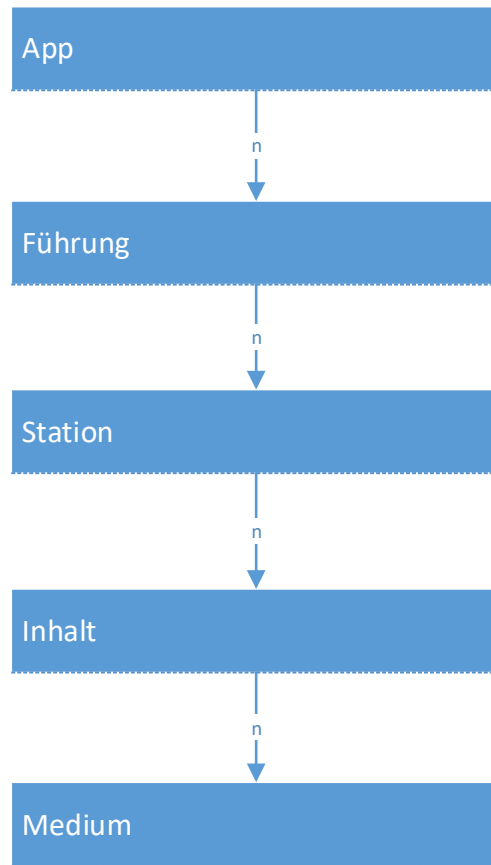


Inhalt

Schnittstellendefinition	2
Statisches Appdesign.....	3
Basiskonfiguration.....	4
Inhalte der Führungen.....	6
Dateisystem	6
JSON-Dateien	8
How-To	9

Schnittstellendefinition

Das System bietet 3 Schnittstellen zur Pflege der Inhalte und Konfigurationen an. Im Folgenden werden die drei Schnittstellen beschrieben, anschließend erläutert eine Anleitung die spezifische Umsetzung bestimmter Use-Cases. Die Datenstruktur ist hierarchisch aufgebaut:



Die Entität App wird mit der Schnittstelle „Statisches Appdesign“ und „Basiskonfiguration der Anwendung“ konfiguriert. Die Entitäten Führung, Station, Inhalt und Medium werden mit der Schnittstelle „Inhalte der Führungen“ definiert.

1. Statische Appdesigns

Die Visualisierung einer Anwendung definiert sich durch die verwendeten Farben, Icons, Schriften und Größen sowie durch die Anordnung der View-Elemente in einem Layout. Diese statischen Parameter werden vor dem Bauen der Anwendung für die jeweilige native Anwendung definiert. Hier wird inhaltsunabhängig das Branding der Anwendung definiert.

2. Basiskonfiguration der Anwendung

Beim ersten Appstart werden die ersten Basis-Inhalte in die Datenbank der App geladen. Diese Inhalte definieren, welche Inhaltstypen es geben kann, wie diese visualisiert werden sollen und welche Tipps es zu welcher Funktion geben soll. Die AR-Stadtführung besteht aus mehreren Führungsvarianten die jeweils aus einer Menge von Stationen besteht. Jede Führungsvariante muss in der Basiskonfiguration definiert werden.

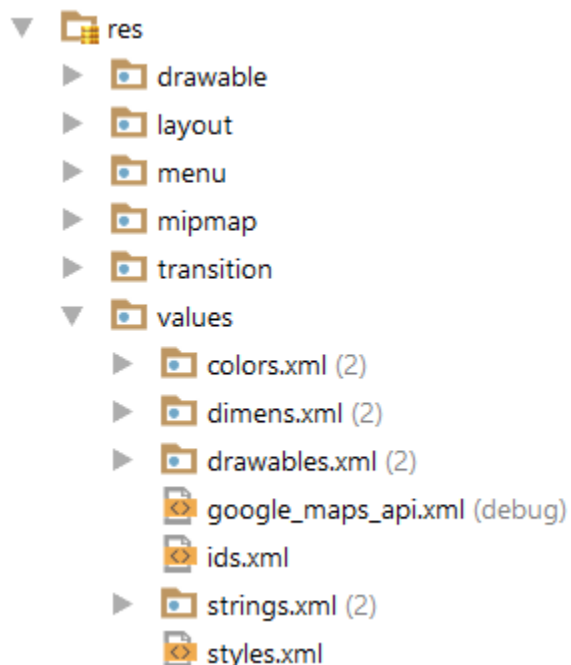
3. Inhalte der Führungen

Es können beliebig viele Führungen für eine AR-Stadtführungs-App definiert werden. Alle Führungen werden definiert durch eine Menge an Stationen. Jede Station besteht aus mehreren beschreibenden Attributen sowie einer Menge an eigenen Inhalten. Ein Inhalt kann einer der in der Basiskonfiguration definierten Inhaltstypen sein und beinhaltet wiederum eine Menge von Media. Ein Medium repräsentiert immer genau eine Datei und wird durch weitere Attribute beschrieben.

