v1 vs v2:

The "v1" API is every API endpoint under /api/v1. This is the original EPIC REST API. It has various issues to be aware of:

- v1 includes some APIs put together a long time ago, even for the original bd_mgmt prototype. The style of this API has developed in an ad-hoc lessons-learned way, often by different developers, and so different parts of the v1 API may handle operations and representation in different ways. For various reasons it has been difficult to prioritize the resources/time necessary to retrofit many of these APIs.
- This API was meant to support the UI and may not properly validate input when used in ways other than what the UI does. This won't cause functional issues with our product but could cause API responses that would be confusing to a user.
- This API was not meant for public consumption and is not as amenable to explanations of how it should be used.

The "v2" API is every endpoint under /api/v2. It has a consistent style of operation and representation, good input validation, comprehensive reference documentation, etc. etc. We are slowly but steadily adding v2 API functionality that matches and extends the APIs available under v1. New EPIC features will be implemented in v2 only, unless they are part of an object that is still only represented in the v1 API.

Each EPIC release (beginning with EPIC 3.0) has more of the v2 API, both in terms of new features and in terms of coverage for v1 functionality. So the available v2 API on an EPIC server depends on its version of EPIC.

The v2 API reference docs for a given EPIC installation can be seen by connecting to the /apidocs URL on its controller host.

Our v2 API docs also do have annotations that mark the EPIC version at which each part of the API was introduced. Unfortunately these annotations are not currently visible in the /apidocs viewer, but we'll work on that. They are at least available by looking at the underlying RAML documents that the /apidocs viewer reads. (Look for the "common.epicversion" annotation.) In our "everest" codebase the RAML docs live in the mgmt/controller/server/apidocs directory. On an EPIC controller the docs are in the /var/www/html/apidocs directory.

Customer demand for API access got ahead of our efforts making a well-groomed public REST API, so by necessity we have started exposing the v1 API to certain customers. Therefore we have no immediate plans to disable any part of the v1 API, but ideally it will go away someday... the input-validation issues alone make it a pain to support, and it is more likely to bit-rot as our UI moves to using v2 instead of v1.

API users should be encouraged to use v2 where possible. This is complicated by the fact that the v2 API is growing across EPIC releases as mentioned above, but as things currently stand here are some rules of thumb:

- Use the v2 cluster API. The v1 cluster API is the oldest/creakiest, and it lacks all of the compelling cluster features from EPIC 3.0 onward.
- In EPIC 3.1 and later, use the v2 session API instead of the v1 login/logout/session APIs. It is easier to work with.
- New features are likely to only be available in the v2 API, especially if they are cluster related, in which case you have to use v2.

Note that session tokens generated from v1/login can be used with the v2 API, and conversely session tokens generated from v2/session can be used with the v1 API. So you can mix and match v1 and v2 API operations using a single session.

v1/v2 API coverage summary (as of EPIC 3.3):

v1 APIs

/bundle
  v1 only

/catalog
  v1 only
  - priority for v2 replacement

/chart
  v1 only

/cluster
  almost completely replaced by v2/cluster (see v2 APIs list)
  - v2/cluster added in 3.0
  - v2/cluster/<cluster_id>/node and nodes param added in 3.2
  - v2 cluster powerops added in 3.2
  - not yet in v2: fetching service status and jobs-for-this-cluster
  - priority for finishing v2 replacement

/config
  starting to migrate to v2/config (not much yet)
  - read-only v2/config/auth added in 3.2
  - read-only v2/config/tenant_types added in 3.3

/dataconn
  v1 only
  - priority for v2 replacement

/feed
  v1 only

/flavor
  v1 only
  - priority for v2 replacement

/floatinfo
  v1 only

/install
  v1 only

/job
  v1 only

/license
  v1 only

/login
  replaced by v2/session (see v2 APIs list)
  - v2/session added in 3.1

/logout
  replaced by v2/session (see v2 APIs list)
  - v2/session added in 3.1

/lock
  v1 only

/phynode
  v1 only

/role
  v1 only

/session
  replaced by v2/session (see v2 APIs list)
  - v2/session added in 3.1

/stats
  v1 only

/tenant
  v1 only

/testauthconfig
  v1 only

/testcluster
  replaced by v2/cluster (see v2 APIs list)
  - v2/cluster dryrun param added in 3.1

/testdataconn
  v1 only
  - priority for v2 replacement

/upgrade
  v1 only

/user
  v1 only

/virtnode
  replaced by v2/cluster and v2/node (see v2 APIs list)
  - v2/cluster/<cluster_id>/node and nodes param added in 3.2
  - v2 cluster powerops added in 3.2
  - v2/node added in 3.2

/virtnodehistory
  v1 only

/workers
  v1 only

v2 APIs

/cluster
  replaces v1/cluster and v1/testcluster APIs, and v1/virtnode API (also see v2/node)
  supports new cluster-related features from 3.0 onward
  - v2/cluster added in 3.0
  - v2/cluster/action_task and action_history added in 3.0
  - v2/cluster/change_task and change_history added in 3.0
  - v2/cluster dryrun param added in 3.1
  - v2/cluster read-only access for site admin added in 3.2
  - v2/cluster/<cluster_id>/node and nodes param added in 3.2
  - v2 cluster powerops added in 3.2

/config
  starting to replace v1/config API
  - v2/config/auth added in 3.2
  - v2/config/tenant_types added in 3.3

/node
  replaces v1/virtnode API (also see v2/cluster)
  - v2/node added in 3.2

/session
  replaces v1/login, v1/logout, and v1/session APIs
  - v2/session added in 3.1

/tag
  new features
  - v2/tag added in 3.1

/template
  new features

```
  - v2/template added in 3.1

/tenant_filesystem
  new features
  - v2/tenant_filesystem added in 3.3
```