659 z M 7.3015,110.5097 C 7.3015,109.6142 7.623,108.8455 8.2641,108.2035 8
.9061,107.5624 9.6749,107.2409 10.5703,107.2409 11.4667,107.2409 12.2354,107.5624 12
.8765,108.2035 13.5186,108.8455 13.8391,109.6142 13.8391,110.5097 13.8391,111.4061 1
3.5186,112.1748 12.8765,112.8159 12.2354,113.458 11.4667,113.7785 10.5703,113.7785 9
.6749,113.7785 8.9061,113.458 8.2641,112.8159 7.623,112.1748 7.3015,111.4061 7.3015,
110.5097 z M 7.6485,110.5097 C 7.6485,111.3172 7.9331,112.0094 8.5042,112.5861 9.074
4,113.1639 9.7637,113.4523 10.5703,113.4523 11.3759,113.4523 12.0643,113.1639 12.635
4,112.5861 13.2075,112.0094 13.493,111.3172 13.493,110.5097 13.493,109.705 13.2075,1
09.0129 12.6354,108.4351 12.0643,107.8574 11.3759,107.568 10.5703,107.568 9.7637,107
.568 9.0744,107.8574 8.5042,108.4351 7.9331,109.0129 7.6485,109.705 7.6485,110.5097
z"
119.   id="achtungsymbol"/>
120. </g>
121. </svg>

```

4.1.3.3 antwortapi.php

```

1. <?php
2. require_once("routen.php");
3. if(!isset($_GET["id"])){$_GET["id"]="none";}
4. if(isset($fragen[$_GET["id"]])){
5.   $frage = $fragen[$_GET["id"]];
6.   $antwort=$frage->getAntwort();
7.   if(!isset($_GET["mode"])){$_GET["mode"]="text";}

```

```

8.     if($_GET["mode"]=="fahrtADetails"){foreach($antwort-
>fahrtADetails as $detail){echo $detail."\n";}}
9.     if($_GET["mode"]=="fahrtBDetails"){foreach($antwort-
>fahrtBDetails as $detail){echo $detail."\n";}}
10.    if($_GET["mode"]=="text"){echo $antwort->text;}
11.    if($_GET["mode"]=="ampeltext"){
12.        $description=$frage->getFahrtenTitles();
13.        if($antwort->ampellicht==0){echo "Weiß ich nicht";}
14.        if($antwort->ampellicht==1){echo $description["fahrtA"];}
15.        if($antwort->ampellicht==2){echo $description["fahrtUnentschieden"];}
16.        if($antwort->ampellicht==3){echo $description["fahrtB"];}
17.    }
18.    if($_GET["mode"]=="ampellicht"){
19.        echo $antwort->ampellicht;
20.    }
21. }else{
22.     echo "Fehler: keine Frage ausgewählt";
23. }
24. ?>

```

4.1.3.4 index.php

```

1. <?php
2. require_once("routen.php");
3. ?>
4. <!DOCTYPE HTML PUBLIC "-
//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
5. <html>
6. <head>
7. <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
8. <meta name="Generator" content="Microsoft Word 12 (filtered)">
9. <link rel="stylesheet" href="style.css" type="text/css">
10. <title>Mobilitätsfragen</title>
11. </head>
12.
13. <body>
14. <div class="title">
15. <p class="title">Mobilitätsfragen</p>
16. </div>
17. <p class="navigation">
18. <a class="activebutton" href="#">Nutzergruppe</a>
19. <span> </span>
20. <a class="button" href="frage.php">Frage</a>
21. <span> </span>
22. <a class="button" onclick="alert('Bitte erst Frage wählen!')"><acronym title="Bitte
erst Frage wählen!">Antwort</acronym></a>
23. </p>
24. <p style="float:right;"><a href="https://www.tillmenke.de/player/validate.php"><im
g src="https://www.tillmenke.de/bullets/validhtml4.01.png" alt="Valid HTML 4.01"></a
></p>
25. <p><br></p>
26.
27. <p class="fragenliste first">
28. <?php
29. $nutzergruppen = array();
30. echo ' <a href="frage.php?category=" class="button frage">Allgemein</a><br>
31. ' ;
32. foreach($fragen as $id=>$frage){
33.     if($frage->category==""){ $frage->category="Allgemein";}
34.     if(!in_array($frage->category,$nutzergruppen)&&$frage-
>category!="FAQ"/*FAQ werden für jede Nutzergruppe ausgegeben*/){
35.         echo ' <a href="frage.php?category='.urlencode($frage-
>category).' ' class="button frage">'.$frage->category.' </a><br>
36. ' ;
37.         $nutzergruppen[]=$frage->category;
38.     }
39. }
40. ?>

```

