Overview of EDGAR Application Programming Interfaces (“APIs”)

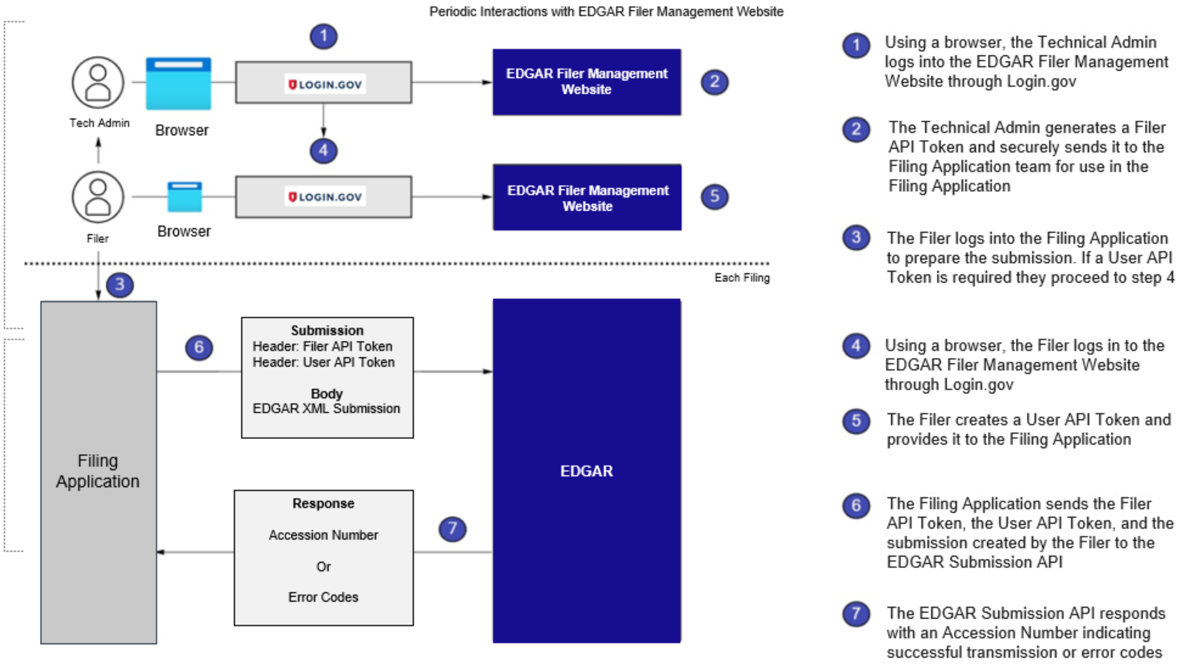
Section 1. Introduction to APIs

The proposed EDGAR Next API design includes three REST APIs:

1. A Submission API that would include transmit TEST and transmit LIVE endpoints (Transmit TEST and Transmit LIVE)
2. A Submission Status API
3. An EDGAR Operational Status API

The User API Token would only be required for the EDGAR Submission API. There would be different inputs and outputs for each of the APIs, as discussed below. Each of the three APIs would follow the same general workflow process. Figure A provides an example of the workflow process for the three APIs.

Figure A: EDGAR Submission API Workflow



1 Using a browser, the Technical Admin logs into the EDGAR Filer Management Website through Login.gov

2 The Technical Admin generates a Filer API Token and securely sends it to the Filing Application team for use in the Filing Application

3 The Filer logs into the Filing Application to prepare the submission. If a User API Token is required they proceed to step 4

4 Using a browser, the Filer logs in to the EDGAR Filer Management Website through Login.gov

5 The Filer creates a User API Token and provides it to the Filing Application

6 The Filing Application sends the Filer API Token, the User API Token, and the submission created by the Filer to the EDGAR Submission API

7 The EDGAR Submission API responds with an Accession Number indicating successful transmission or error codes

Section 2. EDGAR Submission API—Introduction

Filers and filing agents would make submissions using the EDGAR submission API in the following manner:

1. Using a browser, a technical administrator would log in to EDGAR by entering their individual account credentials and completing multi-factor authentication.
    - If a valid filer API token had already been generated, the technical administrator would then skip to step 6 below.
2. The technical administrator would generate a filer API token from the dashboard and securely provide the filer API token to the filing application.
3. The user would log into the filing application to prepare the submission.
4. Using a browser, the user would log in to EDGAR by entering their individual account credentials and completing multi-factor authentication.
5. The user would generate a user API token from the dashboard and securely provide the user API token to the filing application.
6. The filing application would send the filer API token, the user API token, and the submission created by the filer to the EDGAR submission API.
7. The EDGAR submission API would indicate to the filing application whether the submission was successfully transmitted to EDGAR, or, if not, what error may have occurred.
8. The filing application would inform the filer or filing agent whether the submission was successfully transmitted, or, if not, whether an error occurred.

In the majority of cases, where valid API tokens for the filer and the user were already generated and active, we anticipate that the EDGAR submission API would only require steps 3, 6, and 7 above. Step 3 would be the only manual step. Steps 6 and 7 would be performed via machine-to-machine communication.

We plan that, among other things, the filer API token, user API token, and filing submission would be required inputs via the POST method[1] to the EDGAR submission API. The EDGAR submission API would process, validate, and issue an accession number for each request. The filer API token and user API token would use JSON web token (“JWT”).[2] The filing application would provide the header data, filer API token, user API token, and the EDGAR submission data as a required input to the EDGAR submission API. The EDGAR submission API would provide a success or failure message in the JSON format.

Section 3. EDGAR Submission API—Inputs and Outputs

The transmit filings endpoints for TEST and LIVE would allow a filing submitted through the API to be submitted to EDGAR.

The Filer API Token, User API Token, and filing submission would be required inputs via the POST method to the EDGAR Submission API.

The EDGAR Submission API would process, validate, and issue an accession number for each request.

The Filer API Token and User AAPI Token would use JSON web encryption (JWE).

The filing application would provide the header data, Filer API Token, User API Token, and the EDGAR submission data as a required input to the EDGAR Submission API endpoint.

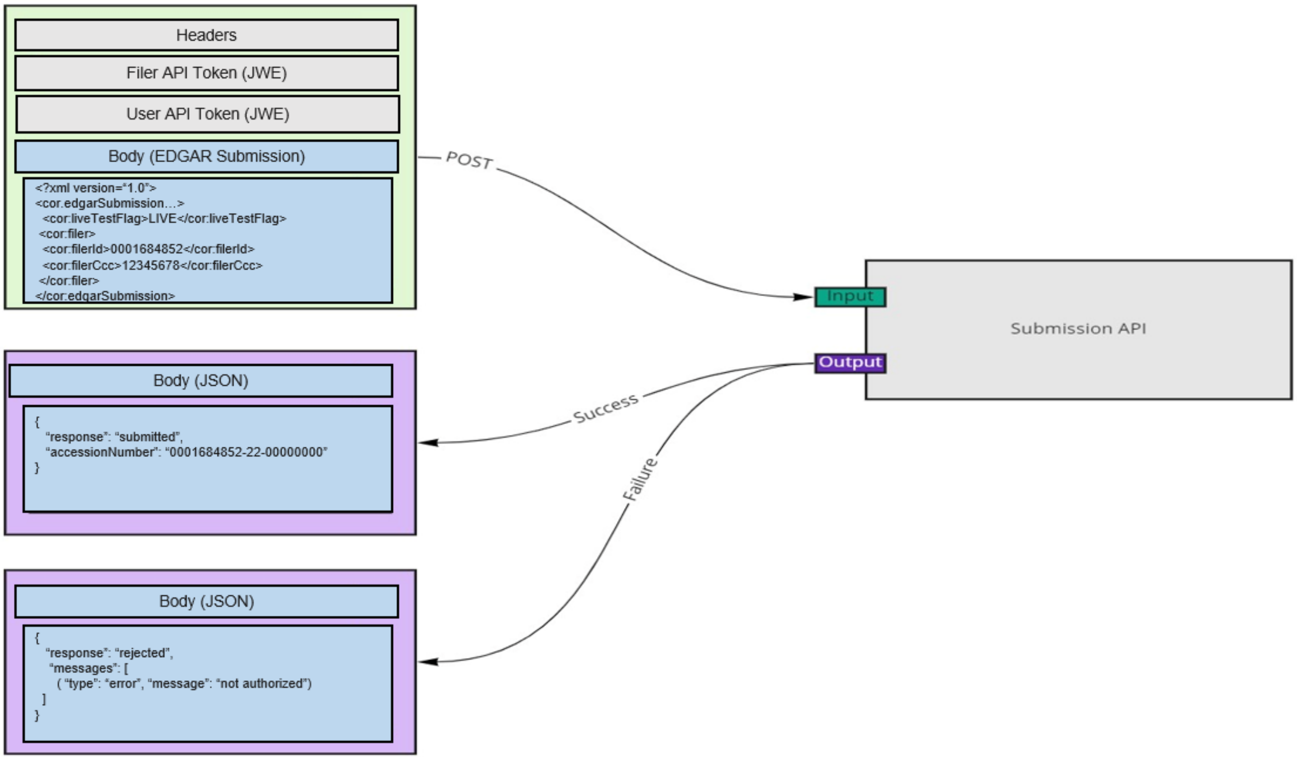
[1] POST is a common HTTP method designed to send data to a server.