Statisches Appdesign

Das Appdesign ist abhängig von definierten Farben, Schriften, Icons und Styles. Diese sind vor dem Bauen des Systems zu definieren. Die Schnittstelle des Appdesigns ist abhängig von der Plattform, für die entwickelt wird.

Android:



Für native Android-Anwendungen dienen XML-Dateien im Ressourcen-Ordner zum Aufbau des App-Designs. Als Schnittstellen dienen etwa die „strings.xml“ zur Definition der (mehrsprachigen) Beschriftungen und die „dimens.xml“ zur Definition der Abstände und (Schrift-) Größen

der einzelnen View-Elemente. In der „colors.xml“ werden alle genutzten Farben für das Projekt definiert. In der Ressource-Datei „drawables.xml“ werden hinterlegte Bilddateien als Icons bestimmten View-Elementen zugewiesen. Die entsprechenden Bilddateien werden in verschiedenen Auflösungsstufen für jedes Display optimiert in dem Ressource-Ordner „drawable“ hinterlegt.

Grundsätzlich werden alle Views als „layout.xml“ in dem Ressource-Ordner „layout“ definiert. Hier wird für jedes View-Element auf die definierten Abstände, Bilddateien, Schriften und Farben aus den genannten Resources-Dateien verwiesen. Im Normalfall müssen Layout-Dateien nicht angerührt werden, solange die grundsätzlichen Strukturen der Views unverändert bleiben.

Basiskonfiguration

Die App hat initiale Basisinhalte. Dies sind die Menge aller Stadtführungen der App mit ihren Attributen sowie den einzelnen Hilfen und Basis-Konfigurationen. Diese Inhalte sind vor dem Bauen des Systems im Bundle des Projekts als Assets in Form von JSON-Dateien zu hinterlegen. Beim ersten Appstart wird aus ihnen eine Datenbank gebaut. Die Struktur der JSON-Dateien der Schnittstelle orientiert sich somit an der Struktur der Datenbank. Jede Datei, die in der Datenbank repräsentiert wird, wird durch folgende 3 Attribute definiert:

Medium
localPath
title
description

Die einzelnen Entitäten der Datenbank halten je eine Menge an Verweisen auf die benötigten Dateien.

Die Basis-Konfiguration besteht aus folgenden JSON-Dateien:

„Base.config“ beinhaltet ein Array aus „ContentTypes“. Diese definieren, welche Typen von Inhalten zu einer Station in der Stadtführung dargestellt werden können. Jeder ContentType wird über eine Menge an Attributen

beschrieben und kann zudem auf eine weitere JSON-Datei verweisen, die eine Menge von Hilfen für den jeweiligen Typ anbietet.

```
{
    "name": "WIKITUDE_AUGMENTATION",
    "fullscreen":true,
    "autostart":true,
    "tipps": "tipps/arTipps.json"
}
```

Die JSON-Datei zur Definition der Tipps beinhaltet eine Menge von Medien (s.o.) die über die gegebenen 3 Attribute definiert werden.

```
[
    {
        "path" : "tipps/tipps_0.png",
        "title" : "Schauen Sie frontal auf das
Objekt?",
        "description" : "Die Erkennung funktioniert
am besten, wenn Sie frontal auf das Objekt schauen.
Ändern Sie gegebenenfalls ihre Position."
    },

```

In der Datei „allTours.json“ werden alle Führungen der Anwendung in einem Array definiert.

```
[
    {
        "name":"Stadtführung Wiedenbrück",
        "description" : "Dies ist die Stadtführung
durch den historischen Stadtkern von Wiedenbrück.",
        "centerOfMapLat" : 51.833771,
        "centerOfMapLong" : 8.309984,
        "thumbnail" : "1/thumbnail.png",
        "extensionPackage" : 1,
        "introductions" :
"introductions/introduction.json",
        "baseAugmentation" :
"1/baseAugmentation/augmentation.json",
        "tipps": "tipps/tipps.json"
    },