```

41. </p>
42. </body>
43. </html>

```

4.1.3.5 frage.php

```

1. <?php
2. require_once("routen.php");
3. $title="Mobilitätsfragen";
4. if(isset($_GET["category"])){
5.   if($_GET["category"]!=""){/*Nutzergruppe in Titel nur, wenn Nutzergruppe gewählt*/
6.     foreach($fragen as $id=>$frage){/*Ausgabe nur, wenn wirklich Frage vorhanden*/
7.       if($_GET["category"]==$frage-
8. >category){$title="Mobilitätsfragen: Nutzergruppe ".htmlspecialchars($frage-
9. >category);}
10.    }
11.  }
12. ?>
13. <!DOCTYPE HTML PUBLIC "-
14. //W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
15. <html>
16. <head>
17. <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
18. <meta name="Generator" content="Microsoft Word 12 (filtered)">
19. <link rel="stylesheet" href="style.css" type="text/css">
20. <title><?php echo $title; ?></title>
21. </head>
22. <body>
23. <div class="title">
24. <p class="title"><?php echo $title; ?></p>
25. </div>
26. <p class="navigation">
27. <a class="button" href="index.php">Nutzergruppe</a>
28. <span> </span>
29. <a class="activebutton" href="?">Frage</a>
30. <span> </span>
31. <a class="button" onclick="alert('Bitte erst Frage wählen!')"><acronym title="Bi
32. tte erst Frage wählen!">Antwort</acronym></a>
33. <p style="float:right;"><a href="https://www.tillmenke.de/player/validate.php"><im
34. g src="https://www.tillmenke.de/bullets/validhtml4.01.png" alt="Valid HTML 4.01"></a
35. ></p>
36. <p><br></p>
37. <p class="fragenliste first">
38. <?php
39. foreach($fragen as $id=>$frage){
40.   if(!isset($_GET["category"])){/*ohne Angabe gebe einfach alle Fragen aus*/
41.   echo ' <a href="antwortampel.php?id='.$id.'" class="button frage">'.$frage-
42. >title.'</a><br>
43.   '};
44.   else{/*nur Fragen zur Nutzergruppe und un kategorisierte und FAQ (die auf jeder Sei
45. te angezeigt werden sollen, u. a. wegen Erfordernis der Namensnennung VRN)*/ if($_GE
46. T["category"]==$frage->category||$frage->category=="||$frage->category=="FAQ"){
47.   echo ' <a href="antwortampel.php?id='.$id.'" class="button frage">'.$frage-
48. >title.'</a><br>
49.   '};}
50. }
51. ?>
52. </p>
53. <?php
54. if(isset($_GET["category"])){if($_GET["category"]!=""){/*Hinweis auf Lesezeichen nur
55. , wenn wirklich Nutzergruppe gewählt*/
56. echo ' <div class="sidebar">

```

```

51. <div class="sidebartext"><p><b>Tipp: </b>Erstellen Sie eine Verknüpfung von Ihrer
    Startseite auf diese Seite, um schnellstmöglich Entscheidungshilfen zu erhalten.</p
    ></div>
52. </div>
53. ';
54. }}
55. ?>
56. </body>
57. </html>

```

4.1.3.6 antwortampel.php