[2] JWT is an open, industry standard RFC 7519 method for representing claims securely between two parties. *See* RFC 7519, available at https://www.rfc-editor.org/info/rfc7519. The RFC Series contain technical and organizational documents about the internet. *See generally* The RFC Series, available at https://www.rfc-editor.org/.

The output would provide a success or failure message in the JSON format.

Additional information is provided in Figure B.

FIGURE B: EDGAR SUBMISSION API INPUTS AND OUTPUTS



```
<?xml version="1.0">
<cor.edgarSubmission…>
  <cor:liveTestFlag>LIVE</cor:liveTestFlag>
 <cor:filer>
  <cor:filerId>0001684852</cor:filerId>
  <cor:filerCcc>12345678</cor:filerCcc>
 </cor:filer>
</cor:edgarSubmission>
```

```
{
   "response": "submitted",
   "accessionNumber": "0001684852-22-00000000"
}
```

```
{
   "response": "rejected",
    "messages": [
      ( "type": "error", "message": "not authorized")
  ]
}
```

Section 4. EDGAR Submission Status API—Inputs and Outputs

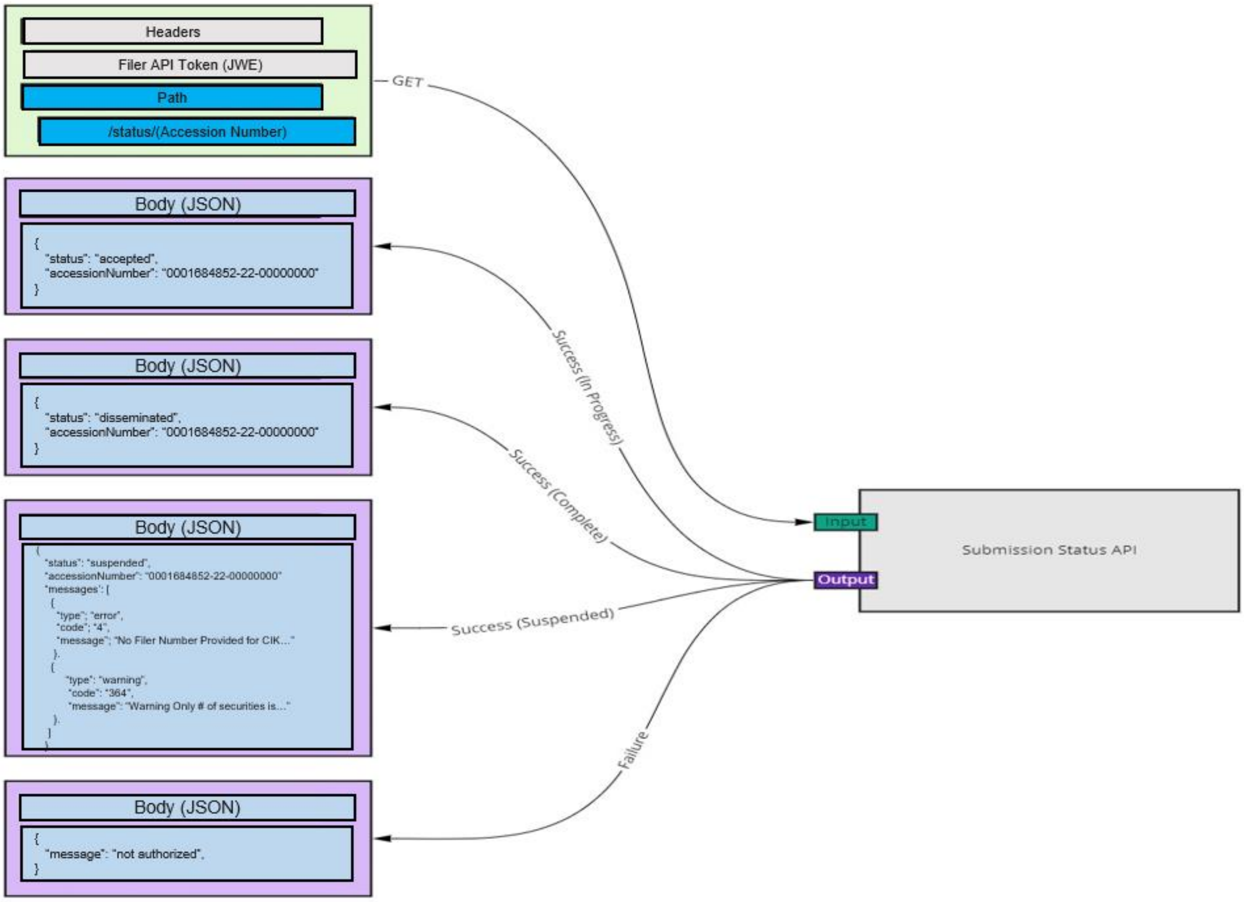
The EDGAR Submission Status API would require the Filer API Token and accession number(s) in order to return submission status by accession number if valid (meaning the accession number exists and the Filer API Token is from the submitting filer).

The data required by the EDGAR Submission Status API to return status would include header data, the Filer API Token, and accession number as a path variable, via the GET method.

The EDGAR Submission Status API would return the status of the filing in the JSON format.

Additional information is provided in Figure C.

FIGURE C: EDGAR SUBMISSION STATUS API INPUTS AND OUTPUTS FOR A SINGLE SUBMISSION



```
{
  "status": "accepted",
  "accessionNumber": "0001684852-22-00000000"
}
```

```
{
  "status": "disseminated",
  "accessionNumber": "0001684852-22-00000000"
}
```

