



eyepin Interface API 2.0.4

09.03.2010

Table of contents:

1. Introduction	3
2. Response XML.....	4
3. Request Commands	5
3.1. Address-Commands.....	5
3.1.1. AddressInsert	5
3.1.2. AddressUpdate	8
3.1.3. AddressGet.....	8
3.1.4. AddressDelete.....	9
3.1.5. AddressSignIn.....	9
3.1.6. AddressSignOut	9
3.2. Content Commands.....	9
3.2.1. ContentInsert.....	9
3.2.2. ContentDelete	11
3.3. Newsletter Commands.....	12
3.3.1. NewsletterCreate	12
3.4. Sweepstake Commands	16
3.4.1. SweepstakeParticipate	16
3.5. Misc Commands.....	17
3.5.1. GetURL	17

1. Introduction

The following is a guideline to the standard eyepin interface. This interface gives you the opportunity to handle data in both directions - import and export data records from your CMS/CRM to eyepin and vice versa.

It's reasonable that most customers prefer to use their own CMS/CRM to track their company related content (e.g. addresses, articles).

The data transfer takes place via XML-Records (Extended Markup Language). Depending on what you intend to do, there are special commands for altering (edit, remove, add) data records, which will be explained in this document later.

The idea is very simple, a URL is called which pass the data/request to eyepin and you get a response back (2. Response XML).

Further more there are so called event-handlers, which can perform actions (e.g. calling a url with passing data) when something happens to the eyepin user database (e.g. someone changing his personal information).

Transport:

- URL request via HTTP
- Send command and data via POST-variables

Security:

Login via basic-authentication

It's required to send the interface username & password from your eyepin account within the header

eyepinURL:

<http://letter.eyepin.com/interface2.php>

POST-Variables:

cmd.....Command – see below

data.....data to be submitted – the actual data, either a value or a XML structure
(action dependent)

2. Response XML

Each request returns a response-XML in the following form:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<response>
  <code></code>
  <description></description>
  <data></data>
</response>
```

Description of XML -Tags:

code

- 2000 .. OK
- 3001 .. Database error
- 3002 .. Error: Access from this IP denied
- 3003 .. Error: Command missing
- 3004 .. Error: Data missing
- 3005 .. Error: Username and/or password wrong
- 3006 .. Error: Command needs authentication
- 3007 .. Error: Cannot parse XML
- 3008 .. Error: Unknown command
- 3009 .. Error: Command is not available any more

Non listed codes are explained in detail through the description which can be found in the xml response.

description

Description of the response code

data

In case your request returns data, it can be found here. The data can be either an ID or a whole xml-structure (request dependent).

3.Request Commands

3.1.Address-Commands

3.1.1.AddressInsert

Data:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<address key="attribute1">
  <email></email>
  <salutation></salutation>
  <title></title>
  <firstname></firstname>
  <lastname></lastname>
  <street></street>
  <zip></zip>
  <city></city>
  <country></country>
  <phone></phone>
  <fax></fax>
  <mobile></mobile>
  <internet></internet>
  <company></company>
  <function></function>
  <attribute1></attribute1>
  ...
  <attribute20></attribute20>
  <status></status>
  <newsletterthemes></newsletterthemes>
  <newslettertopics></newslettertopics>
  <language></language>
  <newsletterformat></newsletterformat>
  <handleoptin></handleoptin>
  <handleoptout></handleoptout>
</address>
```

Use <![CDATA[`text`]]> to send html-code or special characters in a field

Description of XML-Tags:

attribute "key"

Determines by which address-field the data-record will be synchronized. if "key" is not specified, default-key is email. This can be used for example if you have stored your own unique UserID in an attribute-field.

attribute "save-empty-fields"

if set to "1" empty values will be written to database. default empty values are ignored.

email, title, firstname, lastname, street, zip, city, country, phone, fax, mobile, internet, company, function

Type: all VARCHAR(255), except title VARCHAR(30), zip VARCHAR(10)

Required: the field which is set by attribute "key", otherwise "Email"

Standard address fields

salutation

Type: VARCHAR(10)

