



# BEAD Challenge and Rebuttal API

*Version 1.0*

*Broadband Equity, Access, and Deployment Program*

September 2023




*This document is intended solely to assist recipients in better understanding the Broadband Equity, Access, and Deployment (BEAD) Program and the requirements set forth in the Infrastructure Investment and Jobs Act, [Notice of Funding Opportunity \(NOFO\)](#), as modified by the [BEAD Restructuring Policy Notice \(RPN\)](#). This document does not and is not intended to supersede, modify, or otherwise alter applicable statutory or regulatory requirements, the terms and conditions of the award, or the specific application requirements set forth in the NOFO not modified by the RPN. In all cases, statutory and regulatory mandates, the terms and conditions of the award, and follow-on policies and guidance, shall prevail over any inconsistencies contained in this document.*

# BEAD Challenge And Rebuttal API

## OVERVIEW OF THE PORTAL



### Core Functions

-  **Convey Challenges** | A core function of the challenge portal is to convey challenges to service providers and allow these service providers to rebut or concede challenges.
-  **Core Set of Functions** | The application programming interface (API) described in this document provides a core set of functions that allows eligible challengers to submit challenges and service providers to enumerate and respond to these challenges.
-  **API Client** | The challenger and service provider implementation acts as the API client, querying the server running on the Eligible Entity challenge portal.

### Modified Subset

Since many service providers are already responding to challenges in the FCC Broadband Data Collection (BDC), this API is a modified subset of the API described in “Broadband Data Collection — Filer API Specifications” (last updated August 16, 2023). The modifications are limited to accommodating the different challenge types allowed during the Eligible Entity challenge process. The Eligible Entity challenge process also has fewer phases, thus changing the bucket labels.

### “submitChallenge” Endpoint

To accommodate Eligible Entities whose approved Initial Proposal includes “modifications to reflect data not present in the National Broadband Map,” but which want to allow broadband providers to challenge such modifications, **the API adds a “submitChallenge” endpoint.** Challenge portals MAY also allow units of local government, nonprofit organizations, and broadband providers to submit challenges through this endpoint.

# BEAD Challenge And Rebuttal API

## OVERVIEW OF THE PORTAL



### Applying Existing Software

Implementors of the FCC BDC API should be able to adjust their existing software to handle the additional cases and conditions without fundamentally changing their application architecture. In most cases, they can simply not include irrelevant fields in the responses. However, challenge portal implementors do not have to build the complete BDC functionality. For robustness, challenge portals **MUST** ignore any unknown fields.

The keywords "MUST," "MUST NOT," "REQUIRED," "SHALL," "SHALL NOT," "SHOULD," "SHOULD NOT," "RECOMMENDED," "MAY," and "OPTIONAL" in this document are to be interpreted as **described in RFC 2119**.

All requests **MUST use HTTPS to an API endpoint** specified by each Eligible Entity.



### Recommendations



**Sandbox API** | It is **RECOMMENDED** that Eligible Entities provide a separate “sandbox” (demo) API end point that allows providers to retrieve and respond to fictional (test) challenges. Eligible Entities **SHOULD** provide a technical contact so that implementor questions can be addressed.



**Withdrawn Challenges** | Eligible Entities **SHOULD** notify the provider by email if a challenge is withdrawn. All requests return JSON.



**ID Number** | Both the challenge evidence and the rebuttal share the same file ID numbering space, i.e., the same number **MUST NOT** be used for both an evidence file and a rebuttal file. The file IDs **MUST** be unique across all challenges and all provider FRNs.

# BEAD Challenge And Rebuttal API

## COMMON ENDPOINT REQUIREMENTS AND BEHAVIORS



### Base Path

All API calls are prefixed by `/api/filer`. The prefix is not shown in the API paths below.



### API Authentication

**The API uses custom headers to authenticate the request.** The username header field contains the username of the querier, registered with the challenge portal. **It is RECOMMENDED that the challenge portal uses email addresses as identifiers**, but MAY choose other ASCII-only identifiers. The identifier is case-sensitive. The `hash_value` header field contains an ASCII API key (access token), typically generated dynamically by the challenge portal.

Implementations MAY also offer OAuth2 authentication.



### Standard Response

All API endpoints that return a JSON response<sup>1</sup> return an object with the elements `data` (the endpoint-specific data, if any), `result_count` (the number of results), `status_code` (the HTTP status code), `message` (any error message), `status` (the HTTP status message), and `request_date` (date of request).

