

Gallagher HR4 & HR5 Hand Held EID Tag Reader and Data Collector



Animal Data Interface (ADI) Integration Reference Document

Date: 11th August 2015
Version: 1.5

1.	INTRODUCTION	3
1.1.	Disclaimer	3
1.2.	About this document	3
1.3.	Intended Audience	3
1.4.	Related Resources	4
1.5.	Document Amendment Register	4
2.	HR4 / HR5 DATA TRANSFER	5
2.1.	Connecting to the reader via USB	5
2.2.	Connecting to the reader via Bluetooth	5
2.3.	Data transfer mode	5
3.	HTTP REQUESTS AND RESPONSES	6
3.1.	Request format	6
3.2.	Request for Animal Resource:	7
3.3.	Request for Traits Resource:	8
3.4.	Request for Treatments Resource:	8
3.5.	Request for Session Resource:	9
3.6.	Response format	10
4.	DEVICE CONSIDERATIONS	11
4.1.	Device Serial Number	11
4.2.	Device Configuration	11
4.3.	Draftlist / Sortlist	11
4.4.	Partial Success	12
4.5.	'PUT' vs. 'POST /merge'	12
4.6.	Mothering Session	12
4.7.	Device Time	12
4.8.	Treatment and Activity	12
4.9.	Deleting of an Animal	12
4.10.	Supported Characters	13
4.11.	Animal Notes created time	13
4.12.	Reserved Names of Traits, Life Data and Activities	13
4.13.	Updating animals and its life data	14
4.14.	Limits	14
4.15.	Changing draft gate names	15
4.16.	Namespace changes	15
4.17.	Animal Alerts	15
5.	KNOWN ISSUES AND LIMITATIONS	16
5.1.	Response on Invalid XML or Invalid Data	16
5.2.	Reader cannot recover after a read timeout.	16
5.3.	PUT sessions currently does not work.	16

1. INTRODUCTION

1.1. Disclaimer

This document gives certain information about products and/or services provided by Gallagher Group Limited or its related companies (referred to as “Gallagher Group”).

The information is indicative only and is subject to change without notice meaning it may be out of date at any given time. Although every effort has been taken to ensure the quality and accuracy of the information, Gallagher Group makes no representation as to its accuracy or completeness and it should not be relied on as such. To the extent permitted by law, all express or implied, or other representations or warranties in relation to the information are expressly excluded.

Neither Gallagher Group or any of its directors, employees or other representatives shall be responsible for any loss that you may incur, either directly or indirectly, arising from any use or decisions based on the information provided.

Except where stated otherwise, the information is subject to copyright owned by Gallagher Group and you may not sell it without permission. Gallagher Group is the owner of all trademarks reproduced in this information. All trademarks which are not the property of Gallagher Group, are acknowledged.

Copyright © Gallagher Group Ltd 2015. All rights reserved.

1.2. About this document

This document illustrates guidelines for 3rd party Animal Data Interface application development and should be read in conjunction with REST API document (REF2).

This document also summarises a list of supported REST APIs in the latest HR4/5 release (version vHH1.04.xxx) and known limitations.

1.3. Intended Audience

This document is intended for any interested third party, to use as developer documentation, who intends to integrate their system with another system using the REST interface to transfer the Animal Data XML schema.

1.4. Related Resources

Ref	Document Title	Version	Issue Date	Author
REF1	Animal Data Interface Schema document	1.1	17 Mar 2013	Gallagher Group Ltd, Tru-Test Group Ltd
REF2	Animal Data Interface Schema (XSD)	1.1	17 Mar 2013	Gallagher Group Ltd, Tru-Test Group Ltd
REF3	Animal Data Interface REST API.	1.0		Gallagher Group Ltd, Tru-Test Group Ltd
REF4	Chunked Encoding: http://en.wikipedia.org/wiki/Chunked_transfer_encoding			

1.5. Document Amendment Register

Version	Issue Date	Details of Change	Contributors
1.0	6 th December 2013	Initial draft.	
1.1	3 rd June 2015	Note about max chunk length for receive.	
1.2	24 th June 2015	Adding Disclaimer and Copyright	
1.3	8 th July 2015	Updating namespace and changing details around PUT /session	
1.4	29 th July 2015	Updating Disclaimer and Copyright	
1.5	11 th August 2015	Updating connection details, draft names and other feedback	

2. HR4 / HR5 DATA TRANSFER

The Gallagher HR4 and HR5 hand held EID tag readers (hereafter known as reader) uses the HTTP protocol to send and receive data during a transfer. The data payload format is in XML and uses the Animal Data Interface Schema (REF1). Data transfer can be done through USB, Bluetooth.