Value "Herr" for male and value "Frau" for female

attribute1 - attribute20

Type: VARCHAR(255)

There are 20 custom-fields in the address-database, you can rename and set the type of the attributes through eyepin, still you have to use attribute1-attribute20 when using them (doesn't matter if it's an interface or a signup form issue).

status

Type: CHAR(1), Required: YES

Subscribe-status of address. "1" means subscribed and "0" unsubscribed.

language

Type: CHAR(2)

2-letter-code (e.g. "en" or "de")

newsletterthemes

Type: VARCHAR(255)

Comma-separated IDs of topics available in your eyepin-account. given IDs will be appended to existing IDs. If you want to replace IDs, please use tag "newslettertopics" instead.

newslettertopics

Type: VARCHAR(255)

Comma-separated IDs of topics available in your eyepin-account. given IDs will replace existing IDs. This tag overrules tag "newsletterthemes".

newsletterformat

Type: CHAR(4)

Determines in which format the user receives his newsletters. "both" for multipart, and "text" for text-only-version.

handleoptin

Type: CHAR(1)

if set to "1" the confirmation-mail for subscription will be sent. This sets "status" automatically to "1" regardless of what is given. This feature takes single- or double-optin-setting from eyepin preferences.

handleoptout

Type: CHAR(1)

if set to "1" the confirmation-mail for unsubscription will be sent. This sets "status" automatically to "0" regardless of what is given. This feature takes single- or double-optout-setting from eyepin preferences.

Response codes:

4101 .. Error: Specified syncfield not given in sent data

4102 .. Error: Email is not valid
4103 .. Error: Address is blacklisted

There is an alternative variant of sending data for eCRM-Accounts, where subscribe-status and topics for each customer can be specified:

Data:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<address key="attribute1">
  <email></email>
  <salutation></salutation>
  <title></title>
  <firstname></firstname>
  <lastname></lastname>
  <street></street>
  <zip></zip>
  <city></city>
  <country></country>
  <phone></phone>
  <fax></fax>
  <mobile></mobile>
  <internet></internet>
  <company></company>
  <function></function>
  <attribute1></attribute1>
  ...
  <attribute20></attribute20>
  <language></language>
  <newsletterformat></newsletterformat>
  <medium>
    <customerid></customerid>
    <status></status>
    <topicsids></topicsids>
  </medium>
  ...
  <medium>
    <customerid></customerid>
    <status></status>
    <topicsids></topicsids>
  </medium>
</address>
```

Description of XML-Tags:

medium->customerid

Type: INT(10) , Required: YES
ID of customer

medium->status

Type: CHAR(1) , Required: YES
Subscribe-status of address. "1" means subscribed and "0" unsubscribed.

medium->topicsids

Type: VARCHAR(255)
Comma-separated IDs of topics available in your eyepin-account. given IDs will replace existing IDs.

3.1.2.AddressUpdate

This command does not exist. Use AddressInsert instead, which automatically determines if a record has to be inserted or updated.

3.1.3.AddressGet

Data:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<addressget key="email">
  <value></value>
</addressget>
```

Description of XML-Tags:

attribute "key"

If not given, default-key is email. Please note that if you do not specify an field with unique values, you will get only the first found record.

data in Response XML:

```
<data>
  <email></email>
  <salutation></salutation>
  <title></title>
  <firstname></firstname>
  <lastname></lastname>
  <street></street>
  <zip></zip>
  <city></city>
  <country></country>
  <phone></phone>
  <fax></fax>
  <mobile></mobile>
  <internet></internet>
  <company></company>
  <function></function>
  <attribute1></attribute1>
  ...
  <attribute20></attribute20>
  <language></language>
  <newsletterformat></newsletterformat>
  <medium>
```

```
<customerid></customerid>
<status></status>
<topicids></topicids>
</medium>
...
<medium>
  <customerid></customerid>
  <status></status>
  <topicids></topicids>
</medium>
</data>
```

Response codes:

4201 .. Error: Record not found
4202 .. Error: Key not given

3.1.4.AddressDelete

Data: Email or AddressID (will be recognized automatically by eyepin)