```

1. <?php
2. require_once("routen.php");
3. if(!isset($_GET["id"])){$_GET["id"]="none";}
4. if(isset($fragen[$_GET["id"]])){
5.     $frage = $fragen[$_GET["id"]];
6.     $ampelmodell = 1;
7.     if($frage->art=="verbindungswahl"){ $ampelmodell=1;}
8.     if($frage->art=="richtungswahl"){ $ampelmodell=2;}
9.     if($frage->art=="umstieg"){ $ampelmodell=1;}
10.    if($frage->art=="wartezeit"){ $ampelmodell=1;}
11.    $antwort=$frage->getAntwort();
12.    $achtungbox="";
13.    if($antwort->isEchtzeit===false){ $achtungbox="ja";}
14.    $ampellicht=""; /*kein Licht als Standardlicht*/
15.    if($antwort->ampellicht==0){ $ampellicht="";}
16.    if($antwort->ampellicht==1){ $ampellicht="rot";}
17.    if($antwort->ampellicht==2){ $ampellicht="gelb";}
18.    if($antwort->ampellicht==3){ $ampellicht="gruen";}
19.    $description=$frage->getFahrtenTitles();
20.    ?>
21. <!DOCTYPE HTML PUBLIC "-
    //W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
22. <html>
23. <head>
24. <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
25. <meta name="Generator" content="Microsoft Word 12 (filtered)">
26. <link rel="stylesheet" href="style.css" type="text/css">
27. <title>Mobilitätsfragen: <?php echo $frage->title; ?></title>
28. </head>
29.
30. <body>
31. <div class="title">
32. <p class="title">Mobilitätsfragen: <?php echo $frage-
    >title; ?> <span class="titledate">Stand: <?php echo date("d.m.Y, H:i")." Uhr"; ?></
    span></p>
33. </div>
34. <p class="navigation">
35. <a class="button" href="index.php">Nutzergruppe</a>
36. <span> </span>
37. <a class="button" href="frage.php">Frage</a>
38. <span> </span>
39. <a class="activebutton" href="#">Antwort</a>
40. </p>
41. <p style="float:right;"><a href="https://www.tillmenke.de/player/validate.php"><im
    g src="https://www.tillmenke.de/bullets/validhtml4.01.png" alt="Valid HTML 4.01"></a
    ></p>
42. <p><br></p>
43.
44. <?php if($frage->art=="nurtext"){ }else{ ?>
45. <p class="first"></p>
46. <object data="ampel.svg.php?achtungbox=<?php echo $achtungbox; ?>&licht=<?php echo
    $ampellicht; ?>&modell=<?php echo $ampelmodell; ?>" width="100%" id="ampel" type="i
    mage/svg+xml">Ihr Browser kann leider keine SVG-
    Objekte anzeigen. Diese Funktionalität ist für die Ampeldarstellung zwingend erforde
    rlich.</object>
47. <table class="ampeltitle" width="100%">

```

```

48.     <tr>
49.         <td width="33%" style="text-
align:left"><p class="description"><?php echo $description["fahrtA"]; ?></p></td>
50.         <td width="33%" style="text-
align:center"><p class="description"><?php echo $description["fahrtUnentschieden"];
?></p></td>
51.         <td width="33%" style="text-
align:right"><p class="description"><?php echo $description["fahrtB"]; ?></p></td>
52.     </tr>
53. </table>
54. <?php } ?>
55.     <p class="first"></p>
56.     <p class="subtitle"><?php echo str_replace("\n", "<br>", $antwort->text); ?></p>
57.     <p class="fragenliste first">
58.         <a href="index.php" class="button frage">Ich habe weitere Fragen.</a><br><!--
Link für bessere Erreichbarkeit auf Mini-Touchscreen auf Raspberry Pi-->
59.         <a href="antwortdetails.php?id=<?php echo $_GET["id"]; ?>" class="button frage">I
ch möchte Fahrtdetails sehen.</a>
60.     </p>
61. </body>
62. </html>
63. <?php
64. }else{
65. ?>
66. <!DOCTYPE HTML PUBLIC "-
//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
67. <html>
68. <head>
69. <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
70. <meta name="Generator" content="Microsoft Word 12 (filtered)">
71. <link rel="stylesheet" href="style.css" type="text/css">
72. <title>Mobilitätsfragen</title>
73. </head>
74.
75. <body>
76. <div class="title">
77. <p class="title">Frage</p>
78. </div>
79. <p class="navigation">
80. <a class="button" href="index.php">Nutzergruppe</a>
81. <span> </span>
82. <a class="button" href="frage.php">Frage</a>
83. <span> </span>
84. <a class="activebutton" href="#">Antwort</a>
85. </p>
86. <p style="float:right;"><a href="https://www.tillmenke.de/player/validate.php"></a
></p>
87. <p><br></p>
88.
89. <p class="first">Fehler: Es wurde keine Frage ausgewählt.</p>
90. </body>
91. </html>
92. <?php } ?>

```

4.1.3.7 antwortdetails.php

```

1. <?php
2. require_once("routen.php");
3. if(!isset($_GET["id"])){$_GET["id"]="none";}
4. if(isset($_fragen[$_GET["id"]])){
5. $frage = $_fragen[$_GET["id"]];
6. $description=$frage->getFahrtenTitles();
7. $antwort=$frage->getAntwort();
8. ?>
9. <!DOCTYPE HTML PUBLIC "-
//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
10. <html>

```