2.1. Connecting to the reader via USB

For USB

- Baud rate: 115200 bps
- Data bits: 8
- Stop bits: 1
- Parity bit: No Parity
- Hardware Flow control: Enabled (RTS CTS)

The reader's USB port uses the DTR signal to detect that the USB port has been opened. The DTR line must be enabled in COM port settings.

2.2. Connecting to the reader via Bluetooth

Connecting via Bluetooth should be possible via any Bluetooth enabled device or Bluetooth dongle. The passcode for the device is 0000.

2.3. Data transfer mode

Once connected to the reader, the reader must be switched into '*data transfer mode*' before any xml can be exchanged. The reader does not support multiple port data transfer simultaneously.

For example, if the reader is already in data transfer mode through USB interface it will deny any http request coming from the Bluetooth interface with a reply of '*HTTP/1.1 503 Service Unavailable*'.

To set the reader to data transfer mode, send the following request:

```
POST /config/datatransfer?value=on HTTP/1.1
```

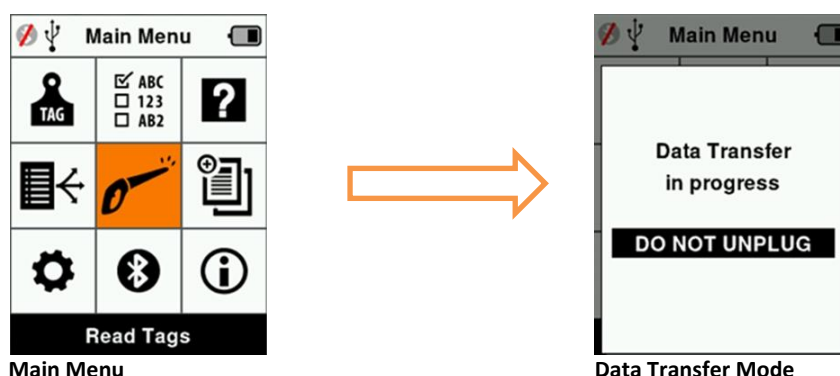
The reader will reply "HTTP/1.1 200 Ok" if it succeeds.

To get out of data transfer mode, send the following request:

```
POST /config/datatransfer?value=off HTTP/1.1
```

The reader will reply "HTTP/1.1 200 Ok" if it succeeds.

Note: Some screens that cannot go into data transfer mode. This is because pending edits of traits or uncancellable operations can block going to the main menu screen where data transfer takes place. If this takes place then the reader will respond with "HTTP/1.1 503 Service Unavailable" instead.



Once in data transfer mode you may begin sending HTTP requests to the reader. The connection to the reader is persistent.

3. HTTP REQUESTS AND RESPONSES

3.1. Request format

The reader supports the HTTP methods *GET*, *POST*, *PUT* and *DELETE* (REF3).

HTTP Method	Description
GET	returns a resource (a collection or specific item)
POST	creates new resource
PUT	resource is to be replaced
DELETE	deletes a resource

```
{method} {URI}{optional parameters} HTTP/1.1
{headers}

{XML data payload if required by method}
```

Where, the optional parameters format is:

```
?param1=value1&param2=value2...
```

Currently, the reader only supports a single parameter.

The supported data transfer mechanism is ‘*chunked transfer encoding*’ (REF4). This uses the header ‘*Transfer-Encoding: chunked*’.

Note: The reader only supports receiving chunk lengths of up to 255 bytes (FF).

If it receives more than 255 bytes in a single chunk, the transfer will end unexpectedly.

The reader does not have this limit when sending responses and often responses with chunks longer than this.