3.1.5.AddressSignIn

Data: Email or AddressID (will be recognized automatically by eyepin)

3.1.6.AddressSignOut

Data: Email or AddressID (will be recognized automatically by eyepin)

3.2.Content Commands

3.2.1.ContentInsert

Data:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<content>
  <catid></catid>
  <title></title>
  <opener></opener>
  <detail></detail>
  <pictureurl></pictureurl>
  <detailurl></detailurl>
  <detailname></detailname>
  <mediaitemid></mediaitemid>
```

```
<topics></topics>
</content>
```

Use <![CDATA[`text`]]> to send html-code or special characters in a field

Description of XML-Tags:

catid

Type: INT(10)
CategoryID of folder in eyepin-system. Leave empty or "0" to put article in root-folder.

title

Type: VARCHAR(255)
Title of article

opener

Type: TEXT
Teaser-text

detail

Type: LONGTEXT
Text of landing-page, when user clicks "more"

pictureurl

Type: VARCHAR(255)
URL to external picture. Overrides parameter "mediaitemid" (see below for further information)

detailurl

Type: VARCHAR(255)
URL to external landing-page (e.g. a page on your website). If given the parameter "detail" is unnecessary.

detailname

Type: VARCHAR()
Name of click-through-link, which goes to landing-page (or detailurl)

mediaitemid

Type: INT(10)
ID of a picture in the eyepin picture-database.

topicids

Type: VARCHAR(255)
IDs of eyepin topics associated with this article.

Response data:

On succes, ID of content

Response codes:

5101 .. Error: Access violation, content does not belong to this customer
5102 .. Error: Content not found

3.2.2.ContentDelete

Data: ContentID

Response codes:

2000 .. Ok, content deleted
5101 .. Error: Access violation, content does not belong to this account
5102 .. Error: Content not found

3.3. Newsletter Commands**3.3.1. NewsletterCreate**

```
<?xml version="1.0" encoding="iso-8859-1"?>
<campaignitem>
  <templateid></templateid>
  <addressfilterid></addressfilterid>
  <fromname></fromname>
  <senddatetime></senddatetime>
  <subject></subject>
  <salutationid></salutationid>
  <title></title>
  <header></header>
  <headermediaitemid></headermediaitemid>
  <footer></footer>
  <footermediaitemid></footermediaitemid>
  <language></language>
  <donewslettertopics></donewslettertopics>
  <sendifnocontent></sendifnocontent>
  <embedimages></embedimages>
  <newsletterformat></newsletterformat>
  <documentids></documentids>
  <snippets></snippets>

  <custom1></custom2>
  ...
  <custom10></custom10>

  // articles for one area
  <area num="1">
    <content id="101" />
    <content id="102" />
    <content id="103" />
  </area>

  // articles for one area with content-filter
  <area num="2">
    <content filterid="4" id="201" source="Content"
      id_alt="211" source_alt="Content" />
    <content filterid="5" id="202" source="Content"
      id_alt="212" source_alt="Content" />
  </area>

  // articles for one area with groups
  <area num="3">
    <group title="group 1">
      <content id="201" />
      <content id="202" />
    </group>
    <group title="group 2">
      <content id="203" />
      <content id="204" />
    </group>
  </area>
</campaignitem>
```

