Automated Electronic Positive Response via REST API

Automated Electronic Positive Response (EPR) via REST API

This allows the member to submit EPR by POSTING JSON object to a REST URL.

Requirements

- Must be a MEMBER of DigAlert
- Must have a valid Token. Tokens are available by contacting the center PRIOR to any posting (live or test).
- Use the POST method with a content type of application/json and a data payload of a JSON object
- JSON objects MUST contain a 32 character authorization token and a responses array of up to 100 positive response objects.

TESTING Automated EPR

For members needing to test Automated EPR via REST API use the following URL

- Connect to https://testresp.digalert.org/positive_response (this URL is for TESTING purposes only. All responses submitted to this URL will NOT show on tickets or deliver to excavators)
- To verify the response was successful, first log into testresp.digalert.org via a web browser. Click https://newtint.digalert.org/newtinweb/gen_resp_inquiry_usas.nas, enter the ticket number and click Submit.
Components

Components of the JSON Object

- The parameter names MUST all be in lower case and be enclosed in quotes
- All parameters must be string values except for id which can be numeric, in which case the enclosing quotes are not needed
- The comments and URL fields have maximum length of 255 characters. If you want any line breaks in the comments, they should be represented by the 4 character string \r\n
- The order of the items do not matter
- The batch input consists of the authorization token and a responses array. The line breaks and indentation are shown below for clarity but are not required.

A positive response object consists of (in any order)

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>This is an optional argument used for reference only and not used in the single response format.</td>
</tr>
<tr>
<td>Ticket</td>
<td>The complete ticket number EXCLUDING the revision (A121231234)</td>
</tr>
<tr>
<td>Member</td>
<td>The member code posting the response</td>
</tr>
<tr>
<td>Response</td>
<td>The three digit code indicating the response of the locate request. These codes can be found Electronic Positive Response Codes</td>
</tr>
<tr>
<td>Respondent</td>
<td>The person's name or initials of who is entering the response. A minimum of three (3) characters are required. If the person only has two initials, send a space between them to fulfill the 3 character requirement.</td>
</tr>
<tr>
<td>URL</td>
<td>This is an optional argument used to attached a link to the member's web site with additional information for this ticket. It has a maximum length of 255 characters and must be properly encoded according to RFC 3986</td>
</tr>
<tr>
<td>Comments</td>
<td>This is an optional argument used to add additional comments to store with the response. This has a maximum length of 255 characters</td>
</tr>
</tbody>
</table>

Examples of Transmission and Return Values

When posting a large number of responses, allow sufficient time for the batch to complete.

Posting Response(s) Example
Data Sent

```json
{
    "token": "AaBbCcDdEdFgHhSsTtUuVvWwXxYyZz",
    "responses": [
        {
            "id": 1,
            "ticket": "A123456789",
            "member": "MYUTIL",
            "response": "123",
            "respondent": "John Doe"
        },
        {
            "response": "60",
            "member": "MYUTIL",
            "respondent": "John Doe",
            "ticket": "B123456789"
        },
        {
            "id": "{6F9619FF-8B86-D011-B42D-00C04FC964FF}",
            "ticket": "A987654321",
            "member": "MYUTIL",
            "response": "234",
            "respondent": "Larry Locator",
            "url": "http://somedomain.com/resource"
        },
        {
            "ticket": "B987654321",
            "member": "MYUTIL",
            "comments": "The entry gate was locked",
            "response": "345",
            "respondent": "Mark Lineman",
            "id": 4
        }
    ]
}
```

The returned data after the headers will have a results array and will resemble

```json
{
    "results": [
        {
            "id": 1,
            "ticket": "A123456789",
            "member": "MYUTIL",
            "response": "123",
            "status": "NNN Description"
        },
        {
            "ticket": "B123456789",
            "member": "MYUTIL",
            "response": "60",
            "status": "NNN Description"
        },
        {
            "id": "{6F9619FF-8B86-D011-B42D-00C04FC964FF}",
            "ticket": "A987654321",
            "member": "MYUTIL",
            "response": "234",
            "status": "NNN Description"
        },
        {
            "id": 4,
            "ticket": "B987654321",
            "member": "MYUTIL",
            "response": "345",
            "status": "NNN Description"
        }
    ]
}
```
Possible results for NNN shown above in 'Status':

250 OK
251 Duplicate response (same response code that was already sent)
252 Ticket has been cancelled
450 Insufficient information
451 Invalid ticket
452 Ticket has expired
453 Invalid member code
453 Cannot respond for a contract locator
453 Cannot respond for an excluded member
454 Member not on ticket
454 Member not allowed for current login
455 Invalid response code
455 Invalid response code for ticket category
456 Invalid explanation code
457 Respondent name under 3 bytes (use "F L" if no middle initial)
457 Malformed URL
457 URL is too long
457 Maximum comment length exceeded

If an ID value was given with a response, it will be returned.
Replies beginning with 2xx indicate the response was received and successful.
Replies beginning with 45x indicates invalid data was sent.
Replies beginning with 5xx indicate an error and should try again later.

A reply code in the 450 to 459 range usually requires corrective action before attempting to resend the information. The single exception is the 451 response for an invalid ticket. Because the tickets are taken on more than one server, there may be latency or a temporary internet connectivity error in transferring the data between the server where the ticket was created and the server where the response is being posted. These responses should be resubmitted after a delay of a few minutes.

Important!
The application should not blindly resend responses that fail to be accepted. Any response that has been repeatedly rejected should be manually checked. Any response that has not been accepted for more than 7 days since it was entered should have its status cleared so that the application no longer tries to submit it.