```

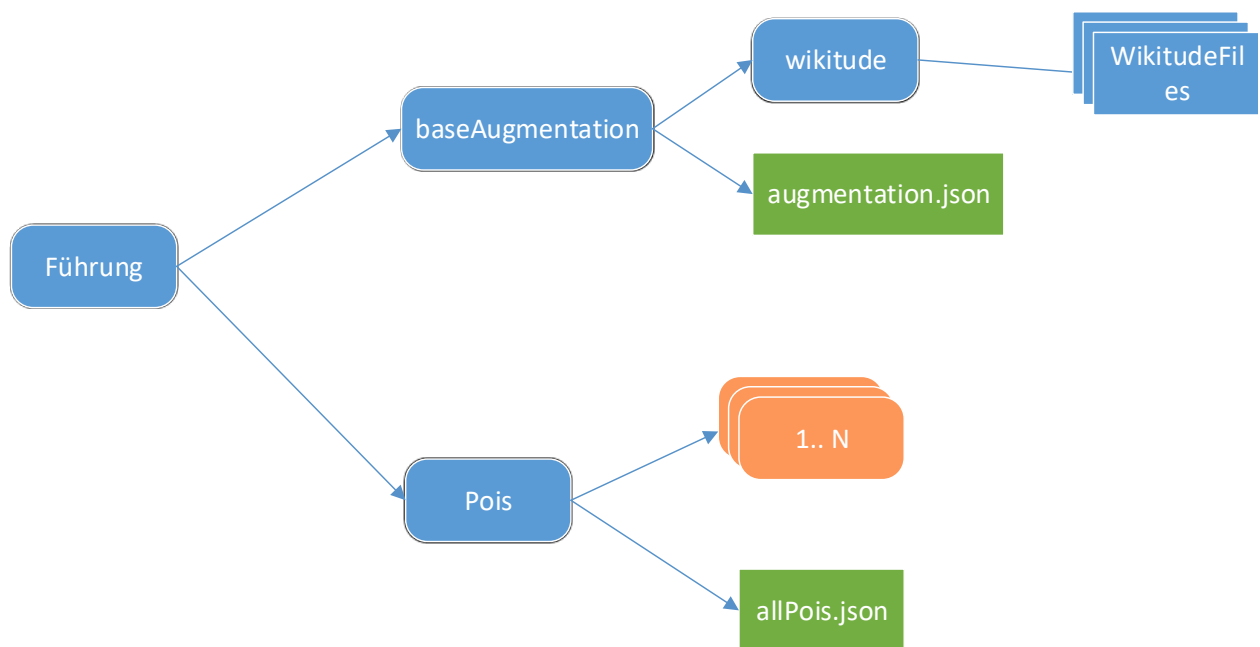
Die referenzierten JSON-Dateien „tipps“, „baseAugmentation“ und „introductions“ sind alle jeweils eine Menge an Medien, die entsprechend aufgebaut sind (s.o.). Alle in der Datenbank repräsentierten Dateien werden somit durch die Attribute „Path“, „name“ und „description“ beschrieben.

Inhalte der Führungen

Inhalte der Führung werden durch JSON-Dateien beschrieben und in Form von Dateien in einer Dateistruktur gepflegt. Eine vollständige Führung wird anschließend komprimiert auf einem Server für den Client zum Download angeboten.

