Automated Checking Electronic Positive Response via REST API

Automated Checking Electronic Positive Responses (EPR) via REST API

This allows the member to audit EPRs by requesting all responses submitted via 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).

• Connect to https://resapi.digalert.org/positive_response (This URL is for PRODUCTION use only.

TESTING Automated EPR Checking

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

• Connect to https://testresp.digalert.org/positive_response (this URL is for TESTING purposes only.)
Components

Checking responses via REST API can use either the POST or GET methods. Both are explained below.

The examples show 3 tickets, this can be one ticket or as many up to 100.

Using the GET method

GET /positive_response?token=TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT&tickets=T1+T2+T3

*Replace T1, T2, T3, etc with complete ticket numbers (A121231234, A121231234, etc.)*

Using the POST method

- 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

POST /positive_response

```
{
    "token": "TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT",
    "tickets": ["T1", "T2", "T3"]
}
```

If the JSON object is improperly formatted, the reply will be **HTTP/1.1 400 Bad Request**

If the authorization token is missing or incorrect, the reply will be **HTTP/1.1 403 Forbidden**

If there are more than 100 responses in the array, the reply will be **HTTP/1.1 413 Request Entity Too Large**

Otherwise, the reply will be **HTTP/1.1 201 Created** and the data after the headers will contain results.
Examples of Return Values

When posting a large number of responses, allow sufficient time for the batch to complete.
All items (fields) that are empty WILL NOT be present.

Results for a Single Ticket Will Resemble

```json
{
    "member": "MEMBER",
    "response": "NNN",
    "description": "DDDDDDDDDDDDDDDDDDDDDDDDDDDDDD",
    "responded": "YYYY-MM-DD HH:MM:SS",
    "respondent": "RRRRRRRRRRRRRRRRRRRR",
    "url": "UUUUUUUUUUUUUUUUUUUUUUUUUUUU",
    "comments": "CCCCCCCCCCCCCCCCCCCCCCCCCCCCC"
}
```

Results for Multiple Tickets Will Resemble

```json
{
    "status": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "results": [{
        "ticket": "TTTTTTTTTT",
        "completed": "YYYY-MM-DD HH:MM:SS",
        "work_date": "YYYY-MM-DD HH:MM:SS",
        "expires": "YYYY-MM-DD HH:MM:SS",
        "count": N,
        "responses": [{
            "member": "MEMBER",
            "response": "NNN",
            "description": "DDDDDDDDDDDDDDDDDDDDDDDDDDDDDD",
            "responded": "YYYY-MM-DD HH:MM:SS",
            "respondent": "RRRRRRRRRRRRRRRRRRRR",
            "excavator_rep": "EEEEEEEEEEEEEEEEEEEEEE",
            "url": "UUUUUUUUUUUUUUUUUUUUUUUUUUUU",
            "comments": "CCCCCCCCCCCCCCCCCCCCCCCCCCCCC"
        },
        { "member": "MEMBER",
          "response": "NNN",
          "description": "DDDDDDDDDDDDDDDDDDDDDDDDDDDDDD",
          "responded": "YYYY-MM-DD HH:MM:SS",
          "respondent": "RRRRRRRRRRRRRRRRRRRR",
          "excavator_rep": "EEEEEEEEEEEEEEEEEEEEEE",
          "url": "UUUUUUUUUUUUUUUUUUUUUUUUUUUU",
          "comments": "CCCCCCCCCCCCCCCCCCCCCCCCCCCCC"
        }], ...
      }, ...
    }
}
```

If an invalid ticket number is given or the ticket is no longer on the system, the ticket completed (submitted), work, and expires dates will show null.