The end of the payload must be followed by two carriage returns and a line feed character. Below is an example of *PUT* treatment request with type ‘*5 in 1 Vacc*’.

```
PUT /treatments/5%20in%201%20Vacc HTTP/1.1
Transfer-Encoding: chunked

FF
<?XML version="1.0" encoding="utf-8"?><ads:body
XMLns:ads="http://animaldatainterface.org/Schema/2014/ADI"><ads:header><ads:createdBy>Gallagher
Group Ltd</ads:createdBy><ads:dateTimeCreated>2013-11-
26T09:34:28.9464557+13:00</ads:dateTimeCreated></ads:head
6F
er><ads:treatments><ads:treatment><ads:type>5 in 1
Vacc</ads:type></ads:treatment></ads:treatments></ads:body>
0
```

Another example is to get an XML file of all the animals in session with id 4:

```
GET /animals?sessionId=4 HTTP/1.1
```

3.2. Request for Animal Resource:

URI	Method	Parameters
/animals/{idType}/{id}	DELETE	
/animals	GET	
/animals	GET	sessionId
/animals	GET	count
/animals/{idType}/{id}	GET	
/animals	POST	
/merge/animals	POST	
/animals	PUT	

Parameter definitions:

- **{idType}:** valid entries are *'eid'* (electronic ID) and *'tag'* (visual ID tag).
The *'eid'* is 15 characters and sent as country code (3-characters) + tag data (12-characters).
- **sessionId:** the numeric identifier generated by the reader whenever it creates a session. This parameter is used when you want to retrieve all animals that belong to a particular session.
- **count:** parameter accepts either *'yes'* or *'no'* value.
Default value is *"no"*.
When an invalid value is passed, the default value is kept.

NOTE: The URI *'/merge/animals'* is used to update the resource. Please refer to Animal Data Interface REST API (REF3) for more details on the *"/merge/"* URI.

Example of animal related HTTP request:

GET /animals?count=yes HTTP/1.1

- returns the total number animal in the reader

GET /animals/tag/TAG53 HTTP/1.1

- returns the animal details of an animal with *"tag"* TAG53

GET /animals/EID/554000000000123 HTTP/1.1

- returns the animal details of an animal with *"eid"* 554 000000000123

3.3. Request for Traits Resource:

URI	Method	Parameters
/traits	DELETE	
/traits/{name}	DELETE	
/traits	GET	
/traits/{name}	GET	
/traits	POST	
/merge/traits	POST	
/traits/{name}	PUT	

NOTE:

- *{name}* in URI is restricted to reader valid characters only.
- a trait resource refers to session traits and animal life data.

Example of Trait related HTTP request:

GET /traits HTTP/1.1

- return all traits in the reader

GET /traits/breed HTTP/1.1

- returns only the trait named breed if exist on the reader

3.4. Request for Treatments Resource:

URI	Method	Parameters
/treatments/{type}	DELETE	
/treatments	GET	
/treatments/{type}	GET	
/treatments	POST	
/treatments/{type}	PUT	

Parameter definitions:

- *{type}*: refers to the name of the treatment, e.g. *Drenched*, *Weaned*, etc.

NOTE:

- treatment is a List of Value trait that has either an item value of 'yes' or 'no'.

Example of Treatment related HTTP request:

GET /treatments HTTP/1.1

- returns all treatments in the reader

GET / treatments /tagged HTTP/1.1

- returns the treatment "tagged" in the reader if exist

3.5. Request for Session Resource:

URI	Method	Parameters
/sessions/{id}	DELETE	
/sessions	GET	
/sessions	GET	count
/sessions/{id}	GET	
/sessions/{id}	GET	type
/sessions	POST	
/merge/sessions	POST	
/sessions/{id} (<i>see limitation 5.3</i>)	PUT	

Parameter definitions:

- count:** parameter accepts either 'yes' or 'no' value.
 Default value is "no".
 When an invalid value is passed, the default value is kept.
- type:** valid entries are 'standard' and 'draftlist' (visual ID tag).
 If an invalid value is passed the reader will keep the default value which is standard.
- {id}:** the numeric identifier generated by the reader whenever it creates a session.

Example of Session related HTTP request:

GET /sessions HTTP/1.1

- returns all session in the reader with summary information only

GET /sessions /34 HTTP/1.1

- returns session information including animals information in the session with id 34

GET /sessions?type=draftlist HTTP/1.1
--

- returns all draftlist in the reader with summary information only

GET /sessions/23?type=draftlist HTTP/1.1

- returns draftlist information including animals information in the session with id 23

Note: The draft in a session must be a number from 0 – 9.

1 – 9 being the draft gates and 0 being others. If you want to send names for these gates, this can be done while transferring the draft list itself through the default trait fields (see 4.15 Changing draft gate names).