#### Example:

```
{
  "data": [],
  "result_count": 0,
  "status_code": "200",
  "message": null,
  "status": "successful",
  "request_date": "2022-04-08T12:50:38.962Z"
}
```

1. Somewhat similar to [GitHub - omniti-labs/jsend](https://github.com/omniti-labs/jsend): JSend is a specification for a simple, no-frills, JSON based format for application-level communication., but not quite aligned.

# BEAD Challenge And Rebuttal API

## OVERVIEW OF THE REBUTTAL PROCESS



To retrieve challenges and to submit rebuttals, the API client performs the following steps:

- 1 View the List** | View the challenge list using the `getchallenges` endpoint. The `date_range_start` query parameter can restrict challenges to those received since the last query. The endpoint returns, among other information, a list of challenge IDs.
- 2 Retrieve Details** | For each challenge ID, use the `getChallengeDetails` endpoint to retrieve details about the challenge, including the evidence file identifier `evidence_file_id`.
- 3 Retrieve the Evidence File** | Retrieve the evidence file using the `downloadFile` endpoint.
- 4 Concede the Challenge** | To concede the challenge, submit a `submitInitialResponse` request with an “Accept” `response_type` parameter.
- 5 Rebut the Challenge** | To rebut the challenge using an evidence file, first upload the file using the endpoint `uploadResponseFile`. The same file can be used to rebut multiple challenges.
- 6 Obtain the Identifier** | To obtain the file identifier for the evidence file, use the endpoint.
- 7 Rebut the Challenge** | To rebut the challenge, use the endpoint and include the file identifier obtained in the previous step in the `submitInitialResponse` with the “Reject” `response_type` parameter as `response_file_id`. Comments can be included with the `response_comment` parameter.

Broadband providers can either query the challenge portal on a regular basis or wait for an email notification from the challenge portal.

# BEAD Challenge And Rebuttal API

## COMMON FIELDS AND PARAMETER DEFINITIONS



The following properties and query parameters are used. The data type definitions use the terminology in JSON schemas<sup>2</sup>. For strings, the notation ‘string{N}’ indicates a fixed-length string of N characters. Other properties listed in the BDC API are not used. They **MUST** be ignored if returned in a response or included in the query string.

Property or query parameter	Definition	Type	Example
address_primary	Address associated with the challenged location_id.	string	1401 Constitution Ave NW
brand_name	Brand name of the provider being challenged at this location_id.	string	Compuserve
bucket_name	The current bucket (i.e., group) in which a challenge resides (see “challenge states” below). Default: Disputed.	string	Disputed
building_type_code	Building type of challenged location_id. (See <a href="https://us-fcc.app.box.com/v/bdc-data-downloads-output">https://us-fcc.app.box.com/v/bdc-data-downloads-output</a> , p. 73)	string{1} (B, R, X, G, C, E)	R
category_code_desc_abbrev	Category/type of fixed broadband challenge, using the values from the Public Notice	string{1} (A, S, L, D, T, B, E, P, N, C, R)	A
challenge_id	The unique identifier of the fixed broadband or CAI challenge	integer	12345
City	Address city associated with the challenged location_id	string	Washington
date_received	Date the challenge was received from the unit of local government, non-profit, or provider	date	2023-11-05
date_range_end	Challenges submitted on or before	date	2023-11-06
date_range_start	Challenges submitted on or after	date	2023-11-05

2. <https://json-schema.org/>



# BEAD Challenge And Rebuttal API

## COMMON FIELDS AND PARAMETER DEFINITIONS CONT'D



Property or query parameter	Definition	Type	Example
data_vintage	Biannual submission date as-of fate associated with the challenge	date	2022-12-01
due_date	Due date of rebuttal	date	2023-12-05
entity_name	Entity name of the entity that submitted the challenge at this location_id.	string	Lincoln County
evidence_description	Evidence description provided by challenger (if provided).	string	Speed test screenshot.
evidence_file_id	ID of optional evidence file submitted by challenger (if provided)	integer	1234
file_id	File ID associated with the evidence or rebuttal file.	integer	1234
file_name	File name, chosen by uploader, for the evidence or rebuttal file. Must only contain letters, digits, hyphen, period, space, and underscore.	string	map1_dc.pdf
frn	FRN of provider, with leading zeroes	string	0023697626
holding_company_name	Holding company of the FRN being challenged.	string	Acme
ids	List of challenge IDs	array	[7,42,1600]
location_id	location_id of challenge from broadband serviceable Fabric	integer	1357047649