```

11. <head>
12. <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
13. <meta name="Generator" content="Microsoft Word 12 (filtered)">
14. <link rel="stylesheet" href="style.css" type="text/css">
15. <title>Mobilitätsfragen: <?php echo $frage->title; ?></title>
16. </head>
17.
18. <body>
19. <div class="title">
20. <p class="title">Mobilitätsfragen: <?php echo $frage-
>title; ?> <span class="titledate">Stand: <?php echo date("d.m.Y, H:i")." Uhr"; ?></
span></p>
21. </div>
22. <p class="navigation">
23. <a class="button" href="index.php">Nutzergruppe</a>
24. <span> </span>
25. <a class="button" href="frage.php">Frage</a>
26. <span> </span>
27. <a class="activebutton" href="#">Antwort (Fahrtdetails)</a>
28. </p>
29. <p style="float:right;"><a href="https://www.tillmenke.de/player/validate.php"></a
></p>
30. <p><br></p>
31.
32. <p class="first"><?php echo str_replace("\n","<br>",$antwort->text); ?></p>
33. <?php if($frage->art=="nurtext"){else{ ?>
34. <h1><?php echo $description["fahrtA"]; ?></h1>
35. <p><?php foreach($antwort-
>fahrtADetails as $detail){echo str_replace("\n","<br>",$detail)."<br>";} ?></p>
36.
37. <h1><?php echo $description["fahrtB"]; ?></h1>
38. <p><?php foreach($antwort-
>fahrtBDetails as $detail){echo str_replace("\n","<br>",$detail)."<br>";} ?></p>
39. <?php } ?>
40. <p class="fragenliste first">
41. <a href="antwortampel.php?id=<?php echo $_GET["id"]; ?>" class="button frage">Ich
möchte lieber wieder die Ampel sehen.</a><br>
42. <a href="index.php" class="button frage">Ich habe weitere Fragen.</a><!--
Link für bessere Erreichbarkeit auf Mini-Touchscreen auf Raspberry Pi-->
43. </p>
44. </body>
45. </html>
46. <?php
47. }else{
48. ?>
49. <!DOCTYPE HTML PUBLIC "-
//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
50. <html>
51. <head>
52. <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
53. <meta name="Generator" content="Microsoft Word 12 (filtered)">
54. <link rel="stylesheet" href="style.css" type="text/css">
55. <title>Mobilitätsfragen</title>
56. </head>
57.
58. <body>
59. <div class="title">
60. <p class="title">Frage</p>
61. </div>
62. <p class="navigation">
63. <a class="button" href="index.php">Nutzergruppe</a>
64. <span> </span>
65. <a class="button" href="frage.php">Frage</a>
66. <span> </span>
67. <a class="activebutton" href="#">Antwort</a>
68. </p>

```