```
</campaignitem>
```

Use <![CDATA[`text`]]> to send html-code or special characters in a field

Description of XML-Tags:

templateid

Type: INT(10) , Required: YES
ID of template

addressfilterid

Type: INT(10) , Required: YES
ID of address-filter, set to "0" if you want to send to all addresses

fromname

Type: VARCHAR(50)
displayed sender-name (sender-email is fix defined in eyepin)
leave empty to use default-fromname defined in your eyepin-app

senddatetime

Type: DATETIME (YYYY-mm-dd HH:ii:ss)
If set, the newsletter is scheduled for sending, otherwise the newsletter is in „Drafts“

subject

Type: VARCHAR(255) , Required: YES
Subject of newsletter

salutationid

Type: INT(10)
ID of salutation

title

Type: VARCHAR(255)
Title of newsletter

header

Type: TEXT
Editorial-text of newsletter

headermediaitemid

Type: INT(10)
ID of an picture in the eyepin picture-database.

footer

Type: TEXT
Footer-text of newsletter

footermediaitemid

Type: INT(10)
ID of an picture in the eyepin picture-database.

language

Type: CHAR(2) , Required: YES
two-letter of used language (e.g. "de" or "en")

donewslettertopics

Type: CHAR(1)
Set to „1“ so enable that articles with assigned topics will only be included if an user has subscribed to the according topic

sendifnocontent

Type: CHAR(1)
If set to „1“ the newsletter will not be sent to a user if there are no articles available he has subscribed to (topics)

embedimages

Type: CHAR(1)
If set to „1“ all pictures of the newsletter will be embedded into the newsletter.
This is not a standard-feature, and has to be enabled by request.

newsletterformat

Type: CHAR(1), Default: both
Set to „both“ for multipart-newsletter or „text“ for text-only-newsletter

documentids

Type: VARCHAR(255)
Comma-separated IDs of documents to attach

snippets

Type: VARCHAR(255)
Comma-separated IDs of enabled template-snippets, the selected template provides

custom1 - custom10

Type: VARCHAR(255)
Custom-fields (are defined in template)

area attribute "num"

Type: SMALLINT(6)
Contains articles for the selected area (template has to provide this area)

content attribute "id"

Type: INT(10)
ID of article

content attribute "source"

Type: VARCHAR(16), Default: "Content"
Source of article
Possible values: "Sweepstake", "Event", "Survey", "Form"

content attribute "filterid"

TYPE: INT(10)

ID of address-filter for use as dynamic content-filter. if an user matches the filter he will get the standard-article (content->id) otherwise, if set, the alternative-article (content->id_alt)

content attribute "id_alt"

Type: INT(10)

ID of alternative-article (for use with content->filterid)

content attribute "source_alt"

Type: VARCHAR(16), Default: "Content"

Source of alternative-article (for use with content->filterid)

Possible values: "Sweepstake", "Event", "Survey", "Form"

group attribute "title"

Type: VARCHAR(255)

Title of an article-group (template has to provide this feature)

Response data:

On success, ID of newsletter

3.4. Sweepstake Commands

3.4.1. SweepstakeParticipate

```
<?xml version="1.0" encoding="iso-8859-1"?>
<sweepstakeparticipate>
  <id></id>
  <address key="Email"></address>
  <answer></answer>
  <code></code>
  <custom1></custom1>
</sweepstakeparticipate>
```

Use <![CDATA[`text`]]> to send html-code or special characters in a field

id

Type: INT(10), Required: YES

ID of Sweepstake in eyepin

address

Type: VARCHAR(255) , Required: YES

Identifier of the user, default-key is the field "Email"

answer

Type: TEXT

optionally the answer if there's a question defined for the sweepstake

code

Type: VARCHAR(255)

optionally a win-code which is entered by the user

custom1

Type: VARCHAR(255)

optionally extra custom-information which is stored with the participation (i.e. code-origin)

Response codes:

2000 .. Ok, successfully participated

4201 .. Error, Address not found

6001 .. Error, Sweepstake not found

6002 .. Error, Sweepstake not active

6003 .. Error, Address has already participated (only if multiple participation disabled)

3.5.Misc Commands

3.5.1.GetURL

```
<?xml version="1.0" encoding="iso-8859-1"?>
<geturl>
  <type></type>
  <id></id>
  <address key="Email"></address>
</geturl>
```

Generates the link to a newsletter, sweepstake, event or survey with encrypted parameters (optionally personalized).

type

Type: VARCHAR(255), Required: YES

Possible values are: newsletter, sweepstake, event, survey

id

Type: INT(10), Required: YES

ID of item

address

Type: VARCHAR(255), Required: NO

Identifier of the user, default-key is the field "Email"

Response codes:

2000 .. Ok, successfully participated

4201 .. Error, Address not found

7001.. Error, Unknown type

Response data:

On success an URL