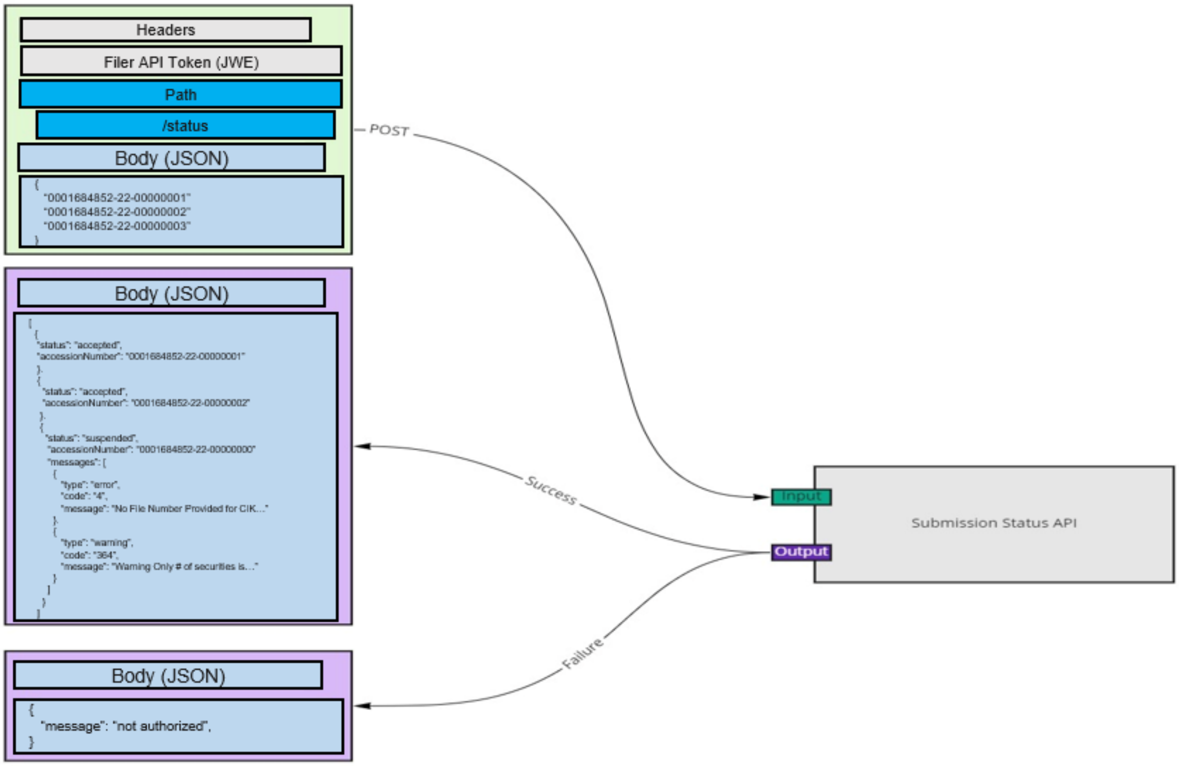
```
{
  "status": "suspended",
  "accessionNumber": "0001684852-22-00000000"
  "messages': [
   {
     "type"; "error",
     "code"; "4",
     "message"; "No Filer Number Provided for CIK..."
    },
   {
       "type": "warning",
        "code": "364",
        "message": "Warning Only # of securities is..."
    },
  ]
}
```

```
{
  "message": "not authorized",
}
```

To request multiple accession number statuses, each accession number would need to be provided in a JSON format list and sent via an HTTP POST in addition to the headers and Filer API Token.

Additional information is provided in Figure D.

FIGURE D: EDGAR SUBMISSION STATUS API INPUTS AND OUTPUTS FOR MULTIPLE SUBMISSIONS



```
{
  "0001684852-22-00000001"
  "0001684852-22-00000002"
  "0001684852-22-00000003"
}
```

```
[
 {
  "status": "accepted",
  "accessionNumber": "0001684852-22-00000001"
 },
 {
  "status": "accepted",
  "accessionNumber": "0001684852-22-00000002"
 },
 {
  "status": "suspended",
  "accessionNumber": "0001684852-22-00000000"
  "messages": [
   {
     "type": "error",
     "code": "4",
     "message": "No File Number Provided for CIK..."
   },
   {
     "type": "warning",
     "code": "364",
     "message": "Warning Only # of securities is..."
   }
  ]
 }
]
```

```
{
  "message": "not authorized",
}
```