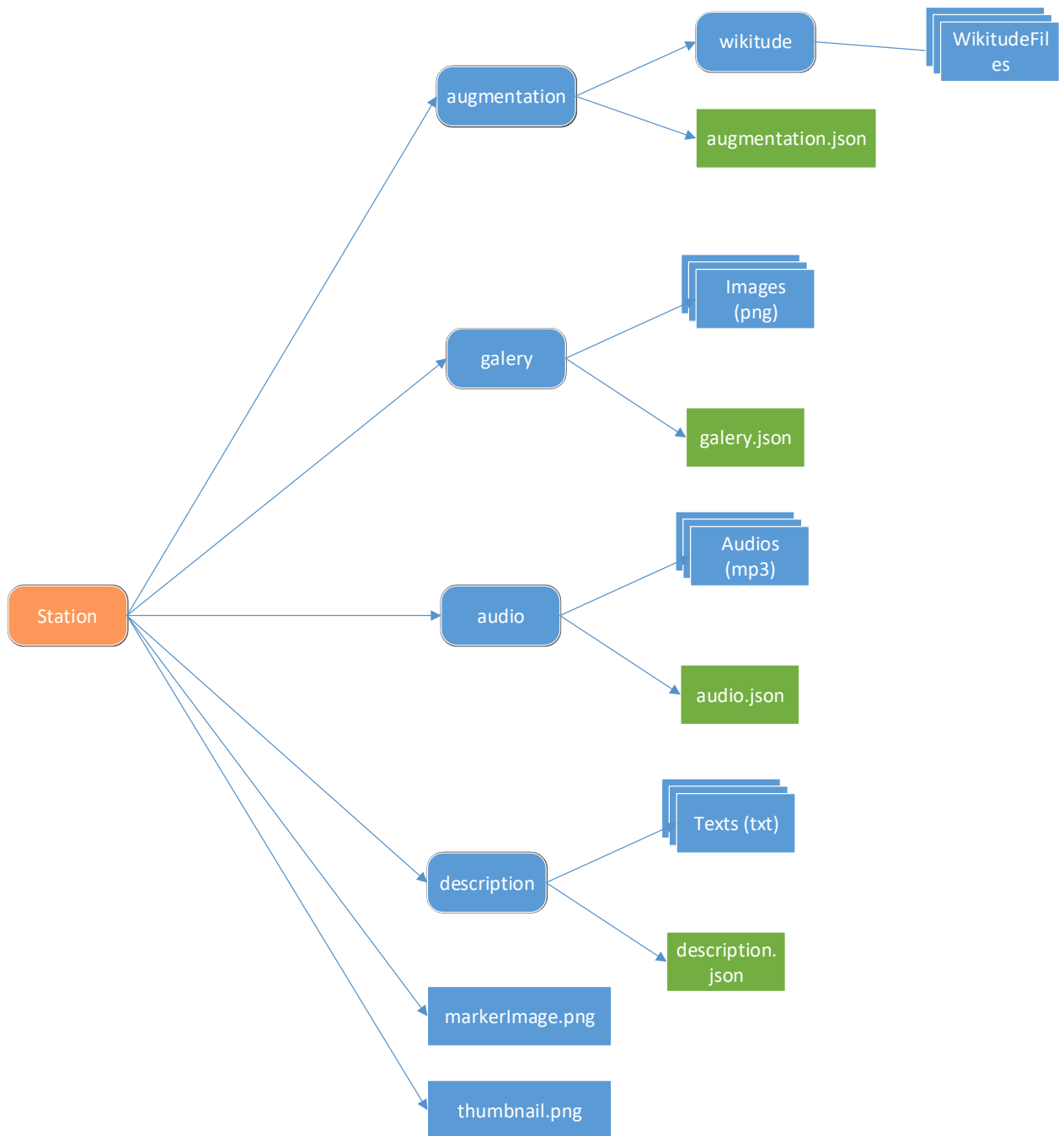
Dateisystem

Das Dateisystem einer Führung ist wie folgt aufgebaut:



Eine Führung hat die Unterordner „baseAugmentation“ und „Pois“. In den Ordner „baseAugmentation“ werden die Projektdateien gelegt, die für das initiale Tracking und die Erkennung der Stationen der Führung nötig sind. Abhängig von dem genutzten AR-Framework kann sich dies ändern. Konfiguriert wird die BaseAugmentation in einer zugehörigen JSON-Datei. Alle Stationen einer Führung werden in dem Unterordner „Pois“ gesammelt. In „Pois“ gibt es für jede Station einer Führung einen Ordner. Dieser sammelt alle Medien die als Inhalte für die Station vorbereitet wurden. In „allPois.json“ werden alle Stationen einzeln benannt und mit Attributen konfiguriert.

Die Ordner jeder Station ist wie folgt aufgebaut:



Jede Station hat für jeden Inhaltstypen einen eigenen Unterordner. Die dargestellte Struktur gilt für die explizite Anwendung der Stadtführung Rheda-Wiedenbrück mit den Inhaltstypen „augmentation“, „galery“, „audio“ und „description“. Jeder Unterordner ist gleich aufgebaut. Er besteht aus einer Menge von Dateien, die diesen Inhalt darstellen (Bilddateien für die Galerie, Audiodateien für die Audios, Text-Dateien für die Beschreibungstexte, Wikitude-Dateien für die augmentation). Außerdem gibt

es in jedem Unterordner eine JSON-Datei zur Beschreibung der jeweiligen Inhalte.