# BEAD Challenge And Rebuttal API

## COMMON FIELDS AND PARAMETER DEFINITIONS CONT'D



Property or query parameter	Definition	Type	Example
page_number	The challenge “batch” (e.g., for page_size = 10000 and page_number = 2, the portal would return challenge number 10,001 through 20,000. For, page_size = 10000 and page_number = 7 would return challenge number 60001 through 70000.) Default: 1	integer	7
page_size	The maximum number of challenge records to return in the API call. Recommended: 10000. The challenge portal is not expected to return a larger number than 10,000 records and SHOULD return 422 (Unprocessable content) status code if the page_size exceeds its capabilities. Default: 10000	integer	10000
request_date	Date the challenging entity or consumer requested service from the challenged provider or conducted the speed test.	date	2023-08-17
request_method_code_desc	Method that challenger used to request service from the provider. 1: email; 2: phone; 3: website; 0: other or unknown.	int{1}	2
resolution_comment	Eligible Entity furnished comments associated with challenge resolution.	string	Provider evidence of service accepted.
resolution_date	Date Eligible Entity adjudicated the challenge	date	2023-12-31



# BEAD Challenge And Rebuttal API

## COMMON FIELDS AND PARAMETER DEFINITIONS CONT'D



Property or query parameter	Definition	Type	Example
resolution_type	Eligible Entity resolution of challenge (if adjudicated); empty if pending. Accept = Eligible Entity has deemed the challenge to be valid; Reject = The Eligible Entity has deemed the challenge to be invalid.	string	Accept
response_file_id	File ID associated with the file uploaded by the provider during the rebuttal period.	integer	1234
response_comment	Comment submitted by provider during the rebuttal period.	string	Service upgraded at location
response_date	Date/time of provider's response during the Initial Review period.	date or date-time	2023-08-17 2023-03-31T09:01:07Z
state	Address state associated with the challenged location_id.	string{2}	DC
technology	Technology code of technology being challenged.	integer	10
uploaded_by	Name of user that uploaded the file.	string	<a href="mailto:name@example.com">name@example.com</a>
uploaded_date	Date file was uploaded	date	2023-08-17
zip_code	Address zip code associated with the challenged location_id.	integer{5}	20230

# BEAD Challenge And Rebuttal API

## CHALLENGE STATES



### Different Bucket Names

The challenge workflow states differ from those in the BDC challenge process and thus use different bucket names.



### Common Application

This facilitates creating a common application for both the Eligible Entity and BDC challenge process, if desired.



### Five Different Buckets

Each challenge can be in one of five buckets, described in the Challenge Policy Notice<sup>3</sup>:

- “Challenged”
- “Disputed”
- “Sustained”
- “Rejected”, and
- “Withdrawn”

*Bucket Names are case-sensitive<sup>4</sup>.*

3. [https://ntia.gov/sites/default/files/publications/bead\\_challenge\\_process\\_policy\\_notice\\_final.pdf](https://ntia.gov/sites/default/files/publications/bead_challenge_process_policy_notice_final.pdf)

4. Roughly, “challenged” corresponds to the “initial review” bucket, “disputed” to “active disputes”.

# BEAD Challenge And Rebuttal API

## API ENDPOINTS



### VIEW CHALLENGE LIST (BY FRN)

**Method:** GET

**Path:** /crowdsource/fixed/provider/getChallenges/{frn}

**Query parameters (optional):** challenge\_id, location\_id, category\_code, technology, date\_range\_start, date\_range\_end, due\_date, bucket\_name, page\_size, page\_number.

**Returns:** challenge\_id, bucket\_name, location\_id, technology, data\_vintage, date\_received, category\_code\_desc\_abbrev, response\_type.

**Description:** Provides a full listing of challenges submitted by eligible challengers against the coverage of a specified FRN for a particular challenge grouping (i.e., “bucket”). Since a single provider may receive many challenges, data is returned in pages of size page\_size. If the frn path element is set to 0, challenges submitted by the user are listed. The due\_date parameter restricts listings to challenges due on or before the date.