Note: The HR4/5 must always have a session on it.

If the last session is deleted, a new one will be created in the default name ([Date]A).

3.6. Response format

The response contains the HTTP protocol return code, a return message and an optional XML payload:

```
HTTP/1.1 {response code} {response message}
{headers}

{XML data payload}
```

For example, if a request was successfully received:

```
HTTP/1.1 200 OK
{headers}

{XML data payload}
```

When sending a response, the data payload must be formatted in XML format, using the Animal Data Interface schema (REF2) to validate against.

Below is an example response using the HTTP header *'Transfer-Encoding: chunked'* after a *'GET /treatments HTTP/1.1'* request:

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked

139
<?XML version="1.0" encoding="UTF-8"?>
<ads:body XMLns:xsi="http://www.w3.org/2001/XMLSchema-instance" XMLns:ads="
http://animaldatainterface.org/Schema/2014/ADI "><ads:header><ads:createdBy>GGL HR4/5
774456789000</ads:createdBy><ads:dateTimeCreated>2013-11-26T10:46:45.728+12:00</ads:dateTim
eCreated></ads:header>
5d
<ads:treatments><ads:treatment reference="1"><ads:type>5 in 1 Vacc</ads:type></a
ds:treatment>
48
<ads:treatment reference="2"><ads:type>Drench</ads:type></ads:treatment>
4a
<ads:treatment reference="3"><ads:type>Drenched</ads:type></ads:treatment>
48
<ads:treatment reference="4"><ads:type>Tagged</ads:type></ads:treatment>
48
<ads:treatment reference="5"><ads:type>Weaned</ads:type></ads:treatment>
1c
</ads:treatments></ads:body>
0
```

4. DEVICE CONSIDERATIONS

4.1. Device Serial Number

If an application sends its requests to multiple readers, it may wish to know the method of resource identification for each reader. It is recommended to use the serial number of the reader for this purpose as each reader has a unique serial number.

The XML below shows the serial number of device to be 1307235544.:

```
<?XML version="1.0" encoding="UTF-8"?>
<ads:body XMLns:ads=" http://animaldatainterface.org/Schema/2014/ADI "
XMLns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ads:header>
    <ads:createdBy>GGL HR4/5 1307235544</ads:createdBy>
    <ads:dateTimeCreated>2013-11-25T07:29:22.296+12:00</ads:dateTimeCreated>
  </ads:header>
</ads:body>
```

4.2. Device Configuration

To get the reader configuration, send the following request:

```
GET /config/info HTTP/1.1
```

The reader will return an XML that would be similar to the sample below. All values are strings.