JSON-Dateien

1. allPois.json

Die AllPois.json sammelt alle Stationen der Führung als JSON-Objekte in einem JSON-Array. Jedes Poi hat folgende Attribute:

Name	Name der Station (z.b. Adresse)	
Description	Kurzbeschreibung der Station. Maximal 85 Zeichen	
LocationMapLat	Latitude der Station als Double	
LocationMapLong	Longitude der Station als Double	
Thumbnail	Verweis auf den Pfad zu dem Vorschaubild	
markerImage	Verweis auf den Pfad zu dem Bild des Markerbereichs	
markerIds	JSON-Array aus Strings: Die ids der Marker, die zu der Station im Tracker eingebunden sind.	
contents	JSON-Array aus JSON-Objekten, die jeweils 3 Attribute haben:	
	name	Name des Contents
	contentType	Type des Contents, wie in der baseConfig.json definiert.
	mediaPath	Pfad zu der JSON-Datei die alle Files für diesen Content sammelt.

Beispielhafte „allPois.json“ mit einer Station:

```
allPois: [
{
  "name" : "In der Halle 2",
  "description" : "Das Hemmelmannsches Haus, 1567 erbaut. Mit Darstellungen der menschlichen Untugenden.",
  "locationMapLat" : 51.833771,
  "locationMapLong" : 8.309984,
  "thumbnail" : "1/Pois/1/thumbnail.png",
  "markerImage" : "1/Pois/1/markerImage.png",
  "markerIds" : [
    "marker1",
    "marker2"
  ],
  "contents" : [
    {
      "name" : "Sprecher",
      "contentType" : "AUDIO",
      "mediaPath" : "1/Pois/1/audio/audio.json"
    }
  ]
},
]
```



```

    {
        "name" : "Beschreibungstext",
        "contentType" : "TEXT",
        "mediaPath":
"1/Pois/1/description/description.json"
    },
    {
        "name" : "Galery",
        "contentType" : "GALLERY",
        "mediaPath": "1/Pois/1/galery/galery.json"
    },
    {
        "name" : "3dAnimation",
        "contentType": "WIKITUDE_AUGMENTATION",
        "mediaPath":
"1/Pois/1/augmentation/augmentation.json"
    }
]
}, {...}Hier kann man weitere Stationen hinzufügen
]

```

2. description.json, galery.json, augmentation.json, audio.json

Die config-Dateien aller Inhaltstypen sind gleich aufgebaut. Alle bestehen aus einem JSON-Array, das alle Media sammelt, die zu dem jeweiligen Inhalt gehören. Die Media sind durch folgende 3 Attribute beschrieben:

Title	Name des Medium
Description	Beschreibung des Medium
Path	Der Pfad zu der zugehörigen Datei

Beispielhaft sieht eine entsprechende JSON-Datei für einen einzelnen Beschreibungstext für eine Station wie folgt aus:

```

[
    {
        "path" : "1/Pois/2/description.txt",
        "title" : "Beschreibungstext",
        "description" : "Beschreibungstext zu der Station"
    }, {...}Hier kann man weitere Beschreibungs-Text-Dateien
    hinzufügen
]

```

How-To

In diesem Kapitel wird für explizite Use-Cases beschrieben, was zu tun ist, um den jeweiligen Use-Case durchzuführen.

Use Case	Schritte:
Neue Führung zu App hinzufügen	<ol style="list-style-type: none"> 1. Einen Neuen Führungs-Ordner im Root anlegen 2. In dem Führungs-Ordner eine Bilddatei „Thumbnail“ anlegen 3. In der JSON-Datei „allTours.json“ ein neues JSON-Objekt in das Array einfügen.
Neue Hilfe zu Führung hinzufügen	<ol style="list-style-type: none"> 1. Lege in den Ordner „/help/“ eine neue JSON-Datei „tourHelp.json“. 2. Schreibe in die JSON-Datei „tourHelp.json“ ein JSON-Array mit allen Hilfe-Elementen. 3. Füge für jedes Hilfe-Element eine Bilddatei in dem Ordner „/help/“ hinzu und schreibe je einen Eintrag in die tourHelp.json, mit den drei Medium-Attributen (title, description, path). 4. Verweise in „allTours.json“ bei dem Attribut „tipps“ auf den Pfad der „help/tourHelp.json“
Neue Einführung zu Führung hinzufügen	Wie „Neue Hilfe zur Führung hinzufügen“ mit dem path „/introduction/“
Neue Basis-Augmentation zu Führung hinzufügen	<ol style="list-style-type: none"> 1. Lege den Ordner „/baseAugmentation/“ an. 2. Lege in dem Ordner die JSON-Datei „augmentation.json“ an. 3. Kopiere das fertige Wikitude-Projekt mit allen Assets usw. 4. Schreibe in die Datei „augmentation.json“ ein JSON-Array mit nur einem Element, das auf die „index.html“ des Wikitude-Projekts verweist. Das Element hat die

