# Overview

## Media Types

The Cablecast API supports two media types, `**application/json**` and `**application/xml**`. The type of response is determined by the `**Accepts**` HTTP header. When performing a `**POST**` or `**PUT**` request the `**Content-Type**` HTTP header should be set to the media type of the payload.

## Response Structure & Sideloading

The Cablecast API places all resources in a single root object for `**application/json**` and `**application/xml**` media types. Endpoints that support a single resource, such as `**POST**` and `**PUT**`, will use the singular form of the resource as a property of the root to transmit the resource. Endpoints that support a collection of resources, such as `**GET**`, will use the plural form of the resource as a property of the root to transmit the resource.

Some endpoints support sideloading related resources by specifying an `**include**` query parameter. The `**include**` parameter should be provided as a comma-separated list of singular resources.

The following example shows what a `**POST**` request for a `**Category**` would look like. Notice the singular `**Category**` property contained within the root.

```
{
  category: {
```

```
      name: 'Category Name'
    }
  }
```

The following example shows the response for a **`GET`** request for **`shows`** using the URL **`/cablecastapi/v1/shows?include=reel,media`**. Notice the **`reels`** and **`media`** collections of sideloaded with the **`shows`** collection.

```
{
  shows: [
    {
      id: 1,
      title: 'An Awesome Show'
    }
  ],
  reels: [
    {
      id: 99,
      showId: 1,
      length: 200
      media: 9999
    }
  ],
  media: [
    {
      id: 9999,
      name: 'An Awesome Media'
    }
  ]
}
```

## Pagination

Endpoints that are expected to expose a large collection of resources such as **`shows`**, **`scheduleitems`**, **`vods`**, and **`digitalfiles`** return paginated results. These endpoints

support two query parameters to control the paginated results. The `**offset**` query parameter specifies the index at which the results will begin. The `**page_size**` query parameter specifies the number of results that will be returned.

Paginated endpoints return a `**meta**` property in the response payload with three properties. The `**offset**` and `**pageSize**` properties correspond to the query parameters provided to the request. (If no query parameters are provided, the `**meta**` object will represent the default values for that endpoint.) Additionally, `**meta**` contains a `**count**` property which indicates how many total results are available on the server for the current paginated result set. An example of the meta property is provided below. The example indicates that the current results included results 23 through 43 and that there are 199 results available on the server.

```
{
  meta: {
    offset: 23,
    pageSize: 20,
    count: 199
  }
}
```

## Authentication

All endpoints that create, update, or delete record as well as endpoints that allow reading sensitive methods require authentication. If no authentication is provided, the request will error with a `**401 un-authorized**` status code. Authorization can be provided in two ways, described below.

### Cookie Based Authentication

The Cablecast API will authenticate requests that contain a cookie linking the request to an active Frontdoor session. This authentication method should be used by applications that run along side the Cablecast web interface such as Plugins. All other applications should use HTTP Basic Authentication described next.

### HTTP Basic Authentication

The Cablecast API will authenticate requests that provide a valid `**Authorization**` header using valid Frontdoor credentials. For more information on basic authentication see: https://en.wikipedia.org/wiki/Basic_access_authentication.

## Caching

To provide the best possible performance for common use cases, the Cablecast API will cache anonymous requests for several minutes. It is recommended that clients respect the `**Cache-Control**` response headers. If un-cached content is required, the request should be authenticated.

## Cross Origin Resource Sharing (CORS)

The Cablecast API is configured, by default, to allow Cross Origin Resource Sharing among all domains for read only `**GET**` requests. This will allow the Cablecast API to be accessed via Javascript from modern web browsers. If modification is required, the CORS headers can be changed by editing the `**Web.config**` located in the `**CablecastAPI**` web directory on the Cablecast server. Below is the default configuration for the CablecastAPI. For more on CORS see https://developer.mozilla.org/en-US/docs/Web/HTTP/Access_control_CORS.

```
<system.webServer>
  <httpProtocol>
    <customHeaders>
      <add name="Access-Control-Allow-Origin" value="*" />
      <add name="Access-Control-Allow-Methods" value="GET,HEAD"/>
      <add name="Access-Control-Allow-Headers" value="Content-Type" />
    </customHeaders>
  </httpProtocol>
</system.webServer>
```

## Versioning & Compatibility

There are two version numbers when interacting with the Cablecast API.

The first version number is the Cablecast version number which can be obtained programmatically using the SystemInfo endpoint. This version number can be used to ensure that the Cablecast server is compatible with the version required. It is recommended that clients confirm the Cablecast version required when accessing the Cablecast API.

The second version number is the Cablecast API version which is currently **`v1`**. The Cablecast API version will only be incremented if breaking changes are introduced to the API, meaning that an endpoint or model property has changed or been removed. Additions to the API, including new endpoints, models, or properties do not cause an increment in the API version. Therefore, it is important that clients be resilient to additional properties in request payloads. It is recommended to only rely on properties needed by the client application, not exact schemas.

When a breaking changed is introduced to the Cablecast API, necessitating a new Cablecast API version, the depreciated endpoints will still be made available for at least one Cablecast release cycle. During this time the depreciated endpoints will include an additional **`Warning`** header indicating the endpoint has been deprecated. It is recommended that clients monitor for the **`Warning`** header and alert users for the need to update the client.