Section 5. EDGAR Operational Status API—Inputs and Outputs

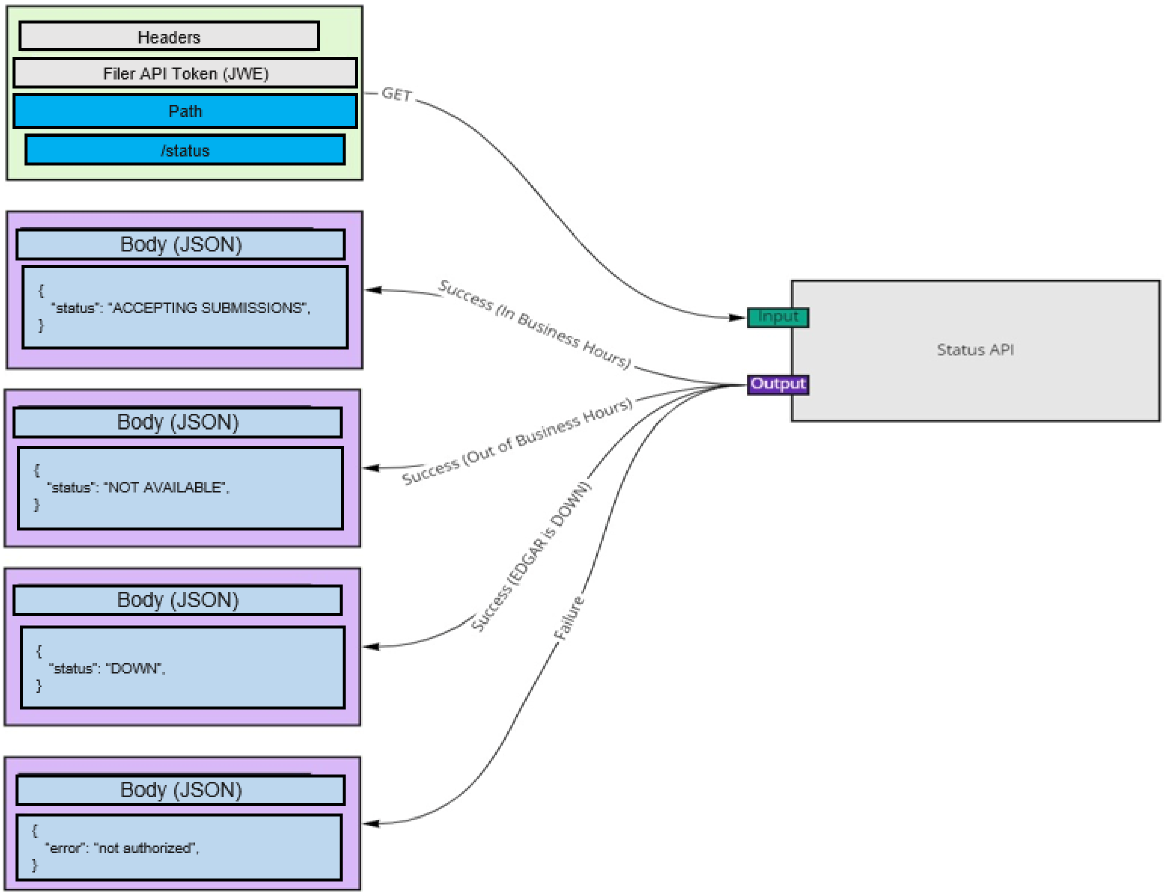
The EDGAR Operational Status API would provide the EDGAR operational status to filing applications with an active Filer API Token. The filing application software would call the EDGAR Operational Status API which would return one of the following statuses:

- "Accepting submissions"
  - EDGAR is available because EDGAR is reporting that it is within operating hours.
- "Not available"
  - EDGAR is not available because EDGAR is reporting that it is not within operating hours.

- “Down”
  - EDGAR is down because the API cannot communicate with EDGAR.
- “Not authorized” (failure)
  - The Filer API Token is not valid

Additional information is provided in Figure E.

FIGURE E: EDGAR STATUS API INPUTS AND OUTPUTS


Headers
Filer API Token (JWE)
Path
/status
GET
Input
Status API
Output
Success (In Business Hours)
Body (JSON)
{ "status": "ACCEPTING SUBMISSIONS", }
Success (Out of Business Hours)
Body (JSON)
{ "status": "NOT AVAILABLE", }
Success (EDGAR is DOWN)
Body (JSON)
{ "status": "DOWN", }
Failure
Body (JSON)
{ "error": "not authorized", }