```

69. <p style="float:right;"><a href="https://www.tillmenke.de/player/validate.php"></a></p>
70. <p><br></p>
71.
72. <p class="first">Fehler: Es wurde keine Frage ausgewählt.</p>
73. </body>
74. </html>
75. <?php } ?>

```

4.2 Konfiguration Raspberry Pi

Die folgenden Stichpunkte dokumentieren die für die Einrichtung des für die Präsentation verwendeten Raspberry Pi vorgenommenen Schritte. URLs verweisen auf den Seiten, deren Anleitung für den Konfigurationsschritt verwendet wurde.

Der Serverdienst läuft aber unabhängig davon auf jedem PHP-fähigen Webserver.

1. - Image von Joy-IT (für TouchScreen optimiert, Änderungen wie dort)
2. - Passwort User pi (Standard: Benutzer pi, Passwort raspberry)
3. - Passwort User root setzen (sudo passwd root)
4. -
root freischalten für ssh https://znll.net/index.php?title=Root_wieder_aktivieren/_Anmelden_als_Root_per_SSH_Ubuntu_14.04
5. - Apache2 und PHP5 installieren (sudo apt-get)
6. - wäre gut: Joe installieren (sudo apt-get)
7. - Windows-Remotedesktopverbindung freischalten <http://www.gieseke-buch.de/raspberrypi/vom-windows-pc-per-remotedesktop-den-raspberrypi-desktop-steuern>
8. apt-get purge realvnc-vnc-server
9. apt-get install xrdp
10. - Desktop-Hintergrund (Datei per SSH uploaden, dann über rechte Maustaste auf Desktop via Remotedesktop)
11. - CI-Schriftarten installieren <https://wiki.debian.org/Fonts>
12. Kopieren in /usr/local/share/fonts (mit Adminrechten)
13. - Chromium Kiosk-Modus als Standardstart (für Touchscreen) <https://webnlist.de/chromium-kiosk-modus-unter-raspbian-jessie/>
14. sudo nano /home/pi/.config/lxsession/LXDE-pi/autostart
15. Auskommentieren @xscreensaver -no-splash
16. neu
17. @xset s off
18. @xset s noblank
19. @xset -dpms
20. @chromium-browser --noerrdialogs --kiosk http://localhost/
21. Update 11.11.2017: ergänzt durch --disable-session-crashed-bubble, um Wiederherstellenfenster, dass sich auf dem kleinen Monitor nicht schließen lässt, zu verhindern, Hinweis von <https://www.administrator.de/frage/google-chrome-taskkill-seite-wiederherstellen-303503.html> auf <https://peter.sh/experiments/chromium-command-line-switches/>
22. zudem erforderlich: --disable-infobars <https://superuser.com/questions/237608/how-to-hide-chrome-warning-after-crash>
23. - DynDNS einrichten <https://raspberrypi.einsteiger.com/raspberrypi-einsteiger-guide-dyndns-teil-5/>
24. sudo apt-get install ddclient
25. Konfigurationsdialog folgen
26. Protokoll: DynDNS2 (danach funktioniert dyn.ns.tillmenke.de)
27. Benutzer: {entfernt}
28. Passwort: {entfernt}
29. Server: raspberrypi.heidelberg.tillmenke.de
30. Interface: use=cmd, cmd=find-local-ip <https://ascending.wordpress.com/2011/02/04/advanced-dynamic-dns-with-ddclient/>
31. mit find-local-ip=Datei mit Inhalt


```

32.          #!/bin/bash
33.          ifconfig eth0
34.          ifconfig wlan0
35.          ssl=yes
36.          Einrichten als Dämon
37.          Test: ddclient -daemon=0 -debug -verbose -noquiet 2 /etc/ddclient.conf
38.          sudo /etc/init.d/ddclient restart
39. - Auflösung einrichten (Joy-it-
  Anleitung für das Display): Auskommentieren entsprechender Zeilen in /boot/config.tx
  t (Standardmäßig ist 800x600 auskommentiert, für optimale Ausgabe am Fernseher 1920x
  1080 auskommentieren, für optimales Display am Gerät 320x240 auskommentieren)
40. - sudo apt-get install php5-curl

```

4.3 Beispielimplementierung Ampel

Die Beispielimplementierung nutzt die API von `antwortapi.php?id={id}&mode=ampellicht`, um daraus zu ermitteln, ob die LED an GPIO-PIN 3¹ (rot, API-Antwort 1), an GPIO-PIN 5 (gelb, API-Antwort 2) oder an GPIO-PIN 7 (grün, API-Antwort 3) mit Strom versorgt werden soll oder sämtliche LEDs ausbleiben sollen (API-Antwort 0). Die LEDs werden alle an GPIO-PIN 25 (Masse) angeschlossen.

Diese Funktion wurde in einem Python-Script umgesetzt, dass auf einem Raspberry Pi mit oben dargestellter Konfiguration (sowie von jedem anderen Raspberry-Pi-Standardbetriebssystem auf UNIX-Basis) problemlos über die Konsole aufgerufen oder (wie `chromium-browser` in der oben dargestellten Konfiguration) als Autostart konfiguriert werden kann.

4.3.1 mobilitaetsampel.py