	Attribute eines Mediums (title, description, path)
Neue Station zu Führung hinzufügen	<ol style="list-style-type: none"> 1. Wenn es den Unterordner „/pois/“ noch nicht gibt, legen den Ordner an. 2. Wenn es die Datei „allPois.json“ noch nicht in dem Ordner „/pois/“ gibt, lege sie an. 3. Lege einen Ordner für jede Station an. 4. Lege eine Bilddatei „Thumbnail.png“ und eine Bilddatei „MarkerImage.png“ in den Unterordner der Station. 5. Schreibe einen neuen Eintrag für die Station in die JSON „allPois.json“. Der Eintrag muss die Attribute name, description, locationMapLat, locationMapLong, thumbnail, markerImage, markerIds und contents haben.
Neuen Beschreibungstext zu Station hinzufügen	<ol style="list-style-type: none"> 1. Lege einen Ordner „description“ in dem Unterordner der Station an. 2. Lege eine Textdatei „description.txt“ in den Ordner „description“. 3. Schreibe den Beschreibungstext (mit Absätzen etc.) in die „description.txt“. Achte darauf, dass die Datei mit UTF-8 codiert ist. 4. Lege eine JSON-Datei „description.json“ an, mit einem JSON-Array das aus einem Element besteht. Das Element verweist auf die description.txt und hat die Attribute title, description, path. 5. Füge in der JSON-Datei „allPois.json“ bei der entsprechenden Station dem Attribut Contents ein neues Element

	<p>hinzu, das mit den Attributen name, contentType und path beschrieben wird. Setze contentType auf „TEXT“ und path auf den Pfad zu der „description.json“</p>
Neue Audio zu Station hinzufügen	<ol style="list-style-type: none"> 1. Lege einen Ordner „audio“ in dem Unterordner der Station an. 2. Lege eine Audiodatei „audio.mp3“ in den Ordner „audio“. 3. Lege eine JSON-Datei „audio.json“ an, mit einem JSON-Array das aus einem Element besteht. Das Element verweist auf die „audio.mp3“ und hat die Attribute title, description, path. 4. Füge in der JSON-Datei „allPois.json“ bei der entsprechenden Station dem Attribut Contents ein neues Element hinzu, das mit den Attributen name, contentType und path beschrieben wird. Setze contentType auf „AUDIO“ und path auf den Pfad zu der „audio.json“.
Neue Augmentation zu Station hinzufügen	<ol style="list-style-type: none"> 1. Lege einen Ordner „augmentation“ in dem Unterordner der Station an. 2. Lege das vollständige Wikitude-Projekt in den Ordner. 3. Lege eine JSON-Datei „augmentation.json“ an, mit einem JSON-Array das aus einem Element besteht. Das Element verweist auf die index.html und hat die Attribute title, description, path. 4. Füge in der JSON-Datei „allPois.json“ bei der entsprechenden Station dem Attribut Contents ein neues Element hinzu, das mit den Attributen name,

	<p>contentType und path beschrieben wird. Setze contentType auf „WIKITUDE_AUGMENTATION“ und path auf den Pfad zu der „augmentation.json“</p>
<p>Neue Bildergalerie zu Station hinzufügen</p>	<ol style="list-style-type: none"> 1. Lege einen Ordner „galery“ in dem Unterordner der Station an. 2. Lege alle Bilder der Galerie in den Ordner. 3. Lege eine JSON-Datei „galery.json“ an, mit einem JSON-Array. 4. Füge zu dem JSON-Array in „galery.json“ je ein Element für jedes Bild der Galerie an. Jedes Element verweist auf eine Bilddatei der Galerie und hat die Attribute title, description, path. 5. Füge in der JSON-Datei „allPois.json“ bei der entsprechenden Station dem Attribut Contents ein neues Element hinzu, das mit den Attributen name, contentType und path beschrieben wird. Setze contentType auf „GALLERY“ und path auf den Pfad zu der „galery.json“