```
HTTP/1.1 200 OK
{headers}

<config>
  <readerType>HR4</readerType>
  <softwareVersion>123</softwareVersion>
  <hardwareVersion>111</hardwareVersion>
  <serialNumber>1234567890</serialNumber>
</config>
```

4.3. Draftlist / Sortlist

A sort list contains a list of animal ID's, each with a sorting assignment. This assignment may be used to separate animals into predefined groups, or to flag specific animals for selection.

The 'Group Name' and 'Group Number' information maybe send in the 'defaultTraits' node of XML. If none are sent it will use the default values where:

Group Number	Group Name
0	Others
1	Group 1
2	Group 2
3	Group 3
4	Group 4
5	Group 5
6	Group 6
7	Group 7
8	Group 8

4.4. Partial Success

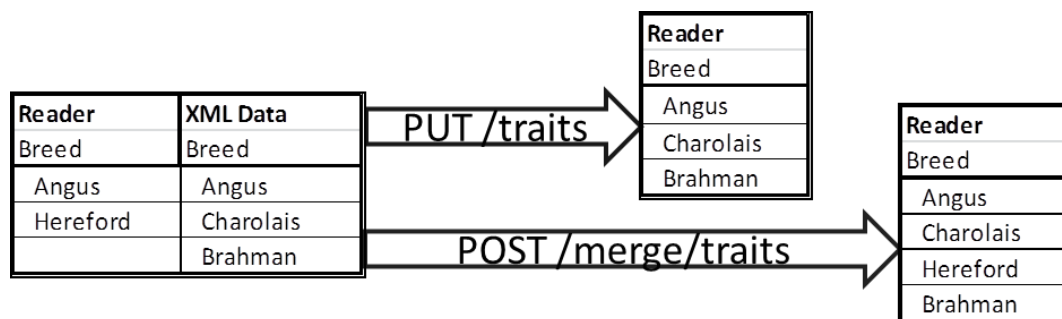
If a request fails half way through, the whole request will fail. For example, if a request is sent for 50 animals to be created and one entry does not have any identification fields, an error is returned and the whole 50 fail to create.

4.5. 'PUT' vs. 'POST/merge'

The 'PUT' command is used to replace the resource. The destination resource will be overwritten with the resource contained in the request.

The 'POST/merge' is used when the existing resource is to be updated.

The figure below illustrates the difference between these two commands using an example where the reader and XML have different values for the trait 'Breed':



4.6. Mothering Session

Mothering is a standard session that links the offspring and the Dam in a session. To pass a mothering session the 'session_kind' node of XML must contain the value 'Mothering'.

4.7. Device Time

The reader stores and sends UTC time in the XML response. It also expects UTC time when receiving data from client.

4.8. Treatment and Activity

Treatment and Activity are treated the same by the HR4/5 reader. Activities such as 'Drenched' and 'Weaned' are passed in the XML through the treatment node.

The reader does not support the following XML nodes:

- internalIdentifier
- product
- dosage
- units
- withholdingperiod
- exportslaughterinterval

An Activity is a list of value trait with 'yes' and 'no' items.

4.9. Deleting of an Animal

Deleting of an animal is carried out by deleting one animal at a time. Therefore the client has to send a *DELETE* request for each animal.

4.10. Supported Characters

The reader supports all uppercase and lowercase letters of English alphabet and the following 5 special characters:

- ‘.’ decimal point
- ‘-’ hyphen
- ‘/’ forward slash
- ‘:’ colon
- ‘ ’ space

These are the only valid characters for use in animal notes, session notes, session names, trait name, etc.

The *Visual ID (VID)* or *Tag* can only contain upper case letters and no space characters. The reader trims trailing and leading spaces.

4.11. Animal Notes created time

When entering multiple session animal notes, they must be separated by at least 1 second. If two different notes are created with the same recorded attributes, the reader will overwrite the first note.

If the resource identifier contains characters that are reserved, for example ‘*space*’ the hex equivalent is to be used, prefixed with ‘%’ (e.g. ‘%20’ = space).

Character	ASCII hex value
<i>space</i>	0x20
-	0x2d
.	0x2e
:	0x3a
/	0x2f

4.12. Reserved Names of Traits, Life Data and Activities

The reader has four types of traits:

- List of values
- Free text
- Numeric
- Date

Trait names are not case sensitive. Some names have been reserved as have special treatment in the reader database:

Trait type	Reserved names
List of value	sex, colour, color, breed, original pic, group, farm, property, area, condition, drench
Text	sire, dam, donor dam, status, original tag
Numeric	live weight, condition score
Date	dob

‘*Sort*’ and ‘*Draft*’ are two reserved words that must not be used as trait name.

For example, it is not possible to create a text trait with name 'breed'. This trait can only be created as list of values type.

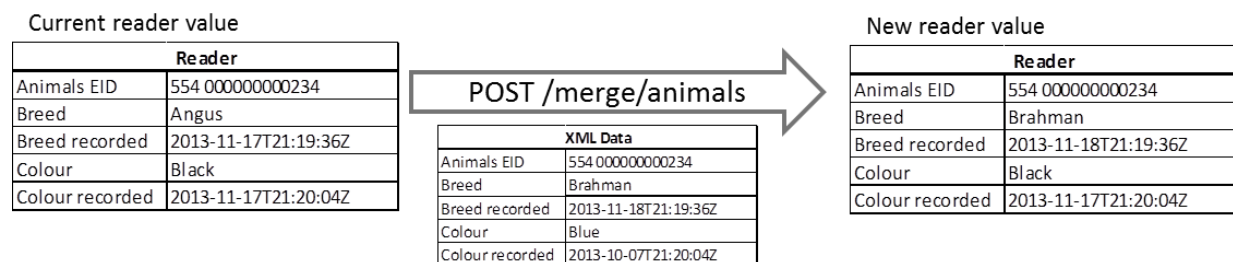
NOTE:

The HR4 reader can only support list of value type traits, and does not support text, numeric or date type traits.

4.13. Updating animals and its life data

When updating the animal life data using *POST /merge/animal* and the reader has the value with the latest time, it will keep that life data value and will not use the value that is passed within the XML.

In the below example, after the *POST /merge/animal* command, the reader's value for 'Breed' was changed to 'Brahman' because it has the most recent time recorded. For the same reason the colour will remain 'Black'.



4.14. Limits

Below are the known limits of the reader:

Parameter	Limit
Maximum session animals in a session	9999
Maximum total number of traits, life data and activities	250
Maximum items in a single List of Value trait	110
Maximum total number of List of Value entries over all traits	2500
Maximum traits per session in HR4	3
Maximum Traits per session in HR5 (not mothering)	9
Maximum Traits per session in HR5 (mothering - Dam is 9th)	8
Maximum session animal notes per session per animal	60
Maximum number of sessions	no limit
Maximum number of sort lists	no limit

4.15. Changing draft gate names

Draft gate names can be set via the defaultTrait field for the draft list. Here is an example...