```

1. # coding=windows-1252
2. print("#####")
3. print("### Projekt Mobilitätsfragen (C) 2017 Till Menke ###")
4. print("#####")
5. print("")
6.
7. fragenid=3;          # hier Fragen-ID wie GET-Parameter für antwortampel.php
8. sekundentowait=60;  # Abfrage erfolgt alle x Sekunden
9.
10. import time
11. import RPi.GPIO as GPIO
12. import urllib as urllib # für Python >3 erstes urllib durch urllib.request ersetzen
13. GPIO.setwarnings(False)
14. GPIO.cleanup()
15. GPIO.setmode(GPIO.BOARD)
16. GPIO.setup(3,GPIO.OUT) # ROT
17. GPIO.setup(5,GPIO.OUT) # GELB
18. GPIO.setup(7,GPIO.OUT) # GRÜN
19. ampellicht="0"
20. try:
21.     while True:
22.         page = urllib.urlopen('http://localhost/antwortapi.php?id='+str(fragenid)+'&mode
=ampellicht')
23.         ampellicht = page.read()
24.         # page = urllib.urlopen('http://localhost/antwortapi.php?id='+str(fragenid)+'&mode
=text')
25.         # text = page.read()
26.         if ampellicht == "0":
27.             GPIO.setup(3,GPIO.OUT) # ROT
28.             GPIO.setup(5,GPIO.OUT) # GELB
29.             GPIO.setup(7,GPIO.OUT) # GRÜN
30.             print(time.strftime("%d.%m.%Y, %H:%M Uhr")+": "+"Ampel aus")
31.         if ampellicht == "1":
32.             GPIO.setup(3,GPIO.LOW) # ROT

```

¹ Nummerierung von aus Richtung der langen Seite der Platine gesehen vorne (unten) links 1 bis hinten (oben) rechts 26.

```

33.     GPIO.setup(5,GPIO.OUT) # GELB
34.     GPIO.setup(7,GPIO.OUT) # GRÜN
35.     print(time.strftime("%d.%m.%Y, %H:%M Uhr")+": "+"Ampel Rot")
36.     if ampellicht == "2":
37.         GPIO.setup(3,GPIO.OUT) # ROT
38.         GPIO.setup(5,GPIO.LOW) # GELB
39.         GPIO.setup(7,GPIO.OUT) # GRÜN
40.         print(time.strftime("%d.%m.%Y, %H:%M Uhr")+": "+"Ampel Gelb")
41.     if ampellicht == "3":
42.         GPIO.setup(3,GPIO.OUT) # ROT
43.         GPIO.setup(5,GPIO.OUT) # GELB
44.         GPIO.setup(7,GPIO.LOW) # GRÜN
45.         print(time.strftime("%d.%m.%Y, %H:%M Uhr")+": "+"Ampel Grün")
46. #     print(text)
47.     print('Warte '+str(sekundentowait)+' Sekunden bis zur nächsten Aktualisierung')

48.     time.sleep(sekundentowait) # x Sekunden warten zwischen Anfragen
49. except KeyboardInterrupt:
50.     print("")
51.     print("Ampelaktualisierung abgebrochen")

```

4.3.2 Aufrufbefehl

```
sudo python mobilitaetsampel.py
```

4.4 bekannte Unzulänglichkeiten

4.4.1 Fahrt erreicht ihr Ziel nicht

Die TRIAS-API liefert auf eine der Dokumentation entsprechende Anfrage nach dem Linienweg nur Fehlermeldungen. Daher muss die Ankunftszeit über die Ankunftstafel der Zielhaltestelle ermittelt werden. Da die Fahrzeit nicht bekannt ist und der API-Linienfilter nicht funktioniert, können nur die etwa 40 nächsten Ankünfte an der Zielhaltestelle ab der Abfahrtszeit berücksichtigt werden. Insbesondere bei großen Haltestellen und längeren Fahrtwegen führt dies dazu, dass die Fahrt nicht mehr gefunden werden kann und damit zurückgemeldet wird, dass die Fahrt ihr Ziel nicht erreicht.

Die Meldung, dass die Fahrt ihr Ziel nicht erreicht, ist damit auch möglich, wenn die entsprechende Fahrt tatsächlich ihr Ziel erreicht, dies aber auf der Ankunftstafel nicht aufgeführt wird.

In der Praxis wird dies jedoch meist nur passieren, wenn die entsprechende Fahrt ohnehin länger dauern würde oder die Fahrt noch zu lange hin ist, um präzise Aussagen zu treffen.

4.4.2 Linie 721

Die TRIAS-API liefert derzeit, anders als die VRN-Webseite, keine Echtzeitdaten über die Linie 721. Alle Verbindungen unter Beteiligung der Linie 721 führen daher zur Meldung, dass mindestens eine der überprüften Verbindungen nicht auf Echtzeitdaten basiert.

4.4.3 Weiterfahrtalternativen

Obwohl in der Typenstruktur vorgesehen werden keine Weiterfahrtalternativen ausgewertet. Grund hierfür ist, dass mit zunehmender Verzweigung die Zahl der nötigen API-Anfragen und der benötigte Rechenaufwand stärker als linear steigt. Zudem werden für die hier vorgestellten Beispiele wie wohl auch für die meisten anderen denkbaren Anwendungen keine Weiterfahrtalternativen benötigt.