### VIEW CHALLENGE LIST (BY CHALLENGE ID)

**Method:** GET

**Path:** /crowdsource/fixed/provider/getChallengeDetails/{challenge\_id}

**Query parameters (optional):** challenge\_id,

**Returns:** challenge\_id, bucket\_name, location\_id, technology, data\_vintage, brand\_name, holding\_company\_name, address\_primary, city, state, zip\_code, building\_type\_code\_desc, category\_code\_desc\_abbrev, request\_date (for challenge types A and S), request\_method\_code\_desc (for challenge type A), date\_received, entity\_name, evidence\_description, evidence\_file\_id, response\_type, response\_comment, response\_file\_id

**Returns (optional):** If the Eligible Entity chooses, the API may return information about the challenger. For broadband providers, the FRN is returned.

frn, entity\_name

**Description:** Provides all details for a given fixed challenge including status, due dates, as well as details on the location, technology, reason for the challenge, evidence, among many others.

# BEAD Challenge And Rebuttal API

## API ENDPOINTS cont'd



### VIEW CHALLENGE DETAILS (BY CHALLENGE ID)

**Notes:** Many of the FCC Filer API output fields, such as the final response or initial revert fields, are not relevant to the Eligible Entity challenge process. Challenge portals SHOULD NOT include these fields in their response; client implementations MUST ignore these fields.

### DOWNLOAD FILE

**Method:** GET

**Path:** /crowdsource/fixed/provider/downloadFile/{file\_id}

**Query parameters (optional):** file\_id,

**Returns:** PDF file (Content-Type: application/pdf)

**Description:** Allows users to download an evidence file submitted by a challenger for a particular fixed challenge (note: not all challenges will include an evidence file) or a rebuttal file submitted by a provider in response to a challenge.

### UPLOAD RESPONSE FILE

**Method:** POST (multipart/form-data)

**Path:** /crowdsource/fixed/provider/uploadResponseFile/{frn}/2

**Query parameters (required):** frn

**Form parameters:** upload

**Returns:** standard response

**Description:** Allows user to upload a PDF response file that can later be used to respond to one or more challenges. The user then obtains a list of file identifiers. The form upload provides the filename parameter, which is also included in the file list below.

### VIEW RESPONSE FILE LIST

**Method:** GET

**Path:** /crowdsource/fixed/provider/getResponseFiles/{frn}

**Query parameters (required):** frn

**Returns:** frn, file\_id, file\_name, uploaded\_by, uploaded\_date

**Description:** Returns a listing of response files uploaded using the Upload Response File API endpoint above. The user uses this function to obtain the file IDs based on their locally-chosen file names.

# BEAD Challenge And Rebuttal API

## API ENDPOINTS cont'd



### DELETE RESPONSE FILE

**Method:** DELETE

**Path:** /crowdsourcing/fixed/provider/deleteResponseFile/{file\_id}

**Query parameters (required):** file\_id

**Returns:** standard response

**Description:** *Deletes file identified by file\_id.*

### SUBMIT RESPONSE (REBUTTAL)

**Method:** POST

**Path:** /crowdsourcing/fixed/provider/submitInitialResponse

**Query parameters (required):** -

**Body parameters (JSON):** ids, response\_type, response\_file\_id, response\_comment

**Returns:** standard response

**Description:** *Allows users to concede or dispute fixed challenges in bulk during the rebuttal period.*

### SUBMIT CHALLENGE

**Method:** POST

**Path:** /crowdsourcing/fixed/provider/submitChallenge/{location\_id}

**Query parameters (required):** -

**Body parameters (JSON):** category\_code\_desc\_abbrev, data\_vintage, evidence\_file\_id, frn, technology

**Returns:** standard response

**Description:** *Allows users to submit a challenge. The service being challenged is described by the combination of location\_id, technology, and provider FRN (frn).*

# BEAD Challenge And Rebuttal API

## API ENDPOINTS cont'd



### WITHDRAW CHALLENGE

**Method:** POST

**Path:** /crowdsource/fixed/challenger/withdrawChallenges/{frn}?id\_list=id1, id2

**Query parameters (required):** frn, id\_list

**Returns:** standard response

**Description:** Allows user to withdraw challenges identified by the list of challenge identifiers in the query. The challenge is added to the “Withdrawn” bucket.

### BUCKET TALLIES

**Method:** GET

**Path:** /crowdsource/fixed/challenger/tallyBuckets/{frn}

**Query parameters (required):** -

**Returns:** Returns a JSON object with the counts of challenges for each bucket. Empty buckets MAY be omitted. Example:

```
{“Challenged”: 10, “Withdrawn”: 5}
```

**Description:** The request returns an object with the bucket names and current challenge counts for the FRN, in any order.