```
<?xml version="1.0" encoding="utf-8"?>
<ads:body xmlns:ads="http://animaldatainterface.org/Schema/2014/ADI">
  <ads:header>
    <ads:version>1</ads:version>
    <ads:createdBy>Gallagher Group Ltd</ads:createdBy>
    <ads:dateTimeCreated>2015-07-22T16:46:14.6824745+12:00</ads:dateTimeCreated>
  </ads:header>
  <ads:sessions count="1">
    <ads:session sessionType="draftList">
      <ads:name>Example Draft List</ads:name>
      <ads:startDate>2015-07-22T16:43:11.4131494</ads:startDate>
      <ads:defaultTraits>
        <ads:defaultTrait>
          <ads:name>1</ads:name>
          <ads:defaultValue>Gate1</ads:defaultValue>
        </ads:defaultTrait>
        <ads:defaultTrait>
          <ads:name>2</ads:name>
          <ads:defaultValue>Gate2</ads:defaultValue>
        </ads:defaultTrait>
      </ads:defaultTraits>
      <ads:animals count="3">
        <ads:animal>
          <ads:animalId>
            <ads:eid>981 0000000000001</ads:eid>
          </ads:animalId>
          <ads:draft>1</ads:draft>
        </ads:animal>
        <ads:animal>
          <ads:animalId>
            <ads:eid>981 0000000000002</ads:eid>
          </ads:animalId>
          <ads:draft>1</ads:draft>
        </ads:animal>
        <ads:animal>
          <ads:animalId>
            <ads:eid>981 0000000000003</ads:eid>
          </ads:animalId>
          <ads:draft>2</ads:draft>
        </ads:animal>
      </ads:animals>
    </ads:session>
  </ads:sessions>
</ads:body>
```

4.16. Namespace changes

The namespace the HR5 uses for communication changed with version 1.08 of the HR5 code. This was to keep it the same as other products and it should not change again. Future changes (if any come up) will be tracked with the version number in the header of the XML.

4.17. Animal Alerts

The HR5 supports alerts, pop-ups that appear when an animal is scanned displaying a particular note. Any note can be set as an alert and there can be only one alert per animal.

The XML schema contains a field called <animalNote> for animal entity. The HR5 has no note that exists without a session, so the HR4/5 takes any existing note passed in this field as the alert for this animal. If this note doesn't match an existing note, then it will do nothing.

5. KNOWN ISSUES AND LIMITATIONS

5.1. Response on Invalid XML or Invalid Data

There are several cases when the reader will respond with the following:

Parsing Error. Payload is not valid XML
--

This does not accurately state if the error is due to invalid XML (failed the Animal Data Interface schema check) or if the data in the XML is invalid (e.g. invalid date format, invalid session name or invalid attributes).

When the chunked mode length is incorrect, the reader's HTTP server will timeout and thus exits the data transfer mode without any reply to the client.

5.2. Reader cannot recover after a read timeout.

In most cases the reader will restart its HTTP server and exit data transfer mode, returning to the Main Menu screen.

On rare occasions when the HTTP server times out waiting for the end of data, or waiting to parse the correct request, it may fail to recover. The only way to correct this issue is to power cycle the reader.

5.3. PUT sessions currently does not work.

Use POST /sessions or POST /merge/sessions until it is fixed.