AlpineBits
Open Data Exchange in the Alpine tourism

AlpineBits Developer's Getting Started Guide

This document is a quick getting started guideline for ***developers*** *that are implementing the AlpineBits interface.*

Context

- make sure you understand what AlpineBits **is** (a definition of an interface for the exchange of touristic data based on **OTA**) and what AlpineBits **is not** (a particular service or database or software).
- get the 2014-04 AlpineBits release kit!
    - dl from http://download.alpinebits.org/AlpineBits_2014-04.zip
    - the latest release is the one from 2014-12-23 (see pdf)
    - contains the document (59 page PDF), example files and schema files
- understand what we mean when we say **client** (the software than initiate the comunication with a request) and **server** (the software that answers with a response)
- understand what kind of information can be transmitted - for this workshop consider the parts "FreeRooms" and "GuestRequests" in particular

Getting started with the docs

- read chapter 1 :)
- read chapter 2
- take your time, do not underestimate this
- think about error handling and logging from the beginning
- AlpineBits uses HTTPS (POST requests with multipart-form data)
- server guys: think about getting a "officially" signed certificate soon to test with different clients

- client guys: make sure your client accepts CAs used by the servers
- authentication is HTTP basic auth
- what platform do you use? How does your platform handle this (for the client case / for the server case)?
  - scripting languages (e.g. PHP) can make use of **libcurl** to do this
  - .NET should have a HTTPClient class that can handle this and one AlpineBits member has written this software and this software you might find useful
  - etc...
- read chapter 3: the so called house keeping actions
- do a test setup, e.g. sending getVersion
- here is a very simple PHP sketch to run a client/server in PHP

FreeRooms and GuestRequests

- decide what feature you need
- have a quick look to the first paragraphs of chapters 4.1 and 4.2

XML

- the actual "payload" is XML
- if you didn't already: get to know XML and related concepts/technologies (well-formed documents, validation, ...)
- you need to parse XML, every platform supports this with different strategies / APIs: i.e. SAX, DOM, you might (or might not) want to use XPath
- how does you database look? existing schema? new schema? ORM? native XML

OTA / AlpineBits Schemas

- the AlpineBits schema is a subset of the OTA schema (Open Travel Alliance), currently we use OTA 2010A
- OTA is large and vast: this tool is handy to browse the OTA schema
- validation: how can you validate on your platform (e.g. xmllint under Linux)?
- get the OTA XSD files (from the OTA site) and the AlpineBits schema (from the ALpineBits kit), AlpineBits is a strict subschema of OTA, during test you should validate against both (this way we can trap errors in our schema), during production you might want to validate only against AlpineBits
- besides, XSD, we also provide a schema in another language (RelaxNG), you might wish to use
- TIS provides an online validator at http://alpinebits.testingmachine.eu/validator you can use if you don't have any tools handy (please do not use this tool in an automated way in production, because it is not running on a server that can manage high traffic)

FreeRooms

- read chap 4.1
- pay attention to the responses: we have 3 levels: Errors (data is bad, server cannot process request, tx failed) / Warnings (data is good, but server cannot process

request, e.g. data too far in the future, tx failed) + Advisories (data good, processing OK, but there is some nagging)

- we use the OTA code tables (you can find the XLS file in the OTA kit) for errors/warnings/advisories, please use a human readable description that helps debugging
- when you test, please do not use real-world data to avoid mixups
- AlpineBits lets you use deltas vs. full syncs, but some partners require deltas and will get upset if you send the full sync at each exchange

GuestRequests

- read chap 4.2
- note that GuestRequests can be: "Anfragen", "Buchungen", "Stornos", not only "Anfragen"
- this particular payload has "4" message types: client request, server response, client ack, server response (success)
- note the use of SelectionCriteria is meant to be used as a check to be send sporadically or recover from data lost at the client, but also note the server ist not required to keep records forever

General info

- each chapter contains a "implementation tips" section, that is useful for implementators
- we use this forum / mailing list for our discussions